University of Zurich^{UZH}

Burkhard Stiller, Daniel Dönni, Lisa Kristiana, Patrick Poullie, Guilherme Machado, Andri Lareida, Corinna Schmitt, Thomas Bocek, Radhika Garg, Christos Tsiaras (Eds.)

Internet Economics IX

TECHNICAL REPORT – No. IFI-2015.01

January 2015

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

B. Stiller et al. (Eds.): Internet Economics IX
Technical Report No. IFI-2015.01, January 2015
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch

Introduction

The Department of Informatics (IFI) of the University of Zürich, Switzerland works on research and teaching in the area of communication systems. One of the driving topics in applying communications technology is addressing investigations of their use and application under economic constraints and technical optimization measures. Therefore, during the autumn term HS 2014 a new instance of the Internet Economics seminar has been prepared and students as well as supervisors worked on this topic.

Even today, Internet Economics are run rarely as a teaching unit. This observation seems to be a little in contrast to the fact that research on Internet Economics has been established as an important area in the center of technology and economics on networked environments. After some careful investigations it can be found that during the last ten years, the underlying communication technology applied for the Internet and the way electronic business transactions are performed on top of the network have changed. Although, a variety of support functionality has been developed for the Internet case, the core functionality of delivering data, bits, and bytes remained unchanged. Nevertheless, changes and updates occur with respect to the use, the application area, and the technology itself. Therefore, another review of a selected number of topics has been undertaken.

Content

This new edition of the seminar entitled "Internet Economics IX" discusses a number of selected topics in the area of Internet Economics. The first talk "IP Flow Information Export (IPFIX) Protocol" provides a detailed characterization of IPFIX. Talk two "Feasibility of Multisig in CoinBlesk" discusses security-related aspects of CoinBlesk with special reference to using multiple private keys. Talk three "Cloud-based Services: To Move or Not to Move" discusses the various aspects that need to be taken into consideration when deciding whether to move services to a cloud or not. Talk four "Applicability of Cryptographic Protocols to Support Service Level Agreements" introduces secure mechanisms which can be used for automated monetary compensations in case of SLA violations. Talk five on "QoE-based Charging" presents a financial framework and current research in the area of QoE-based charging. Talk six on "QoS in Mobile Ad-hoc Networks (MANET)" introduces MANETs, the underlying technologies, as well as QoS aspects. Talk seven on "Municipal Wireless Networks" presents technological and economic aspects of Municipal Wireless Networks. Talk 8 on "Fairness Indices and Notions in Communication Systems" is divided in two parts: 8a discusses specific problems arising in the context of multi-resource allocation along with current solutions, 8b discussess fairness measurement mechanisms. Finally, "Internet Service Providers: Peering and Charging" discussess peering and charging aspects among network providers in today's Internet.

Seminar Operation

Based on well-developed experiences of former seminars, held in different academic environments, all interested students worked on an initially offered set of papers and book chapters. Those relate to the topic titles as presented in the Table of Content below. They prepared a written essay as a clearly focused presentation, an evaluation, and a summary of those topics. Each of these essays is included in this technical report as a separate section and allows for an overview on important areas of concern, sometimes business models in operation, and problems encountered.

In addition, every group of students prepared a slide presentation of approximately 45 minutes to present his findings and summaries to the audience of students attending the seminar and other interested students, research assistants, and professors. Following a general question and answer phase, a student-lead discussion debated open issues and critical statements with the audience.

Local IFI support for preparing talks, reports, and their preparation by students had been granted by Daniel Dönni, Lisa Kristiana, Patrick Poullie, Guilherme Machado, Andri Lareida, Corinna Schmitt, Thomas Bocek, Radhika Garg, Christos Tsiaras, and Burkhard Stiller. In particular, many thanks are addressed to Daniel Dönni for his strong commitment on getting this technical report ready and quickly published. A larger number of pre-presentation discussions have provided valuable insights in the emerging and moving field of Internet Economics, both for all students and supervisors. Many thanks to all people contributing to the success of this event, which has happened in a lively group of highly motivated and technically qualified students and people.

Zürich, January 2015

Contents

Chapter 1

IP Flow Information Export (IPFIX) Protocol

Anaxamene Dimitriades, Simon Hodel

The IP Flow Information Export (IPFIX) Protocol was developed by the Internet Engineering Task Force (IETF) for transmitting flow information between different instances in the network. After defining the requirements, NetFlow v9 was chosen as the base protocol among others to develop IPFIX upon. The aim of the new protocol still was to provide a set of instruments for measuring network traffic. With the template-based approach that was developed, a major improvement on NetFlow v9 has been achieved. Going beyond the limits of a simple protocol, IPFIX has become a data format, which enables efficient self-description of the actual data as well as an highly flexible information model that provides the vocabulary for its initial purpose, but furthermore facilitating the use of the protocol in a customized way for any purpose with similar architecture. This feature makes IPFIX interesting for different business approaches, starting from performant network analysis going on to sensor networks.

Contents

1.1 Introduction

The idea of IPFIX was to develop a flexible protocol that at the same time serves as an information model in order to leave the rigid frame of network traffic analysis and measurement, which its predecessor NetFlow v9 was caught in. After a basic determination of requirements for the new protocol to be developed, NetFlow v9 was elected as the base protocol for IPFIX. The development started with the goal of keeping NetFlow's good aspects and instruments and enriching those with extensible and customizable features, in order to serve technical needs aside of network analysis which although share the same elementary logical structure of devices and communication in between. So the interesting part about IPFIX is its customizability for anyone and any specific needs resembling the architecture of a typical IPFIX setting.

This paper aims at presenting the brief history of IPFIX, its characteristics, its strengths and its usage areas. First, some basic concepts and the predecessor of IPFIX, NetFlow v9, will be presented. After a detailed introduction to IPFIX itself, it will be compared with NetFlow v9. Finally, the real life use of IPFIX will be shown in two cases, as well as the benefits of using it.

1.2 Network Flow

In the context of IPFIX and its requirements defined in advance, it has been decided that IPFIX operates on network flows. This means network traffic is grouped into flows, which will be measured by IPFIX. An IP flow "is defined as a set of IP packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties." [11] A set can consist of one or more IP packets. A flow definition can be made out of one or more properties:

1. "One or more packet header field (e.g., destination IP address), transport header field (e.g., destination port number), or application header field (e.g., RTP header fields [RFC3550]).
2. One or more characteristics of the packet itself (e.g., number of MPLS labels, etc.).
3. One or more of fields derived from packet treatment (e.g., next hop IP address, the output interface, etc.)." [11]

A flow definition is also referred to as a Flow Key. Every IP packet passing the observation point, which is a certain predefined node in a network (see chapter 1.3.1 for exact term definition in the IPFIX context), will be checked against the flow definition. If it satisfies every predefined flow property, it belongs to the flow, otherwise it does not.

1.2.1 Key Data about Cisco's NetFlow

NetFlow is a protocol developed by Cisco Systems, a US enterprise in the field of networking equipment, with its first version being released in 1990 [8]. Even though the names are quite similar, NetFlow must not be confused with Network Flow, whereas the former denotes the protocol and the latter the concept of traffic and packet flow in a network (see chapter 1.2). The current proper NetFlow release is Version 9. A renaming of NetFlow v10 to IPFIX was made later on.

The protocol's aim is to provide information about IP flows in networks, which is gathered at different network nodes like routers or switches, and then forwarded to NetFlow Collectors, that "receives Flow Records from one or more Exporters. It processes the received Export Packet(s); that is, it parses and stores the Flow Record information"

[3], which collect and parse and/or store them for applications that process these data. NetFlow v9 uses a template-based approach to export flow data. ”A template defines a collection of fields, with corresponding descriptions of structure and semantics.” [3] The use of templates provides mainly two advantages: On the one hand, the data overhead is reduced, since meta information has to be sent only once in a template, and not together with every Flow Record. This reduces the data volume, which allows memory savings on both the NetFlow Exporter’s, that is the ”device that monitors packets” [3] and ”creates Flows from these packets” [3], which are being exported afterwards, and the NetFlow Collector’s side. On the other hand, the use of the protocol is more flexible. Changing the content structure of the actual data recorded on network flows (the so called Flow Records, transferred inside Export Packets) does not lead to the need of a new definition of the export format, like it was in previous NetFlow versions.
Originally 79 different Field Types were defined in RFC 3954 for NetFlow v9, describing various kinds of information that may be supported by a NetFlow Exporter [3]. RFC 3954 was released in October 2004, and it stated that ”when extensibility is required, the new field types will be added to the list” [3], which will be updated and available on Cisco’s website [1]. A key point about those Field Types is that they only may be extended by Cisco itself and not by anyone. As of May 2011, totally 104 Field Types were defined.
The introduction of the template-based approach in v9 was the main improvement on the widespread v5, which does not allow customized Flow Keys, but defines 18 exported fields, which are fixed for every Flow Record and will be transmitted in every single of those Flow Records. [12]

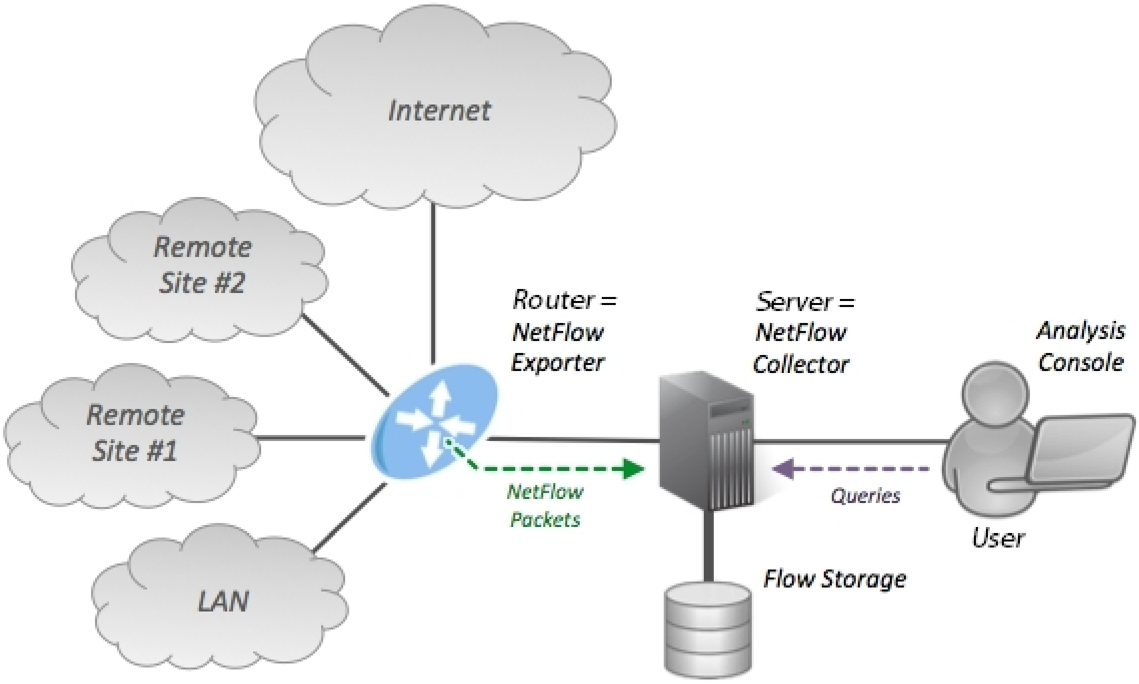


Figure 1.1: A typical setup of a NetFlow implementation on a sample network. [1]

In Figure 1.1 a typical setup of a NetFlow implementation is shown: The router connects a LAN and Remote Sites #1 and #2 with the Internet in a network architecture. It serves as the NetFlow Exporter device, aggregating passing packets to flows and generates Flow Records upon them. Those Flow Records are exported in NetFlow Packets to a server, which itself serves as the NetFlow Collector. The server collects and stores the data and

[1] `http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html`

serves as an interface for the Analysis Console (controlled and/or programmed by a user), which runs queries on the data in order to gather information about the network traffic.

1.3 IPFIX: NetFlow v10

IPFIX denotes basically a set of instruments for collecting and exporting information about network traffic.
It is based on NetFlow v9, which has been selected after an evaluation of several candidate base protocols in RFC 3955 [9]. The Internet Engineering Task Force (IETF) is the developer of IPFIX. After having elaborated the requirements (RFC 3917 [11]) and evaluating the candidate protocols in 2004, the actual protocol was specified in 2008 (RFC 5101 [4], obsoleted by RFC 7011 [6]), as well as the information model (RFC 5102 [10], obsoleted by RFC 7012 [5].
Furthermore, the architecture of IPFIX was defined in 2009 (RFC 5470 [13]).
The initial purpose was to provide a standard of tools for network measurements, more precise focussing on the network layer in the OSI model. As of today, IPFIX can be characterized mainly by three manners [15]:

1. It is a unidirectional *protocol* for data export, operating on the application layers.
2. It is a template-driven *data format*, where template-driven means the customizable composition of the data structure for the data to export.
3. It is an *information model* that provides a large vocabulary for the data format. More than 430 standardized information elements are currently available, with focus on measurement and logging tasks on the network and transport layers.

1.3.1 Terminology and Architecture

In this subsection, the most important terms of IPFIX architecture will briefly be introduced:

- **Flow Record**: ”A Flow Record contains information about a specific Flow that was observed at an Observation Point. A Flow Record contains measured properties of the Flow (for example the total number of bytes for all the Flow’s packets) and usually contains characteristic properties of the Flow (for example the source IP address).” [6]
- **Observation Point**: ”An Observation Point is a location in the network where packets can be observed.” [6] This can be an entire LAN or also a single port of a router for example. Every observation point belongs to an observation domain and can optionally be a superset of other observation points.
- **Observation Domain**: ”An Observation Domain is the largest set of Observation Points for which Flow information can be aggregated by a Metering Process. For example, a router line card may be an Observation Domain if it is composed of several interfaces, each of which is an Observation Point.” [6] Every Observation Domain has a unique ID that will be included in the concerning IPFIX Messages.
- **Packet Treatment**: ”Packet Treatment refers to action(s) performed on a packet by a forwarding device or other middlebox, including forwarding, dropping, delaying for traffic-shaping purposes” [6] and so on.

- **Metering Process**: ”The Metering Process generates Flow Records. Inputs to the process are packet headers, characteristics, and Packet Treatment observed at one or more Observation Points. The Metering Process consists of a set of functions that includes packet header capturing, timestamping, sampling, classifying, and maintaining Flow Records.” [6]

- **Exporting Process**: ”The Exporting Process sends IPFIX Messages to one or more Collecting Processes. The Flow Records in the Messages are generated by one or more Metering Processes.” [6]

- **Collecting Process**: ”A Collecting Process receives IPFIX Messages from one or more Exporting Processes. The Collecting Process might process or store Flow Records received within these Messages” [6].

- **Exporter**: ”A device that hosts one or more Exporting Processes” [6].

- **Collector**: ”A device that hosts one or more Collecting Processes” [6].

- **IPFIX Device**: ”An IPFIX Device hosts at least one Exporting Process. It may host further Exporting Processes as well as arbitrary numbers of Observation Points and Metering Processes.” [6]

- **Set**: ”A Set is a collection of records that have a similar structure” [6].

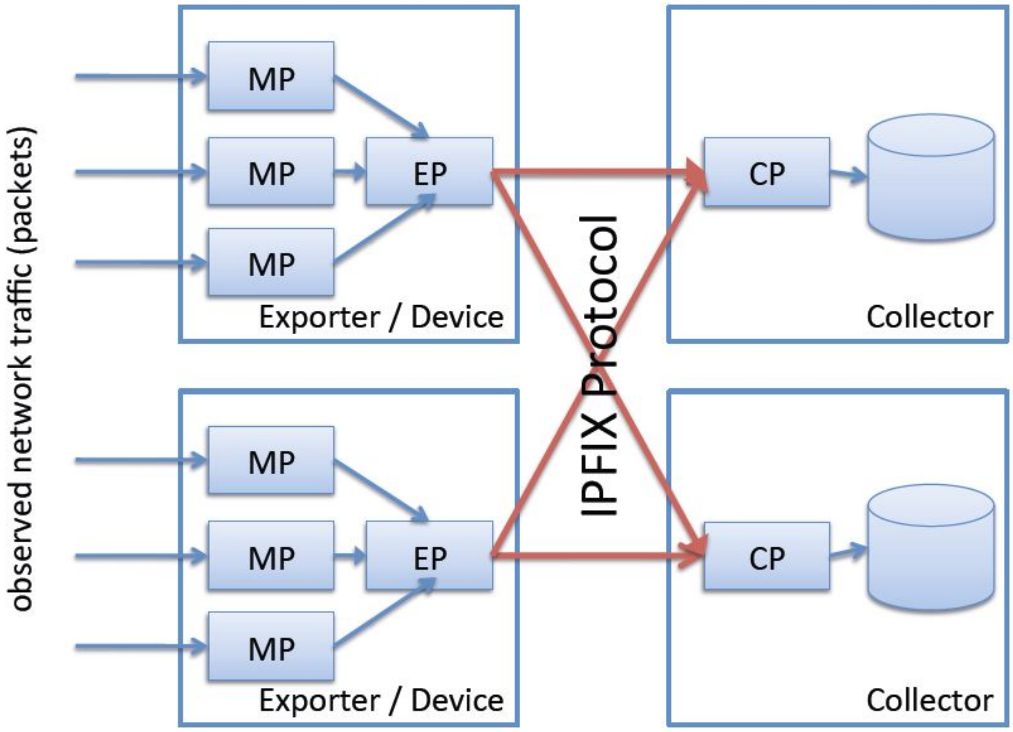


Figure 1.2: The general architecture of IPFIX processes and devices. [15]

In Figure 1.2 a caption of a network running IPFIX is shown. There are four network nodes, of which the two on the left hand side serve as Exporters and IPFIX Devices at the same time, and the two on the right hand side serve as Collectors. The Exporters each host three Metering Processes (marked as ”MP”), which analyze the network traffic (on the very left side) and each generate (different) Flow Records out of these observations. Those are forwarded to the one Exporting Process (marked as ”EP”) running on the same device. The Exporting Processes each propagate the Flow Records as IPFIX Messages to the both Collectors (where the arrows indicate the message routes and their direction), on which one Collecting Process (marked as ”CP”) is running each. In this specific case, those store the received messages in a local memory.

1.3.2 Messages and Templates

All IPFIX information is grouped and sent as messages. An IPFIX Message by definition is ”a message that originates at the Exporting Process and carries the IPFIX records of this Exporting Process, and whose destination is a Collecting Process. An IPFIX Message is encapsulated at the transport layer” [6]. Every IPFIX message consists of a Message Header, followed by zero or more Sets.
The Message Header has the following format: [6]

1. Version: IPFIX Version number.

2. Length: Total message length in octets.

3. Export Time: Time at which the header leaves the Exporter.

4. Sequence Number: Incremental sequence counter for all Data Records sent in the current stream from the current Observation Domain by the Exporting Process.

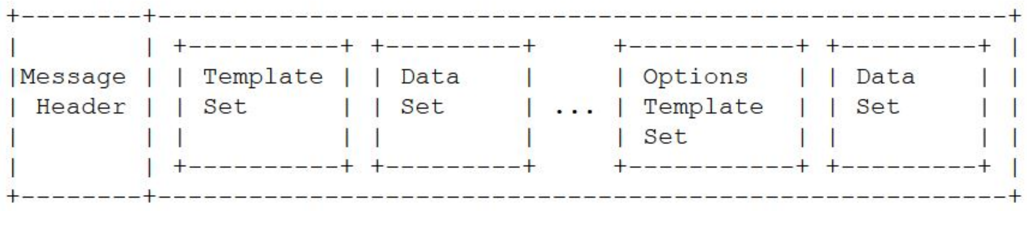
5. Observation Domain ID: Locally unique to the Exporting Process.

After the Message Header, an arbitrary number of sets may be attached to the message. Within a single Set, the records must all be of the same type. There are three different types of sets [6]:

1. Data Set: One or more Data Records, of the same type, that are grouped together. Each Data Record is previously defined by a Template Record or an Options Template Record.

2. Template Set: A collection of one or more Template Records.

3. Options Template Set: A collection of one or more Options Template Records.



```
+--------+--------------------------------------------------------+
|        | +----------+ +---------+     +-----------+ +---------+ |
|Message | | Template | | Data    |     | Options   | | Data    | |
| Header | | Set      | | Set     | ... | Template  | | Set     | |
|        | |          | |         |     | Set       | |         | |
|        | +----------+ +---------+     +-----------+ +---------+ |
+--------+--------------------------------------------------------+
```

Figure 1.3: Example of an IPFIX Message. [6]

In Figure 1.3 an example of an IPFIX Message is shown. It consists of all set types. In this specific case, the first Template Set contains the definitions for the following Data Set, same as the Options Template Set defining the data structure for its following Data Set.
An essential aspect of IPFIX is the Template concept. A Template defines the structure and the semantics of a Data Record in advance, such that a Collecting Process knows what the information it collects in a Data Record is about. This helps to reduce traffic overhead and allows the use of IPFIX in a flexible way. A Template is propagated in a Template Record, which consists of pairs of the types Information Element and Field Length. Every template is identified by a unique ID, which a Data Record later on refers to in order to provide the meta information to the Collecting Process.

An Information Element is a protocol- and encoding-independent description of an attribute that may appear in an IPFIX Record. Information Elements are defined in the IANA ”IPFIX Information Elements” registry [2]. The type associated with an Information Element indicates constraints on what it may contain and also determines the valid encoding mechanisms for use in IPFIX. Currently over 430 standardized Information Elements are available. Examples for such Information Elements are *sourceIPv4Address*, *destinationIPv4Address*, *sourceTransportPort*, *destinationTransportPort* or *protocolIdentifier*, which grouped together are known as the ”Traditional Five Tuple” that builds a popular Flow Key. Moreover multiple counters like *PacketDeltaCount* or timestamps for certain events such as *flowEndSysUpTime* are contained in the registry. A notable feature of IPFIX is the possibility of using self-defined Information Elements. The semantics of such customized Information Elements are not bound to any constraint, they only have to be registered at IANA, which will assign a unique ID together with an Enterprise Number to a new Information Element. The assigned Information Element ID and Enterprise Number will appear as an identifier for the new Information Element in the Field Specifier inside a Template Record or Options Template Record, which refers to a certain value in a corresponding Flow Record delivered afterwards.

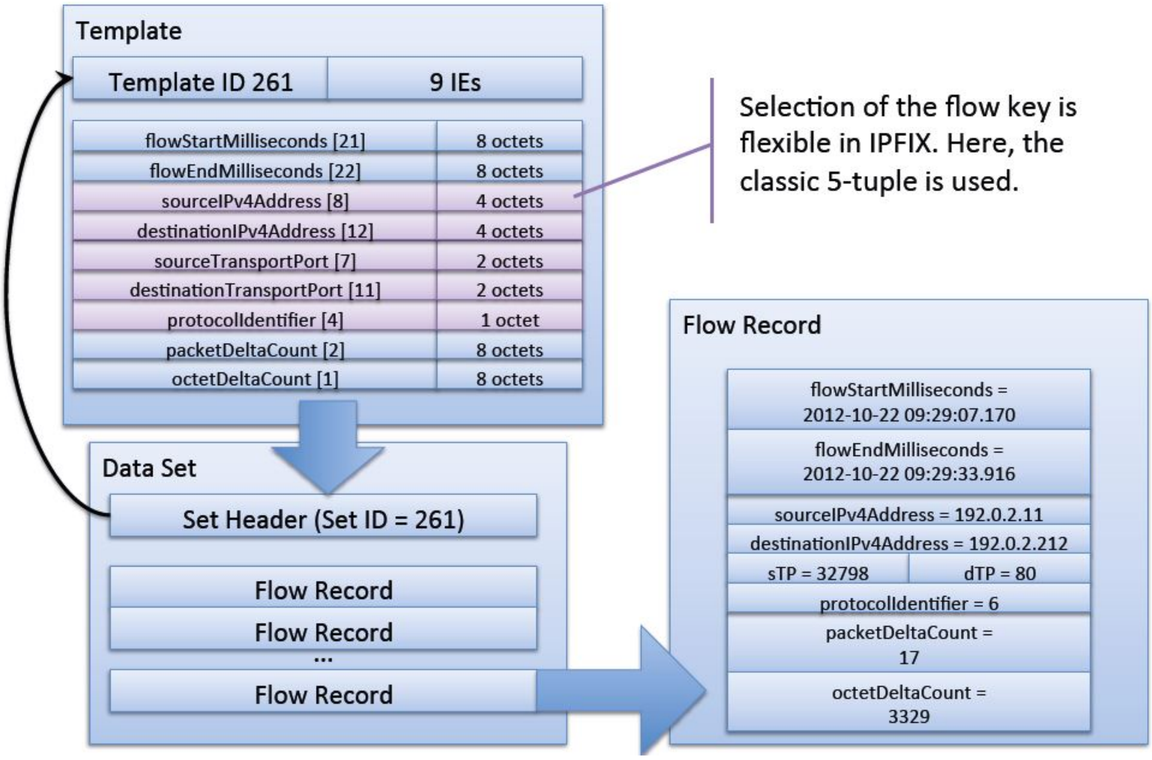


Figure 1.4: Combined functionality of a Template with the actual data in a Flow Record. [15]

In Figure 1.4 the mode of operation of a Template is shown. The Template on the top was assigned the ID 261 and it counts 9 Information Elements (marked as ”IEs”), which is indicated in the Template Header. The 9 types of Information Elements contain the ”Traditional Five Tuple”, as well as two timestamps and two counters. (In the actual protocol only the number in brackets is included, which refers to the entries in the IANA registry.) Together with each Information Element the length in octets is indicated. The Data Set at the bottom left corner is sent after the Template and contains the Template ID 261 as a Set Header element. The arrows between the Template and the Data Set indicate the transmission order as well as the logical connection between these two messages: The

[2] http://www.iana.org/assignments/ipfix/

Data Set is sent after the Template (thick blue arrow), and the Data Set contains Flow Records with the structure defined in the Template with the ID 261 (thin blue arrow). This link ensures the correct interpretation of the following Flow Records in the Data Set. The blue thick arrow connecting the Data Set with the Flow Record on the right hand side symbolizes a zoom-in on one particular Flow Record that the Data Set contains: It is shown in detail on the right hand side, and as defined in the template, there are the 9 fields of information containing the actual value.
Options Templates define records to a specific Scope. A Scope gives the context of the reported Information Elements in the Data Records and is only available in Options Template Sets. It refers to an entity in real world or in IPFIX architecture or protocol, such as an entire LAN, a single router interface, an Exporting Process or a Template. Those are defined by the use of a set of Information Elements. Collecting Processes should minimally support *observationDomainId*, *exportingProcessId*, *meteringProcessId*, *templateId*, *lineCardId*, *exporterIPv4Address*, *exporterIPv6Address*, and *ingressInterface* as Scope Information Elements. An Option Template is used to describe information about the collection infrastructure, metadata about flows (or a set of flows) or common properties of a set of flows [15].

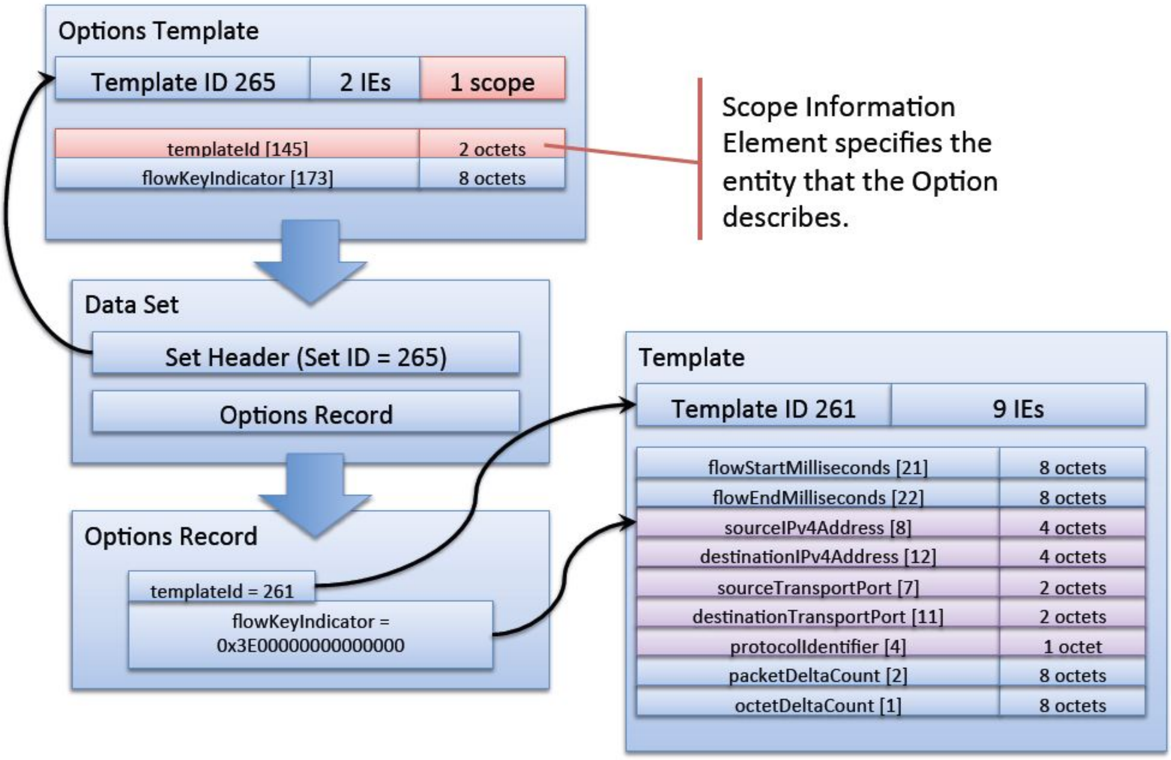


Figure 1.5: Example of the functionality of an Options Template. [15]

In Figure 1.5 the functionality of an Options Template is visualized. The Options Template on the top is sent first to a Collecting Process. In its header, it is uniquely identified by the Template ID 265. The second header field indicates the total number 2 of Information Elements (marked as ”IEs”) of the Options Template, including those which serve as Scope identifier. The Scope field determines which Information Elements belong to the Scope definition, starting from the first straight up to the last Information Element, which here is only one. So the Information Element *templateId* serves as the Scope identifier and has a length of 2 octets, whereas the second Information Element is the Option. In the Data Set that is received by the Collecting Process afterwards (indicated by the thick blue arrow showing the transmission order), the Set ID in the header, 265, refers to the Options Template above (indicated by the thin black arrow connecting Data Set and Options Template). The thick blue arrow connecting the Data Set and the Options [Template] Record symbolizes a zoom-in on the only Options [Template] Record ("defines how to scope the applicability of the Data Record" [6]) that the Data Set contains, which itself consists of the actual data. The Template ID 261 specifies the corresponding Template as the Scope (sent even before the Options Template and thus already known by the Collecting Process), which is indicated by the first thin black arrow connecting the Options Record and the Template, whereas the *flowKeyIndicator* value describes this scope, indicated by the second thin black arrow pointing from the *flowKeyIndicator* to the five Information Elements in the Template.

1.3.3 Transport

IPFIX was designed to be transport protocol independent. However, PR-SCTP (Stream Control Transport Protocol with Partial Reliability extension) must be implemented in every IPFIX installation. It should "be used in deployments where Exporters and Collectors are communicating over links that are susceptible to congestion. SCTP is capable of providing any required degree of reliability when used with the PR-SCTP extension." [6] SCTP-PR provides several features more than TCP/UDP like multiple streams between sender (Exporter) and receiver (Collector), partial reliability, where certain packets can be skipped for retransmissions, allows unordered delivery of packets and has all in all a simpler state machine than TCP. Especially the partial reliability is important for IPFIX, since Templates must be sent reliably in order to guarantee the correct data interpretation at the Collector. Hence SCTP-PR allows best-effort reliability on a UDP-level, but still provides TCP-level congestion control, that aims to ensure the performance of a network as good as possible by using several mechanisms that should prevent congestion and the consequential (partial) collapse of the network [15].

The connection-oriented and reliable TCP (Transport Control Protocol) may be used for IPFIX installations, even though the use of SCTP-PR is recommended. The implementation of TCP makes sense if IPFIX data has to be transferred over links that are prone to congestion, such as the Internet [15].

The connection-less and unreliable UDP (User Datagram Protocol) may be used as well, but should only be used on "dedicated networks within a single administrative domain" [15] due to its characteristics. Again, the use of SCTP-PR is preferred to UDP.

1.3.4 Security

As the requirements for IPFIX were defined, three basic security principles were raised: Confidentiality, integrity, and authenticity should be granted by the final version, with regard to the capability of using IPFIX securely on the Internet. [11] This lead to a more precise wording of the security requirements, as well as the protocols that are used within IPFIX.

1.3.4.1 Requirements

There are three basic security requirements for IPFIX:

1. "It must provide a mechanism to ensure the confidentiality of IPFIX data transferred from an Exporting Process to a Collecting Process, in order to prevent disclosure of Flow Records transported via IPFIX." [6]

2. "It must provide a mechanism to ensure the integrity of IPFIX data transferred from an Exporting Process to a Collecting Process, in order to prevent the injection of

incorrect data or control information (for example Templates), or the duplication of Messages, in an IPFIX Message stream.” [6]

3. ”It must provide a mechanism to authenticate IPFIX Collecting and Exporting Processes, to prevent the collection of data from an unauthorized Exporting Process or the export of data to an unauthorized Collecting Process.” [6]

In order to fulfill these requirements, combined with the characteristics of the proposed protocols on the transport layer (see section 1.3.3), the pairs of transport and session layer protocols in the following section are recommended to use.

1.3.4.2 Protocols

Depending on the used protocol on the transport layer, one of the two following protocols should be applied on the session layer [6]:

- If TCP is used as the transport protocol, TLS (Transport Layer Security) is applied.
- With UDP or SCTP(-PR) used on network layer, the quasi datagram version of TLS, DTLS (Datagram Transport Layer Security), is applied.

Both of them were designed to fulfill the three security concepts, which are requirements for IPFIX and are implemented for the use in uncontrolled or non-dedicated networks. The alternative to using those security protocols is to run the IPFIX installation inside a dedicated secure tunnel. In the latter case, security issues are shifted to the responsibility of the network tunnel that has to grant the secure transport of the messages. Hence, the application of the security protocols are not an IPFIX-specific issue anymore.

1.4 Comparison of IPFIX and NetFlow

Basically, NetFlow and IPFIX widely correspond, since IPFIX is the further development of NetFlow v9, from which it adopted the core concepts and refined it according to the determined requirements.

The main difference concerns the information model. While NetFlow provides 79 predefined ”Fields” for different information content, IPFIX provides currently more than 430 standardized Information Elements. The range of 1 - 127 of IPFIX Information Elements is compatible with the ”Fields” used by NetFlow v9. [5] The significant difference however is the customizability of the information transfered by IPFIX. The user is free to define individual Information Elements that can be used enterprise-independent. New Information Elements do not have to be linked to network traffic measurement at all, which offers a new horizont of a generic information model coming along with a data exchange protocol. Customized Information Elements have to be registered at IANA, which assigns a Type ID and an Enterprise ID to the new Information Element, such that it is globally unique and can be identified by this tuple. NetFlow does not provide this option.

This feature makes IPFIX interesting for purposes beyond of network traffic measurement. Basically everywhere a similar architecture exists with data sources and sinks, IPFIX can be customized and implemented for any individual purpose.

In Figure 1.6 five imaginary new Information Elements to be registered at IANA are shown. Beside the information about the platform, the kind of the sensor and the vendor of the technical unit, both Enterprise ID and Type ID are indicated. Whereas the Enterprise ID for two different Information Elements can be the same, like for the first and the second row, this holds as well for the Type ID, like in the first and the third row. However, the tuple <Enterprise ID, Type ID> is unique and provides for a global identification of the

Hardware platform	Platform vendor	Sensor	Vendor technical unit	Enterprise ID	Type ID
TelosB	Advantic Sys.	Temperature	Sensiron SHT11	3841	33025
TelosB	Advantic Sys.	Humidity	Sensiron SHT11	3841	33026
TelosB	Advantic Sys.	Light	Hamamatsu S1087	3845	33025
IRIS	Crossbow Inc.	Temperature	Panasonic ERT-J1VR103J	3843	32771
IRIS	Crossbow Inc.	Light	TAOS TSL2550	3846	33282

Figure 1.6: Table of custom Information Elements. [14]

Information Element. None of the Information Elements in the list does denote data in the native field of network traffic measurement, but in sensor networks, what the flexible Information Model of IPFIX allows.

1.5 IPFIX Applications

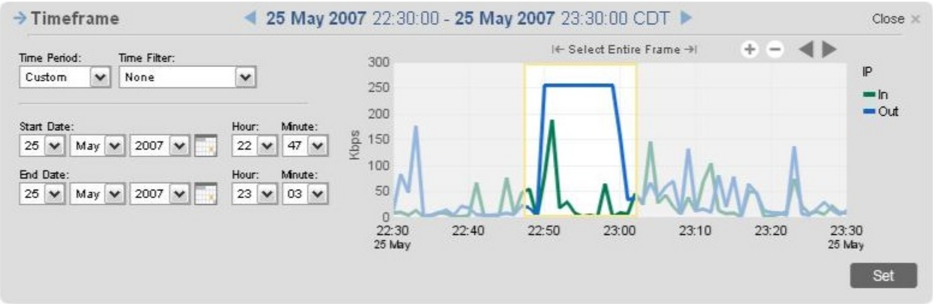
As it will be seen in the following, IPFIX can be used to solve diverse problems, and present some great benefits. Thos use case of IPFIX are presented for two real life cases, with the use of a production interface to illustrate our cases. This shows in what manner IPFIX can be used, and how easy it may be to use.

1.5.1 Case: Server unavailable

This case comes from CA Technologies [2] as well as the figures shown below.
In this case, the issue is the following:

> ”In the last 10 minutes, users in the New York office have suddenly started calling to complain that they cannot access key financial applications or server resources in the London data center.” [2]

From there a technician is able to use the IPFIX interface to get the information he needs. He is able to create a report and make a first step for the localisation of the issue.



→ Timeframe ◀ 25 May 2007 22:30:00 - 25 May 2007 23:30:00 CDT ▶

Time Period: Custom

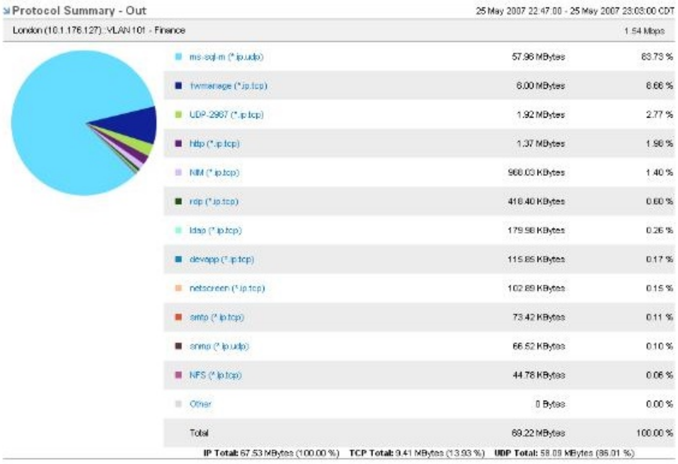
Time Filter: None

Start Date: 25 May 2007 Hour: 22 Minute: 47

End Date: 25 May 2007 Hour: 23 Minute: 03

Figure 1.7: Interface Timeframe Report. [2]

On Figure 1.7 one can see the interface he can use. On the left the selection parameter appears, where the date and time can be picked. On the right is the actual data corresponding to the request. In this case, a peak can be localised between 22:50 and 23:00 where Outs are reaching an unusual high level.
By using the interface it is possible to zoom in on the issue. A detailed report like Figure 1.8 is obtained. On the right are present protocols and their use of the network, and on the left a pie chart representation of those data. It is easy to see that there is a problem



Protocol Summary - Out 25 May 2007 22:47:00 - 25 May 2007 23:03:00 CDT

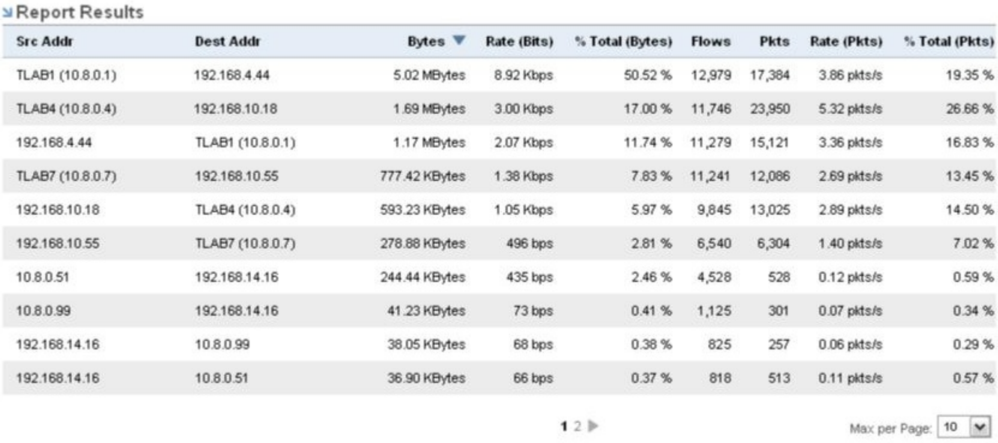
London (10.1.176.127)::VLAN 101 - Finance 1.54 Mbps

ms-sql-m (*.ip.udp)	57.96 MBytes	83.73 %
fwmanage (*.ip.tcp)	6.00 MBytes	8.66 %
UDP-2967 (*.ip.tcp)	1.92 MBytes	2.77 %
http (*.ip.tcp)	1.37 MBytes	1.98 %
NIM (*.ip.tcp)	968.03 KBytes	1.40 %
rdp (*.ip.tcp)	418.40 KBytes	0.60 %
ldap (*.ip.tcp)	179.98 KBytes	0.26 %
devapp (*.ip.tcp)	115.85 KBytes	0.17 %
netscreen (*.ip.tcp)	102.89 KBytes	0.15 %
smtp (*.ip.tcp)	73.42 KBytes	0.11 %
snmp (*.ip.udp)	66.52 KBytes	0.10 %
NFS (*.ip.tcp)	44.78 KBytes	0.06 %
Other	0 Bytes	0.00 %
Total	69.22 MBytes	100.00 %

IP Total: 67.53 MBytes (100.00 %) **TCP Total:** 9.41 MBytes (13.93 %) **UDP Total:** 58.09 MBytes (86.01 %)

Figure 1.8: Detail Protocol Report. [2]

located on the protocol ”ms-sql-m”, as the allocation for this protocol is taking 84 percent of the total capacity, which is very high.
At this point, the issue is localised but the reason of the problem is not found yet. The choice is made to dig deeper in the protocol's information, the Figure 1.9 shows the results of this request. On the figure is the report under table format corresponding to the use of the network (in packets and bytes). It's observable that six components take the most of the bandwith, the technician is able to diagnose that the problem was caused by a virus. In the present case, IPFIX helped the technician to identifiy a known virus, that he was able to remove, which fixed the issue.



Report Results

Src Addr	Dest Addr	Bytes ▼	Rate (Bits)	% Total (Bytes)	Flows	Pkts	Rate (Pkts)	% Total (Pkts)
TLAB1 (10.8.0.1)	192.168.4.44	5.02 MBytes	8.92 Kbps	50.52 %	12,979	17,384	3.86 pkts/s	19.35 %
TLAB4 (10.8.0.4)	192.168.10.18	1.69 MBytes	3.00 Kbps	17.00 %	11,746	23,950	5.32 pkts/s	26.66 %
192.168.4.44	TLAB1 (10.8.0.1)	1.17 MBytes	2.07 Kbps	11.74 %	11,279	15,121	3.36 pkts/s	16.83 %
TLAB7 (10.8.0.7)	192.168.10.55	777.42 KBytes	1.38 Kbps	7.83 %	11,241	12,086	2.69 pkts/s	13.45 %
192.168.10.18	TLAB4 (10.8.0.4)	593.23 KBytes	1.05 Kbps	5.97 %	9,845	13,025	2.89 pkts/s	14.50 %
192.168.10.55	TLAB7 (10.8.0.7)	278.88 KBytes	496 bps	2.81 %	6,540	6,304	1.40 pkts/s	7.02 %
10.8.0.51	192.168.14.16	244.44 KBytes	435 bps	2.46 %	4,528	528	0.12 pkts/s	0.59 %
10.8.0.99	192.168.14.16	41.23 KBytes	73 bps	0.41 %	1,125	301	0.07 pkts/s	0.34 %
192.168.14.16	10.8.0.99	38.05 KBytes	68 bps	0.38 %	825	257	0.06 pkts/s	0.29 %
192.168.14.16	10.8.0.51	36.90 KBytes	66 bps	0.37 %	818	513	0.11 pkts/s	0.57 %

Figure 1.9: Results Report. [2]

1.5.2 Case: Poor performances

This case comes from CA Technologies [2] as well as the figures shown below.
In this case, the issue is the following:

> ”Users in Singapore are complaining about poor performance when accessing critical business applications in the Houston Data Center. The network manager has suggested a 120,000 dollar bandwidth upgrade to fix the problem; however, the network engineer is not confident that this will resolve the issue. The network engineer uses ReporterAnalyzer to understand the cause of the poor performance.” [2]

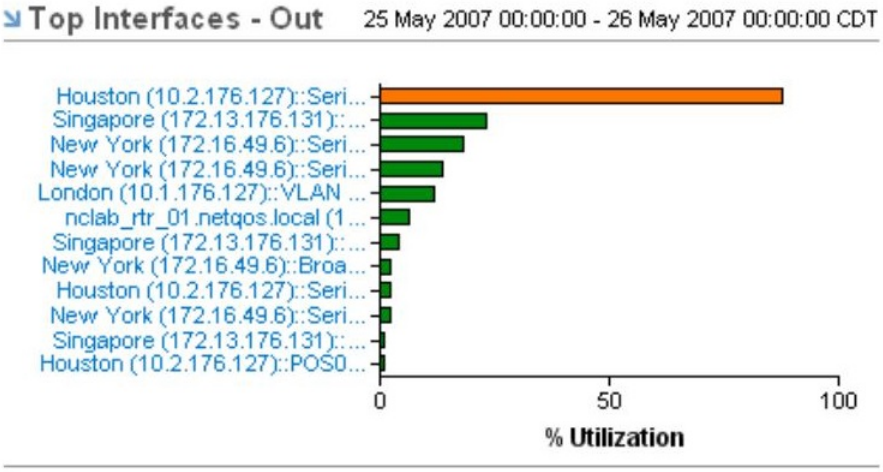
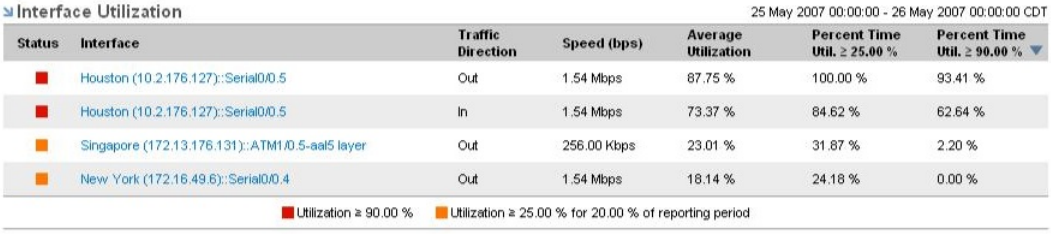


Figure 1.10: Top Out Report. [2]



Interface Utilization

25 May 2007 00:00:00 - 26 May 2007 00:00:00 CDT

Status	Interface	Traffic Direction	Speed (bps)	Average Utilization	Percent Time Util. ≥ 25.00 %	Percent Time Util. ≥ 90.00 % ▼
🟥	Houston (10.2.176.127)::Serial0/0.5	Out	1.54 Mbps	87.75 %	100.00 %	93.41 %
🟥	Houston (10.2.176.127)::Serial0/0.5	In	1.54 Mbps	73.37 %	84.62 %	62.64 %
🟧	Singapore (172.13.176.131)::ATM1/0.5-aal5 layer	Out	256.00 Kbps	23.01 %	31.87 %	2.20 %
🟧	New York (172.16.49.6)::Serial0/0.4	Out	1.54 Mbps	18.14 %	24.18 %	0.00 %

🟥 Utilization ≥ 90.00 % 🟧 Utilization ≥ 25.00 % for 20.00 % of reporting period

Figure 1.11: Utilization Report. [2]

Figure 1.10 presents the utilization (in %) of the Outs on a bar graph, it is visible that the network in Houston is used too much. It is also visible in Figure 1.11, which displays In and Out utilization in excess (over 90% or over 25% for 20% of reporting period), that Houston In and Out reaches an average utilization over 90%, that is not adequate.

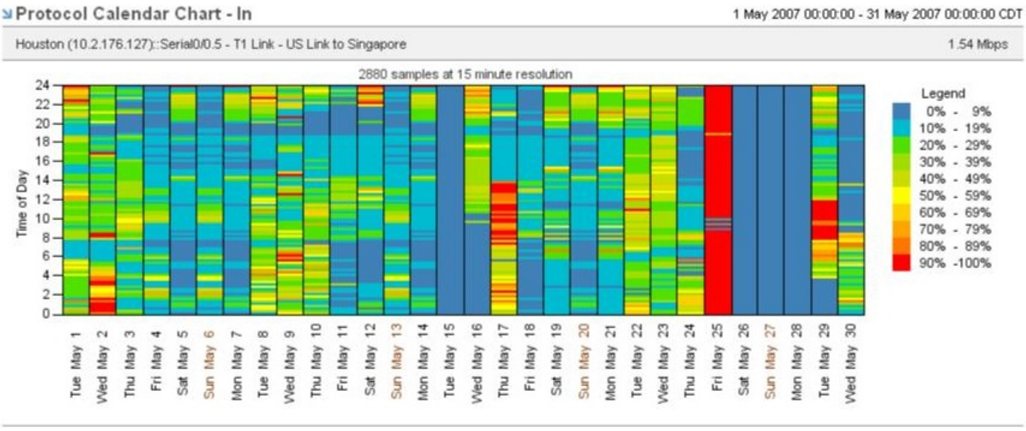
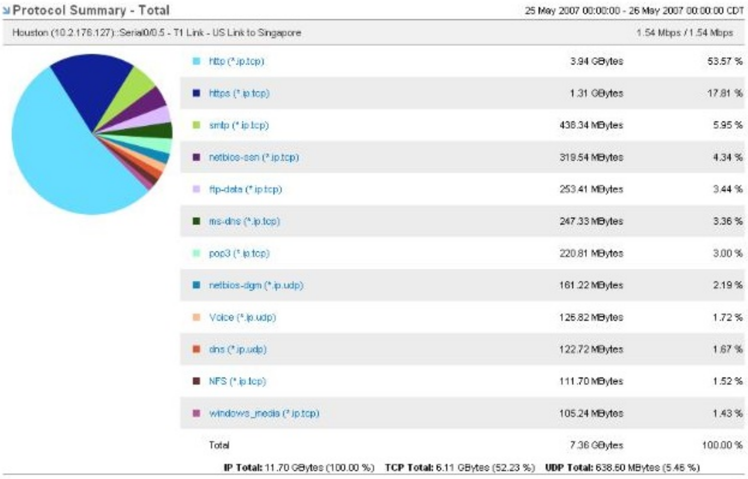


Figure 1.12: Protocol Calendar Chart. [2]

Another aspect of the IPFIX interface allows to see the data on a calendar chart like in Figure 1.12. Utilization is made clear for each day and hour for each month. Here, it can be spot an unusual activity on the 25 May where the utilization goes beyond 90%.

In this case as well, the engineer can go deeper into the data and spot the protocol usages that are unusual. As shown in Figure 1.13 (similar in style to 1.8), the http protocol asks for further investigation.

The interface allows to access a different view of the traffic like Figure 1.14, with this graph form is easily visible the behavior of the http traffic compared to a baseline, In on top, and Out down. Here the http traffic Out present unusual peaks. Those peaks are



Protocol Summary - Total 25 May 2007 00:00:00 - 26 May 2007 00:00:00 CDT

Houston (10.2.176.127)::Serial0/0.5 - T1 Link - US Link to Singapore 1.54 Mbps / 1.54 Mbps

http (*.ip.tcp)	3.94 GBytes	53.57 %
https (*.ip.tcp)	1.31 GBytes	17.81 %
smtp (*.ip.tcp)	438.34 MBytes	5.95 %
netbios-ssn (*.ip.tcp)	319.54 MBytes	4.34 %
ftp-data (*.ip.tcp)	253.41 MBytes	3.44 %
ms-dns (*.ip.tcp)	247.33 MBytes	3.36 %
pop3 (*.ip.tcp)	220.81 MBytes	3.00 %
netbios-dgm (*.ip.udp)	161.22 MBytes	2.19 %
Voice (*.ip.udp)	126.82 MBytes	1.72 %
dns (*.ip.udp)	122.72 MBytes	1.67 %
NFS (*.ip.tcp)	111.70 MBytes	1.52 %
windows_media (*.ip.tcp)	105.24 MBytes	1.43 %
Total	7.36 GBytes	100.00 %

IP Total: 11.70 GBytes (100.00 %) **TCP Total:** 6.11 GBytes (52.23 %) **UDP Total:** 638.60 MBytes (5.46 %)

Figure 1.13: Protocol Report. [2]

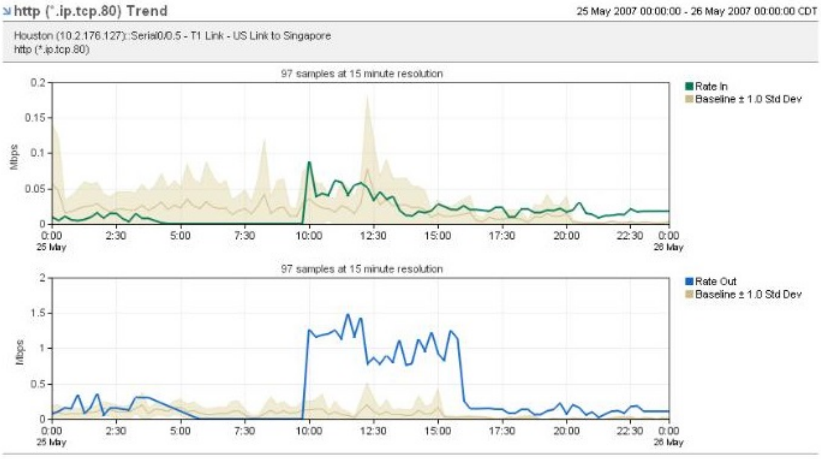


Figure 1.14: 'http' Report. [2]

explained in the Figure 1.15, that the reason for this use of the http protocol is the US Web Proxy Server that is using the most this host (9̃0%).

Later on the engineer could find out that the reason on the issue was that Singapore's users changed the internet proxy to US servers (in order to, presumably, get faster internet access).

A clear advantage of IPFIX is that it allowed to fix the issue and to avoid the department to spend money on material that is not needed.

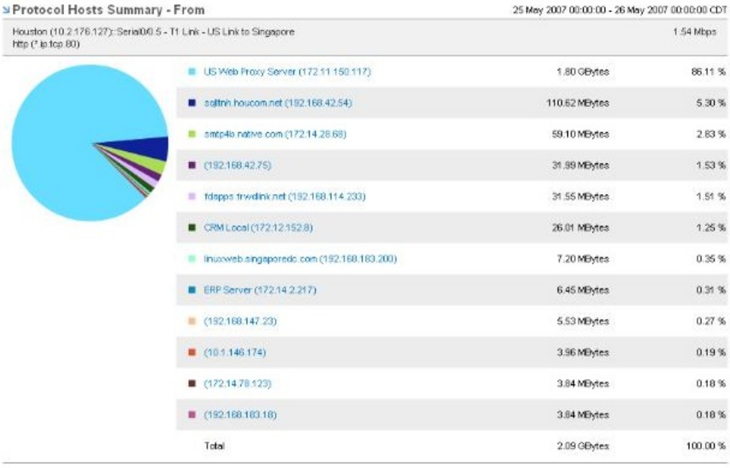
1.5.3 Business' benefits of IPFIX

There are several benefits inherent in using IPFIX.

IPFIX is good to have a system of traffic sources association, which allows for optimization of the network (finding bottlenecks, fixing inefficiencies) and to make investment on the network infrastructure. For this use, IPFIX allows to work with precise numbers and may allow to save money while still being able to build a good network system.

Another benefits of IPFIX is its capacity to work with real time information. This is useful to perform a constant following of the performances to keep track of the activity on the network. Yet, it also allows to solve more rapidly the problem that may occur, by using the IPFIX interface which allows a quick access to the required information.

A different benefit of IPFIX is that it is implemented within the network, so it does not need for a heavy added infrastructure like probes. This makes it more scalable and



Protocol Hosts Summary - From 25 May 2007 00:00:00 - 26 May 2007 00:00:00 CDT

Houston (10.2.176.127): Serial0/0.5 - T1 Link - US Link to Singapore 1.54 Mbps
http (* ip tcp 80)

US Web Proxy Server (172.11.150.117)	1.80 GBytes	86.11 %
sqltnh.houcom.net (192.168.42.54)	110.62 MBytes	5.30 %
smtp4b.native.com (172.14.28.68)	59.10 MBytes	2.83 %
(192.168.42.75)	31.99 MBytes	1.53 %
fdapps.trwdlink.net (192.168.114.233)	31.55 MBytes	1.51 %
CRM Local (172.12.152.8)	26.01 MBytes	1.25 %
linuxweb.singaporedc.com (192.168.183.200)	7.20 MBytes	0.35 %
ERP Server (172.14.2.217)	6.45 MBytes	0.31 %
(192.168.147.23)	5.53 MBytes	0.27 %
(10.1.146.174)	3.96 MBytes	0.19 %
(172.14.78.123)	3.84 MBytes	0.18 %
(192.168.183.18)	3.84 MBytes	0.18 %
Total	2.09 GBytes	100.00 %

Figure 1.15: Hosts Report. [2]

cheaper as there is no maintenance cost or heavy investment. Moreover it allows to gather data that are more ”complete” compared to other systems, due to the implementation of IPFIX on every network gear, so that there is no undefined area.

1.6 Conclusion

This paper highlighted several aspects of IPFIX, such as it being built on a previous Cisco work, NetFlow v9. IPFIX covers the important improvement on NetFlow of the extensible and customizable Information Elements, which leads to the freedom of the field of application, going beyond the area of network analysis and supervision.

The main purpose of IPFIX were presented, they are its ability to measure the traffic network (in real time) and to give more flexibility for the network administrator. To finish, it is also indicated that IPFIX is not only interesting from a technical point of view, but it also allows for very powerful usage in the real world.

In a few words, though IPFIX is technically interesting, the more fascinating about it are the possibilities that it offers. The increased usability and functionalities compared to other solutions are great assets that make it a very interesting tool to use in complement with the Cisco products.

Bibliography

[1] Amp 32: Overview of NetFlow Data Export process including exporter, collector, storage, and analysis workstation; http://commons.wikimedia.org/wiki/File:NetFlow_Architecture_2012.png, December 2014.

[2] CA Technologies; http://www.ca.com/~/media/files/whitepapers/netflow-v2_233048.pdf, September 2014.

[3] B. Claise, Ed., ”Cisco Systems NetFlow Services Export Version 9”, IETF, RFC 3954; http://tools.ietf.org/html/rfc3954, October 2004.

[4] B. Claise, Ed., ”Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information”, IETF, RFC 5101; http://tools.ietf.org/html/rfc5101, January 2008.

[5] B. Claise, Ed., B. Trammell, Ed., ”Information Model for IP Flow Information Export (IPFIX)”, IETF, RFC 7012; http://tools.ietf.org/html/rfc7012, September 2013.

[6] B. Claise, Ed., B. Trammell, Ed., P. Aitken, ”Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information”, IETF, RFC 7011; http://tools.ietf.org/html/rfc7011, September 2013.

[7] Cisco: NetFlow Version 9 Flow-Record Format; http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html, May 2011.

[8] Brad Hale: Geek’s Guide to the NetFlow v9 Datagram; PDF http://web.swcdn.net/creative/pdf/Whitepapers/NetFlow_Datagram_-_Final.pdf, 2012.

[9] S. Leinen, ”Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX)”, IETF, RFC 3955; http://tools.ietf.org/html/rfc3955, October 2004.

[10] J. Quittek, S. Bryant, B. Claise, P. Aitken, J. Meyer, ”Information Model for IP Flow Information Export”, IETF, RFC 5102; http://tools.ietf.org/html/rfc5102, January 2008.

[11] J. Quittek, T. Zseby, B. Claise, S. Zander, ”Requirements for IP Flow Information Export (IPFIX)”, IETF, RFC 3917; http://tools.ietf.org/html/rfc3917, October 2004.

[12] Matthew Robertson: NetFlow Monitoring for Cyber Threat Defense; http://www.slideshare.net/CiscoCanada/net-flow-monitoringforcyberthreatdefensematthewrobertson, April 2014, 17.

[13] G. Sadasivan, N. Brownlee, B. Claise, J. Quittek, ”Architecture for IP Flow Information Export”, IETF, RFC 5470; `http://tools.ietf.org/html/rfc5470`, March 2009.

[14] Corinna Schmitt: Secure Data Transmission in Wireless Sensor Networks, Dissertation, Series ”Network Architectures and Services” (NET), Chair for Network Architectures and Services, Technische Universitaet Muenchen; July 2013.

[15] Brian Trammell, Benoit Claise: Applying IPFIX to Network Measurement and Management; Presentation at IETF 87, Berlin, July 2013; `http://www.ietf.org/edu/tutorials/ipfix-tutorial.pdf`, November 2014.

Chapter 2

Feasibility of Multisig in CoinBlesk

Andreas Albrecht, Adrian Gasparini

Bitcoin is a digital payment system that works without any centralized authorities or services. It is built as a fully distributed peer-to-peer (P2P) system which is open to everyone. CoinBlesk is an online Bitcoin wallet service targeting mobile devices such as smartphones and tablets. It enables instant payments in Bitcoins between users over near field communication (NFC). In general, the technical foundation of Bitcoin is able to cope with the standard use case of a payment system. However, users face many challenges which are increasingly important now that Bitcoin has become much more popular. One crucial aspect is the safety of Bitcoins because many users and services were and still are targets of criminals, victims of fraud or have lost money due to missing wallet backups or theft. The multisig concept is a technical answer to these issues. While traditional Bitcoin transactions need to be signed by one private key, multisig enables the use of more than one private key associated with a transaction. For instance, it is possible to create an address associated with three private keys out of which two are required to spend money. As a consequence, a lost key does not lead to loss of money and the level of security is increased because more than one key is required. This paper discusses the feasibility of the multisig concept in CoinBlesk.

Contents

2.1 Introduction and Problem Statement

Bitcoin [37, 7] is a digital payment system that works without any centralized authorities or services. It is built as a fully distributed peer-to-peer (P2P) system which is open to everyone. Bitcoin has received much attention in the past as it is the first implementation of a fully distributed cash system with economic success. The current market capitalization is approximately 4.6 billion USD [16, 23], which reflects that people are willing to pay fiat money such as USD in exchange for BTC.

A downside of the economic success is that it is attractive for criminals to attack users and steal Bitcoins from them accordingly or to commit fraud. Even though Bitcoin is a public system, it is difficult to prosecute such incidents due to the non-reversibility of transactions and the pseudonymity of users in the network. Hence, a crucial part of Bitcoin and probably its future is the security of the user's assets in particular if Bitcoin is used as a replacement for traditional currency and payment systems such as credit cards or PayPal.

2.1.1 Motivation

Multisig [8, 2, 18, 19] is a concept in Bitcoin that enables additional security features. Users sign their regular Bitcoin transactions with their private key to spend coins. On the other hand, multisig transactions require one or more signatures, i.e. they need to be signed with one or more private keys. For instance, it is possible to specify that two signatures of private keys out of three are required to create a valid transaction and hence, to spend money. This is very powerful and flexible and allows implementing many additional features such as contracts or access restrictions.

A key feature of multisig is that it allows implementing customer protection by means of escrow services in a distributed and secure fashion. Traditional payment systems like credit card system or PayPal [40] allow customers to file for a refund if they do not agree with a transaction. Since transactions are non-reversible in Bitcoin, the receiving party has - once a transaction is issued - more power compared with the sender in a case of a dispute. Since there is no central authority that can mediate between disagreeing payer and payee, technical countermeasures are required.

Furthermore, multisig can be used to distribute control over money by distributing keys either by storing them at different places or by giving them to multiple people.

2.1.2 Feasibility of Multisig in CoinBlesk

CoinBlesk [22] is an online Bitcoin wallet service targeting mobile devices such as smartphones and tablets. It is developed by students and staff members of the Communication Systems Research Group (CSG) of the University of Zurich. It enables instant payments in Bitcoins between users over near field communication (NFC), i.e. by holding mobile devices together.

To achieve this, CoinBlesk does not simply rely on the Bitcoin network and its public ledger, but introduces a centrally managed service that operates as a trusted mediator between payer and payee and keeps track of transactions. As a result, the service needs to be in full control of the assets of its users and executes payments between them on demand. This avoids that users can spend money without the service noticing it. A consequence of this design is that users must fully trust the CoinBlesk service operator as they give up full control over their Bitcoins. This is very similar to a bank in traditional payment systems.

This paper investigates how multisig can mitigate these issues and analyzes how the multisig concept can be integrated into the CoinBlesk architecture. Furthermore, the legal situation of Bitcoin services and CoinBlesk in Switzerland is described followed by a discussion of how multisig addresses law and regulation issues (e.g. banking license).

2.1.3 Outline

The remaining part of this paper is organized as follows. Section 2.2 gives a short introduction to Bitcoin basics followed by a presentation of CoinBlesk and its design in Section 2.3. Multiple multisig techniques are introduced in Section 2.4. In particular, multisig as implemented in the Bitcoin protocol and alternatives are studied. The integration of multisig in CoinBlesk and its implications are described in Section 2.5. Related work is discussed in Section 2.6. The focus is on techniques used in Bitcoin payment solutions, wallet services and concepts that use multisig or offer similar functionality as CoinBlesk. Finally, the summary and conclusions follow in Section 2.7.

2.2 Bitcoin

Bitcoin is a digital payment system that works without any centralized authorities or services. It is built as a fully distributed peer-to-peer (P2P) system which is open to everyone.

Participating peers have one or more associated Bitcoin addresses which they can use to receive and send Bitcoins (BTC). Note that the term *Bitcoin* refers to both, the system as well as the unit of the currency. Transactions of users are public and broadcasted within the network. As a result, all peers can keep track of all payments and there is no need to trust single entities. The nodes collectively build a distributed public ledger called blockchain that includes all transactions ever made.

The blockchain records all payment activity between users by including new transactions in blocks that are chained together, which orders the transactions by time. Hence, the network agrees on a unique history, which prevents that users can cheat and, for instance, spend the same Bitcoin multiple times. The process of building the public ledger is called *mining* and is done by so-called miners. Miners collect broadcasted transactions and add them to a set of transactions called block. They then try to solve a difficult time and resource consuming cryptographic puzzle. Solving the puzzle takes approximately 10 minutes for all miners of the network. Once a miner finds the solution to the puzzle, he is rewarded with a certain amount of Bitcoins. Furthermore, the corresponding block is appended at the end of the blockchain.

Due to the decentralized nature and the unique history, transactions cannot be reversed once they are acknowledged by the network and included in the ledger. This is because it would take too much time and computational power to rebuild the blockchain.

Cryptography is a fundamental part of Bitcoin. In particular, public key cryptography is used to send and receive money. Each user has at least one address that is derived from his public key. To spend money, the sender creates a transaction that transfers money to the address of the receiver. In addition, he signs the transaction with the private key that corresponds to the address where the money to spend was received in the past.

2.3 CoinBlesk

CoinBlesk is a mobile Bitcoin payment system (MBPS) that is developed by students and staff members of the Communication Systems Research Group (CSG) of the University of Zurich. It enables instant Bitcoin money exchange between two parties over NFC (near field communication) and supports two typical use cases. First, it allows merchants to request money from customers at a point of sale (POS). Second, it allows sending money between users directly in a peer-to-peer (P2P) fashion. The use cases mainly differ in two aspects: the direction of money and the party who initiates the transaction and specifies the amount of money to transfer.

In the remaining part of this section CoinBlesk and its motivation is presented in Section 2.3.1 followed by a discussion of the architecture in Section 2.3.2. The focus of Section 2.3.3 is on open issues of CoinBlesk. In particular, payment authorization issues are outlined as well as legal questions that play a role if Bitcoin services are offered in Switzerland.

2.3.1 CoinBlesk Overview

Bitcoin is fast compared with traditional bank wire transfers, which typically take at least a day to be processed. However, Bitcoin is slow in comparison with instant credit card payments. This is because Bitcoin transactions need to be verified by the network in a distributed fashion. It takes approximately 10 minutes on average for a transaction to be confirmed once in a newly mined block in the blockchain. If six confirmations are requested for a reliable verification, it already takes about an hour. For some situations, like everyday payments in a shop or restaurant, this is far too long because waiting times are not acceptable in these situations. Sellers can lower the number of confirmations to, for instance, one or even zero. This avoids waiting time by the cost of increased risk to lose money due to failed transactions which will never be confirmed by the network.

Users who pay with Bitcoin usually pay a fee for each transaction. The fee is an incentive for the nodes in the Bitcoin network to process the transaction. In particular, a miner creating a new block collects all fees of the transaction included. For small payments the fee may become a substantial part of the total amount to pay. Hence, it is desirable to avoid transaction fees for these payments. Today, this is not a huge issue because the fees are not very high, but they may increase in the future.

A further problem with Bitcoin is that both, the customer and the merchant need to be connected to the Internet to send and receive transactions form the Bitcoin network. This is problematic for roaming users without Internet connection abroad, as well as for points of sale in areas with bad mobile Internet coverage, e.g. in buildings.

CoinBlesk addresses these issues by introducing a centralized service. The next section presents the CoinBlesk architecture and discusses how it tries to eliminate the aforementioned shortcomings of Bitcoin.

2.3.2 CoinBlesk Architecture

The CoinBlesk architecture consists of a server and a client application, which is installed on the devices of users, i.e. on smartphones and tablets of merchants and customers. Figure 2.1 shows an overview of CoinBlesk's architecture [31]. Currently, the client application is available for Android devices only running Android 4.4+.

Using the CoinBlesk app, a user creates a Bitcoin address, whose corresponding private key is stored on the CoinBlesk server. Then, the user can pay in an amount of Bitcoins to his/her address.

In order to make a deal between two parties, their client devices communicate with each other via NFC. The customer's device issues a request to pay the negotiated amount, signs it with the customer's private key (note: this is *not* a private key corresponding to a Bitcoin address) and sends it to the merchant's device (via NFC). The merchant's device forwards the signed payment request to the CoinBlesk server (via a RESTful service). The server verifies the signature of the payment request. As the server is in control of the users' Bitcoin private keys, users cannot spend money without the server knowing it. The server thus keeps track of the users' account balances and can confirm payments between them instantly (as there is no need to wait for confirmations by the Bitcoin network).

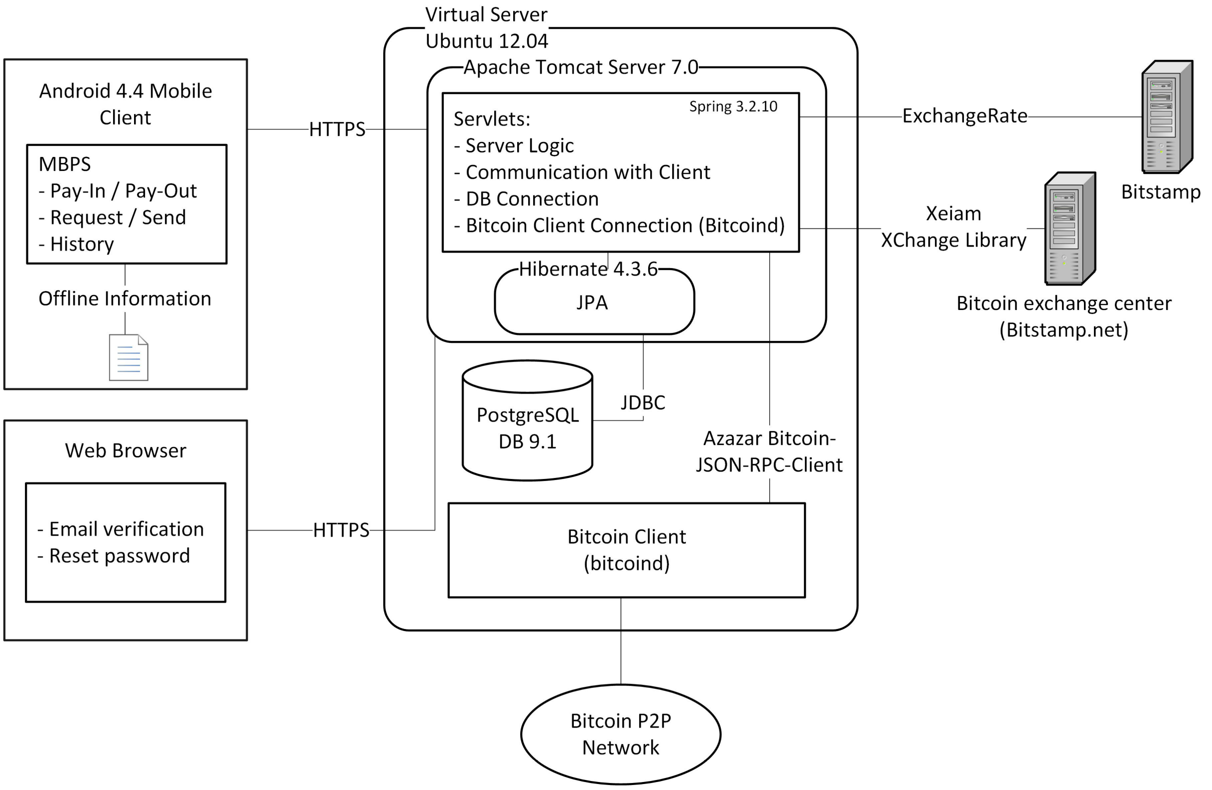


Figure 2.1: CoinBlesk Architecture [31]

There is no need to actually issue Bitcoin transactions until a user wants to pay out his account balance to a Bitcoin address that is not controlled by CoinBlesk. As the account balance of a user may be higher than the amount associated with the user's Bitcoin address (such as in the case of a merchant who has received payments from customers), CoinBlesk takes the required amount of bitcoins as transaction inputs from some of the Bitcoin addresses controlled by CoinBlesk.

2.3.3 CoinBlesk Issues

In the following subsections security issues and the legal situation of Bitcoin services are discussed.

2.3.3.1 Security

A consequence of the centralized design is that users must fully trust the CoinBlesk service operator as they give up full control over their Bitcoins. If the CoinBlesk server fails, the Bitcoins are no longer accessible and may even be lost forever (due to loss of the private keys). Furthermore, if the CoinBlesk server gets compromised, the intruder is in control of the Bitcoins (i.e. key theft). In addition, users are always in a weaker position and CoinBlesk can hold the user's money hostage and deny payments.

2.3.3.2 Legal Situation of Bitcoin Services

Besides the technical view there is also the legal situation that needs to be looked at. Does controlling the users' Bitcoin balance require a license and which regulations may apply? This depends on the country where the service operates. Thus, the focus is on the situation of Switzerland.
In Switzerland there are no Bitcoin-specific laws in place and payments in Bitcoin are legal and unregulated [46, 45] according to the Swiss Financial Market Supervisory Authority

(FINMA). Nevertheless, regulations and laws may apply for services that manage user accounts. The Swiss parliament is aware of Bitcoin and has submitted two postulates regarding Bitcoin [48, 49], to which the Swiss Federal Council has responded with a report [25]. In this report the following is stated regarding the Banking Act:
”Only banks are allowed to accept deposits from the public on a professional basis. Individuals or legal persons who intend to accept deposits from third parties on a professional basis must obtain a banking licence before commencing their activities. [...], an entity acts on a professional basis if it accepts deposits on a permanent basis from more than 20 people or if it advertises such services in any form, [...].”
There are some exceptions to this rule. A banking license is not required if the payment system is only used to acquire goods and services. This holds if no interests are paid and the account balance of each customer is below 3000 CHF. CoinBlesk is intended for the acquisition of goods or services. However, CoinBlesk does not preclude the transfer of money for other purposes. So, it appears questionable if this exception applies. In particular, as users also can send each other money using the CoinBlesk platform which indicates that CoinBlesk may be considered as a sort of trading platform.
A banking license is further required if there is no other institution such as a bank that guarantees the user assets. However, if users have access to their account at all time without participation of the service, a license is not required. Since the private keys are managed by CoinBlesk, this does not apply at the moment.
Furthermore, FINMA states that anti money laundering regulations apply if a service allows purchasing and selling Bitcoins. These rules do not apply as long as CoinBlesk does not allow exchanging (i.e. sell and buy) Bitcoins directly but relies on other existing services such as Bitcoin ATMs to deposit user funds. These regulations further apply for trading services.

2.4 Multisig

In order to spend Bitcoins received to an address, users have to sign the spending transaction with the corresponding private key. Thus, as long as the private key is kept in a safe and secure place, the funds sent to the particular address are safe. However, as soon as the private key is compromised the funds are not secure anymore. Since the private key is the only way to grant access to coins, lost private keys implicate lost coins. In the past, many users lost Bitcoins due to lost private keys, fraud or theft [34, 29, 13]. There are several approaches that tackle these issues and try to secure and protect Bitcoin accounts. Many of them use common practices that are known from other online services.
First, backups of Bitcoin wallets including private keys are very crucial and an easy way to protect against key loss. Furthermore, hierarchical deterministic wallets [52] simplify key management and backups as addresses and private keys are generated deterministically given a seed and hence, only the seed (e.g. a mnemonic) needs to be protected instead of a set of private keys. This eliminates the need of continuous and repeated backups since all private keys are derived from a seed.
Second, centralized services such as Coinbase [21], BitGo [14] or Blockchain.info [17] that are maintained by a service provider and possibly secured by security experts are another option. Many users may think that these systems are more secure than smartphones or home computers due to advertised security and audits. These services may help to protect against attackers that try to steal private keys on user devices. In addition, centralized services often offer additional security measures such as two-factor authentication. However, many users do not trust centralized services because they have full access to the user funds and they probably do not guarantee full protection. In particular, inside at-

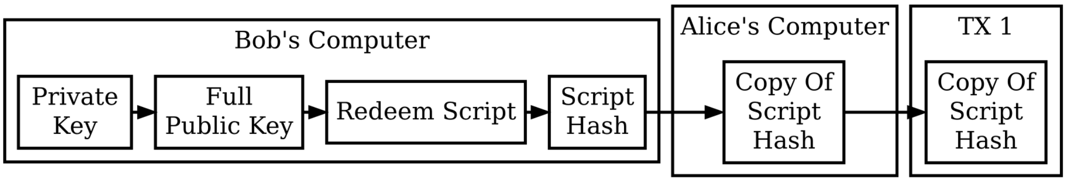


Figure 2.2: Creating redeem script and receiving Bitcoins to script hash. [8]

tacks, social engineering attacks, phishing and other attacks are still feasible and common [34, 13, 33].

Third, there are offline wallets [5, 24], paper wallets [38, 6] or hardware wallets [50, 41] that do not connect to the Internet. They store the private key in a hardware token, on an offline computer or on paper, which reduces attack vectors that require an Internet connection (e.g. malware distributed using mail).

Even though these techniques help to protect Bitcoin funds, they still have flaws. All these approaches have in common that they eventually reveal or grant access to the private key. Hence, the private key represents a single point of failure or attack point. Multisig is an approach that eliminates this. In contrast to regular Bitcoin addresses and transactions, multisig addresses require multiple keys and possibly multiple signatures to be valid.

In the remaining part of this section, multiple multisig approaches are introduced. Two implementation types are discussed: multisig inside the Bitcoin protocol in Section 2.4.1 and alternatives in Section 2.4.2 that are more general and outside of the Bitcoin protocol.

2.4.1 P2SH Multisig

The motivation of pay-to-script-hash (P2SH) [3] is that the sender of coins usually only wants to simply send money, but has little interest in future transactions of the receiver or security measures at the receiver [8]. The receiver, however, is interested in the security of the coins and the conditions for the next spending transaction. Bitcoin introduced a simple scripting language and a technique called P2SH to achieve this, which allows receivers to define conditions in a redeem script that must be met in order to spend money. As the name suggests, payments go to a hash of a script, which means that P2SH uses another addressing scheme in which addresses are derived from a script. Regular addresses, on the other hand, are derived directly from a public key.

Section 2.4.1.1 first shows how P2SH transactions work in general including the new address scheme as they are a means to implement multisig in Bitcoin, which is discussed in Section 2.4.1.2.

2.4.1.1 P2SH Addresses and Transactions

To illustrate P2SH, the following situation based on [8] is considered in which Alice as the sender wants to give Bitcoins to the receiver Bob, who further spends the received money. P2SH consists of two parts: receiving payments and spending funds. Figure 2.2 depicts the first part. Bob needs to give his Bitcoin address to Alice as she needs to include this information in the transaction. Thus, Bob generates a P2SH address [4] as follows. He creates a redeem script using the scripting language. Next, he hashes the script, which results in his Bitcoin address where Alice can send Bitcoins to. Bob does not discard the redeem script as he needs it to spend received coins in the future. Alice creates a transaction that transfers some Bitcoins to the hash of the redeem script. She does not need to possess the actual redeem script in its serialized form. Bob receives the coins eventually once the transaction is broadcasted and confirmed.

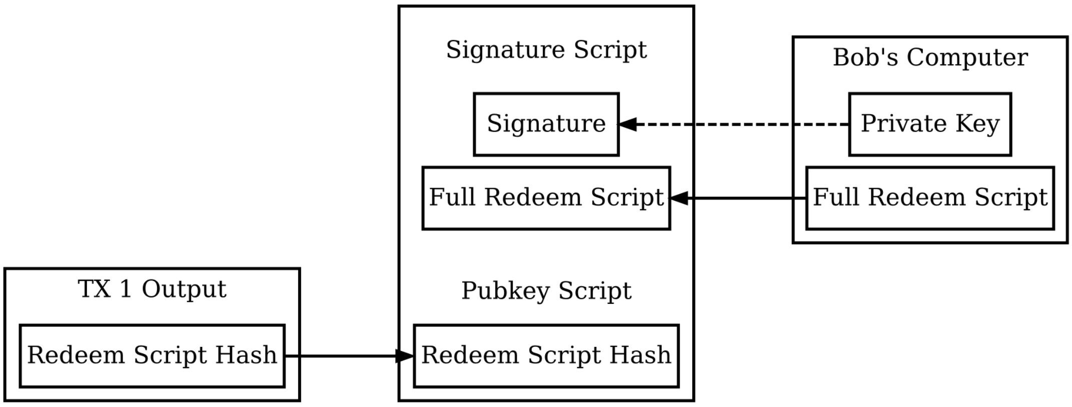


Figure 2.3: Spending Bitcoins paid to a script hash. [8]

In order to spend money sent to a script hash as illustrated in Figure 2.3, Bob creates a transaction as well and a corresponding signature script which includes the redeem script in its serialized form. Furthermore, he signs the transaction with his private key. Miners who validate the transaction process it as follows. First, they check that the hash of the redeem script in Bobs transaction matches the hash included in Alice's transaction. This ensures that Bob provided the same redeem script which was used to generate the Bitcoin address. Second, the script is executed. It may evaluate to true or false and the transaction is only considered valid and included in a block in the former case. The evaluation ensures that the conditions specified in the script are indeed satisfied.

Another simple approach to achieve the aforementioned goal of specifying spending conditions would be to give the script in a serialized form directly to Alice who then could include the script in the transaction. However, this solution has shortcomings because it would require more knowledge at the sender. It would reveal the script content to the sender and would require more data to be transferred. Eventually, however, the script is published as Bob spends the coins. In contrast, the hash of a script is small in size such that it even fits in a QR code, which is in particular beneficial for mobile devices.

2.4.1.2 M-of-N Multisig using P2SH

P2SH allows implementing multisig in Bitcoin [8, 9, 11, 2, 3] by putting the condition that multiple signatures are required into the redeem script. An M-of-N scheme is used where a threshold of M signatures out of N are required for the script to evaluate to true. In general, an M-of-N multisig redeem script looks as follows:

```
OP_M [pubkey 1] [pubkey 2] ... [pubkey N] OP_N OP_CHECKMULTISIG
```

`OP_M` specifies the number of required signatures and `OP_N` the total number of included public keys in the script. In theory `M` and `N` are integers between 1 and 15. In practice they are limited because size limits of scripts apply for standard transactions. Currently, `M`+`N` should not exceed 7 if uncompressed keys are used. The instruction `OP_CHECKMULTISIG` tells others how they should process the signature script respectively verify the transaction. To spend coins, a signature script has to be created which looks as follows:

```
OP_0 [signature x] ... [signature y] [serialized redeem script]
```

It consists of a list of signatures as demanded by the specified threshold `M` of required signatures and the redeem script itself in its serialized form. `OP_0` is a workaround for an implemented off-by-one bug [8].

As a transaction is broadcasted and received by other peers, they validate it by processing the signature script in reverse order from right to left (because the statements are pushed onto a stack). The first statement is `OP_CHECKMULTISIG` and tells a peer to read an integer

N followed by that many public keys. Next, the threshold M is consumed followed by that many signatures. Given the public keys, the peer can verify the signatures and return true or false, which makes a transaction either valid or invalid.

2.4.1.3 Multisig Use in Bitcoin

Some already marked the year 2014 as ”the year of multisig” [32] and predict that multisig is ”the future of Bitcoin” [18]. Since transactions are public, it is possible to investigate how popular multisig actually is at the moment and how the adoption evolved until now. According to [39] approximately 1.5% of all Bitcoins that currently exist are held in a P2SH account. This already shows that P2SH and multisig are not widely used today. The problem is probably that many wallets may be able to process transactions with P2SH outputs (i.e. sending to), but they do not offer the corresponding user interface to manage multisig accounts efficiently and in a convenient way (i.e. receiving to and sending from). However, wallets such as GreenAddress [27], Armory [5] and BitPay [15] start offering these techniques and if the technical multisig details become less visible due to an user interface, it can be expected that the adoption will increase in the future. Furthermore, many Bitcoin services are still in an early stage and may not offer the most recent features yet.

2.4.2 Alternatives

In the following subsections Shamir’s secret sharing and threshold signatures are discussed.

2.4.2.1 Shamir’s Secret Sharing

Shamir’s secret sharing [43] is a cryptographic technique that allows splitting a given secret into multiple parts. To reconstruct the initial secret, a subset of the parts is required. At the beginning when the individual parts are created, it is possible to define a minimum number of parts (threshold) that are required for reconstructing the secret. The individual parts do not leak any information about the secret and hence, the possession of any number of pieces below the threshold is useless for an adversary. The main idea is that the secret is split into parts which then are distributed among different devices, locations or users that jointly reconstruct and use the information at a later point.

The technique uses the fact that a polynomial of degree $k-1$ is uniquely defined by at least k points. For instance, it takes 3 points to define a unique parabola (degree 2) and there are infinitely many parabolas going through 2 points. Given this idea, the secret has to be represented by a curve of a certain degree $k-1$ where k is the desired threshold, i.e. the number of pieces that are needed to reconstruct the secret. Furthermore, there are infinitely many points on a curve, which means that every point on the curve represents the secret partially. Hence, to split a secret into n parts, n points on the curve have to be selected.

Shamir’s Secret Sharing and Bitcoin Shamir’s secret sharing can be used to split a wallet seed or a private key corresponding to a Bitcoin address into multiple pieces of information. As a consequence, multiple pieces are required to spend coins. It is similar to multisig in the sense that multiple parts of a private key are requires to sign a transaction. However, the transaction is signed only once as the private key is reconstructed. Thus, it is not multisig, but can be seen as an alternative in certain use cases where the flexible and powerful features of P2SH multisig are not required. Even though the private key can be discarded after the splitting step, the technique is not guaranteed to be safe because the secret information is available on a machine at least twice: once at the beginning in order to create the pieces and every time after reconstruction. This means that the key generation must be secured and that the machine must not be compromised during these steps since an adversary could steal the information accordingly. Due to these drawbacks, Shamir's secret sharing should be used with care and is safe for one-time use (e.g. private keys should be changed after each reconstruction). The advantage of this approach is that arbitrary data can be split and there is no support of others required to use this technique as it only depends on user's decision to use it and not on the Bitcoin protocol itself or miners.

Currently, Shamir's secret sharing is mostly used to protect wallet backups, passwords and wallet seeds or private keys. For instance, BitAddress.org [6] allows creating a split wallet which splits the private key into multiple pieces. Armory [5] offers a feature which they call fragmented backup that allows creating wallet backups that need multiple fragments to be reconstructed. Ryan Shea [44] implemented a Python module (`secretsharing`) that offers sharing Bitcoin private keys out of the box.

2.4.2.2 Threshold Signatures

Threshold signatures [30] are another technique to split control and its high level idea is very similar compared to Shamir's secret sharing. However, instead of splitting a secret such as a private key among participants, the power of signing is distributed. The meaning of this is that given a threshold t and n players each having a share of a private key, t or more players are required to jointly create a signature. Thus, the capability of executing a cryptographic operation is split rather than the information itself. This eliminates the reconstruction of the private key, which makes it possible to create a signature in a distributed fashion without ever revealing the private key to any participant during the signing procedure. Furthermore, no information is leaked over time even if multiple data are signed. This is a huge advantage compared with Shamir as there is no need to trust that others do not abuse the secret once revealed.

Threshold Signatures in Bitcoin Goldfeder et al. [26] demonstrated that threshold signatures are compatible with Bitcoin signatures and that there is no difference between a jointly signed transaction and a regular transaction. This makes threshold signatures an alternative to multisig using P2SH of the Bitcoin protocol. The advantage is that no Bitcoin support is required which leads to more flexibility. Furthermore, anonymity and confidentiality are increased because the possible signing participants are not published. In P2SH multisig, every public key is included in the redeem script, which reveals the security mechanisms implemented at the owner of the coins to some extent. Since threshold signatures are compatible with regular Bitcoin addresses and signatures, it is fully backward compatible and it scales in the same way as regular transactions do. P2SH transactions, on the other hand, tend to be bigger in size since the redeem script is included. Furthermore, the size limitations of standard transactions restrict the number of signing participants of P2SH multisig. A drawback of threshold signatures is that it operates in rounds which means that participants have to interact with each other continuously and they cannot sequentially sign a transaction independent from each other.

2.5 Multisig in CoinBlesk

As discussed in Section 2.3.3, there are security (i.e. key loss and key theft) as well as legal (i.e. possible obligation for a banking license) issues with CoinBlesk. This section discusses how multisig can mitigate these issues and describes how the multisig concept can be integrated into the CoinBlesk architecture. In general, multisig can improve the trust in the overall CoinBlesk solution by giving the user more control over his Bitcoins as transactions need to be signed by both, the CoinBlesk server as well as by the user.

These measures may also mitigate the legal issues to a certain extent, however it cannot be inferred from this that no banking license is required.

2.5.1 2-of-2 Multisig

With 2-of-2 multisig the server and the user both have a key and both have to express consent regarding a transaction. Since multiple keys and signing operations are required, several modifications of CoinBlesk are discussed starting with the address generation and how users can deposit money into a CoinBlesk account followed by the use cases *sending* and *receiving* money.

2.5.1.1 Address Generation and Bitcoin Deposit

Currently, the server manages an address for each user. If a user wants to deposit money, he just sends the desired amount to the address associated with his user profile by issuing a regular transaction from a wallet or using a Bitcoin ATM. As soon as the transaction is broadcasted and confirmed by the network, the balance is updated. This procedure works fine as long as the server manages the addresses and the association between an address and a user on its own. However, the process of depositing Bitcoins to a multisig address requires some modifications because the address is derived from multiple public keys one of which created by the user itself. As a result, the address has to be created with direct participation of the client device such that the user can contribute a public key to the redeem script. Figure 2.1 outlines the procedure and how the client is involved in the address generation. Each step is described further in detail.

(DEP-1) As a user registers a new account, the CoinBlesk client creates a new key pair that can be used for the user's multisig address. After key generation, the client requests the deposit address from the server. The request includes the public key to be included in the multisig address.

(DEP-2) The server creates a key pair as well and generates a new multisig address associated with the user. The address is derived from the public key of the user and the public key of the server. This results in a redeem script and a hash of the redeem script, which is the P2SH address. All information is stored on the server in the database. Finally, the server returns the address and the redeem script to the client.

(DEP-3) The client checks that (1) his public key is included in the redeem script and (2) that the multisig address is 2-of-2. Furthermore, the client verifies that the hash of the redeem script results in the address. This ensures that the private key can generate valid signatures for transactions containing the script and address.

(DEP-4) Once an address is associated with the user, he can send funds into the multisig address using either a Bitcoin ATM or issuing a regular transaction from his own wallet. In either way, the server eventually gets a notification from the Bitcoin network and updates the user's account balance accordingly. After this, the amount of money is bound to the multisig address and spending requires consent from the server and client.

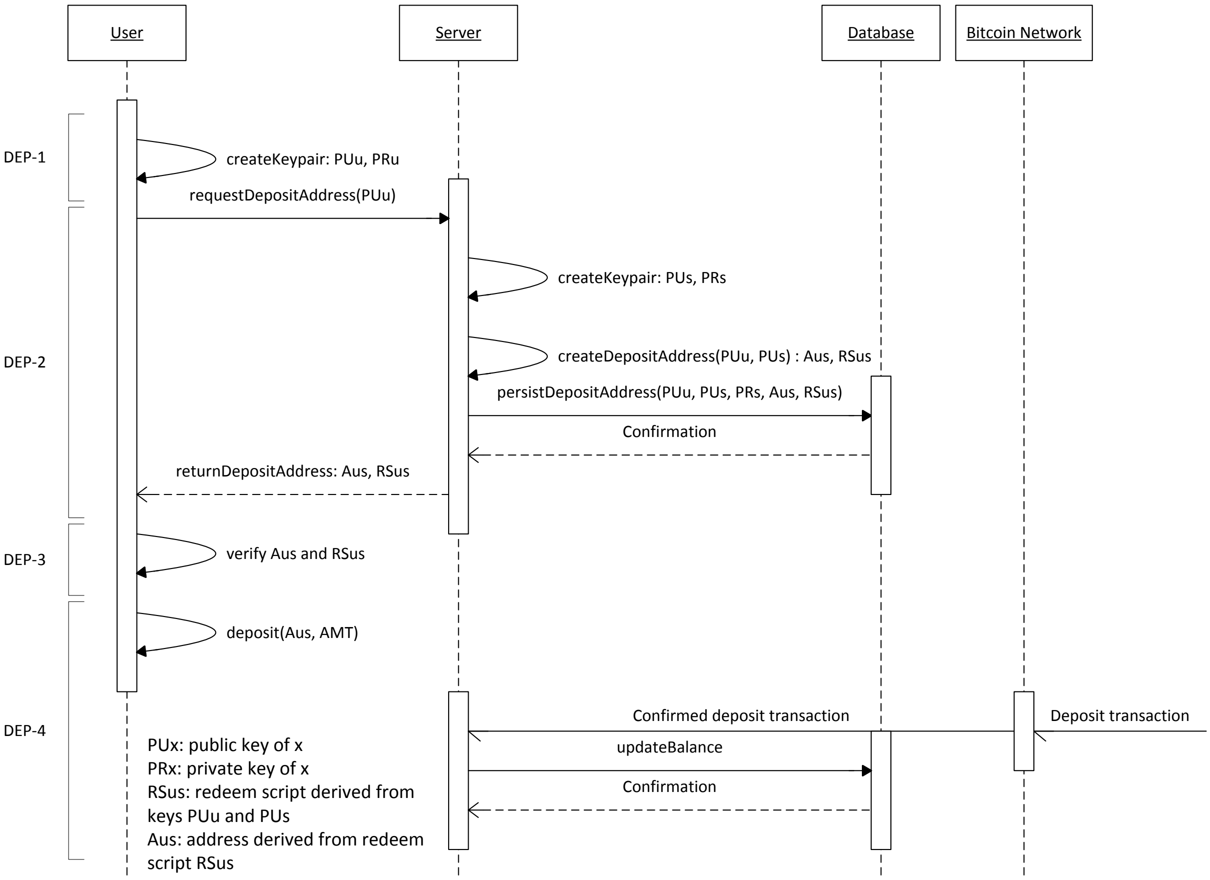

User
Server
Database
Bitcoin Network
DEP-1
DEP-2
DEP-3
DEP-4
createKeypair: PUu, PRu
requestDepositAddress(PUu)
createKeypair: PUs, PRs
createDepositAddress(PUu, PUs) : Aus, RSus
persistDepositAddress(PUu, PUs, PRs, Aus, RSus)
Confirmation
returnDepositAddress: Aus, RSus
verify Aus and RSus
deposit(Aus, AMT)
Confirmed deposit transaction
Deposit transaction
updateBalance
Confirmation
PUx: public key of x
PRx: private key of x
RSus: redeem script derived from keys PUu and PUs
Aus: address derived from redeem script RSus

Figure 2.4: Multisig address generation and Bitcoin deposit

2.5.1.2 Point of Sale Transaction (Request Money)

In the point of sale use case, the payee requests a payment from the payer. The payee specifies the amount and currency and the payer gives his consent to the given contract. With multisig, the protocol needs to be adapted because the Bitcoin transaction needs to be signed by the payer. The extension is based on the current design of CoinBlesk as described in [35]. An overview of the protocol is depicted in Figure 2.5 and the interactions are further discussed next.

(POS-1) The payee enters the amount that he wants to receive as well as the currency. To proceed, the devices of the payee and payer need to be tapped together.

(POS-2) In order to create a transaction, the payee and payer have to get to know each other's username and negotiate the payment details, which works as follows.
First, the payee sends a payment request to the payer. The request includes information such as the username, the amount requested and the currency.
Second, the payer confirms with a payment request as well if he agrees with the outlined contract. By sending back the payment request, he indicates that he is willing to create this transaction and pay the amount to the payee. The message includes the username of the payee, the username of the payer, the amount and the currency.
The current protocol uses only the payment requests of the payee and payer to create and execute a transaction. With multisig, this does not suffice as a Bitcoin transaction needs to be signed eventually. The requests, however, are used to inform the server about the ongoing transaction.

(POS-3) The requests are forwarded to the server who checks that the requests match by comparing the usernames of payee and payer, the amount, etc. In addition, the server verifies that the amount to transfer does not exceed the account balance of the payer.
Next, the server prepares a Bitcoin transaction that transfers the specified amount in BTC from the payer's multisig address to the payee's multisig address. The transaction is not signed by the server yet because it needs to be in control of the final steps of the process such that it can keep track of the account balances and confirm the payment immediately without the risk of double spending. Finally, the unsigned transaction is forwarded from the server via payee to the payer.

(POS-4) The payer checks the amount of the transaction and signs it with his private key. To finalize the transaction, the partially signed Bitcoin transaction is sent back to the server.

(POS-5) The server verifies the signature of the transaction using the payer's public key and checks that the transaction itself was not modified. Next, the transaction is signed with the server's private key as well and persisted in the database (update account balances). In addition, the signed transaction is broadcasted into the Bitcoin network and the clients get a confirmation from the server that the transaction succeeded. After this, the payee can be sure to eventually receive the amount in Bitcoins because he trusts that the server does not cheat (e.g. by not broadcasting the transaction) and prevents double spending.

2.5.1.3 Peer-to-Peer Transaction (Send Money)

In the peer-to-peer use case, the payer sends money to the payee. This is simpler as it does not require any intervention from the payee. The payer specifies the amount and currency and negotiates the Bitcoin transactions details with the server. Figure 2.6 shows the design of the protocol.

(P2P-1) The payer enters the amount that he wants to send as well as the currency. To proceed, the devices of the payer and payee need to be tapped together.

(P2P-2) In order to create a payment request, the payer needs to know the username of the receiver. Hence, he requests the username from the payee, who returns this information.
The payer creates a payment request which includes the username of the payer, the username of the payee, the amount to pay, the currency and a timestamp. The client sends this request to the server to proceed.

(P2P-3) The server checks the payment request and that the amount to pay does not exceed the payer's account balance. Furthermore, a Bitcoin transaction is prepared that transfers the specified amount in BTC from the payer's multisig address to the receiver's multisig address. The unsigned transaction is sent back to the payer to sign.

(P2P-4) The payer checks the amount of the transaction and signs it with his private key. To finish the transaction, the partially signed BTC transaction is sent back to the server to sign.

(P2P-5) The server verifies the signature of the transaction using the payer's public key and checks that the transaction itself was not modified. Next, the transaction is signed with the server's private key as well and persisted in the database (update account balances). Moreover, the signed transaction is broadcasted into the Bitcoin network and the payer gets a confirmation from the server that the transaction succeeded which is also forwarded to the payee.

2.5.1.4 Implications of the New Protocol

The new protocol has certain implications on the current protocol, which are outlined next. This is in particular the case for Bitcoin related aspects because the current implementation does not rely on the network for transferring money.

Key Generation There are multiple possibilities to create the key pair: the keys may be generated randomly or derived from a passphrase. The former is more secure, but it requires that the user creates a backup of the keys to prevent loss. The latter is less secure as the security depends on the strength of the passphrase that the user enters. The passphrase, however, can be human readable and the user can memorize it or write it down on paper more easily. In addition, it is less complex in the presence of multiple devices of a user as no files respectively keys need to be transferred. In either way a backup of the private key is important as payments can only be issued with a valid signature.

Messages and Size This new protocol requires more messages and the total size of data transferred is increased. This is due to the required signature of the Bitcoin transaction and their size. With multisig this cannot be avoided since the client needs to sign the particular transaction content. An alternative would be to have a full-fledged Bitcoin wallet running on the client such that the client itself can assemble the transaction on its own and sign it directly. However, this only works if the client has an up to date wallet and can retrieve transactions using the Internet. One requirement of CoinBlesk is that it should work without requiring Internet on both clients since mobile Internet is not available everywhere and for everyone. A simple optimization is to only transfer the signature back from the client to the server instead of the full transaction. The total traffic of a payment procedure is important because NFC is rather slow and the longer users have to wait the more likely a connection may break because devices are not held together anymore.

Signatures Messages between server and clients are signed to prevent manipulation of the content. Thus, it is not possible that a device that forwards a message to another client on behalf of the server can modify the message and, for instance, change the amount to pay. To avoid confusion between message signatures and Bitcoin transaction signatures, the focus is on the signatures that are relevant for the Bitcoin network.

Transaction Fees Since actual Bitcoin transactions are used and broadcasted, regular fees apply and cannot be avoided anymore. This contradicts the goal of CoinBlesk to reduce fees and make Bitcoin payments between any participants feasible for small payments as well. For larger amounts this is not that important as fees are small in comparison.

2.5.2 2-of-2 Multisig with Refund

An important aspect of 2-of-2 multisig is that money of users may be held hostage by CoinBlesk and that users cannot spend their coins if the service is shut down because the second signature is mandatory. Thus, a mechanism is required that proves and ensures that (1) the user and CoinBlesk have joint control over a certain amount of money and (2) the user gets all the escrowed deposit back if the server denies access or the service

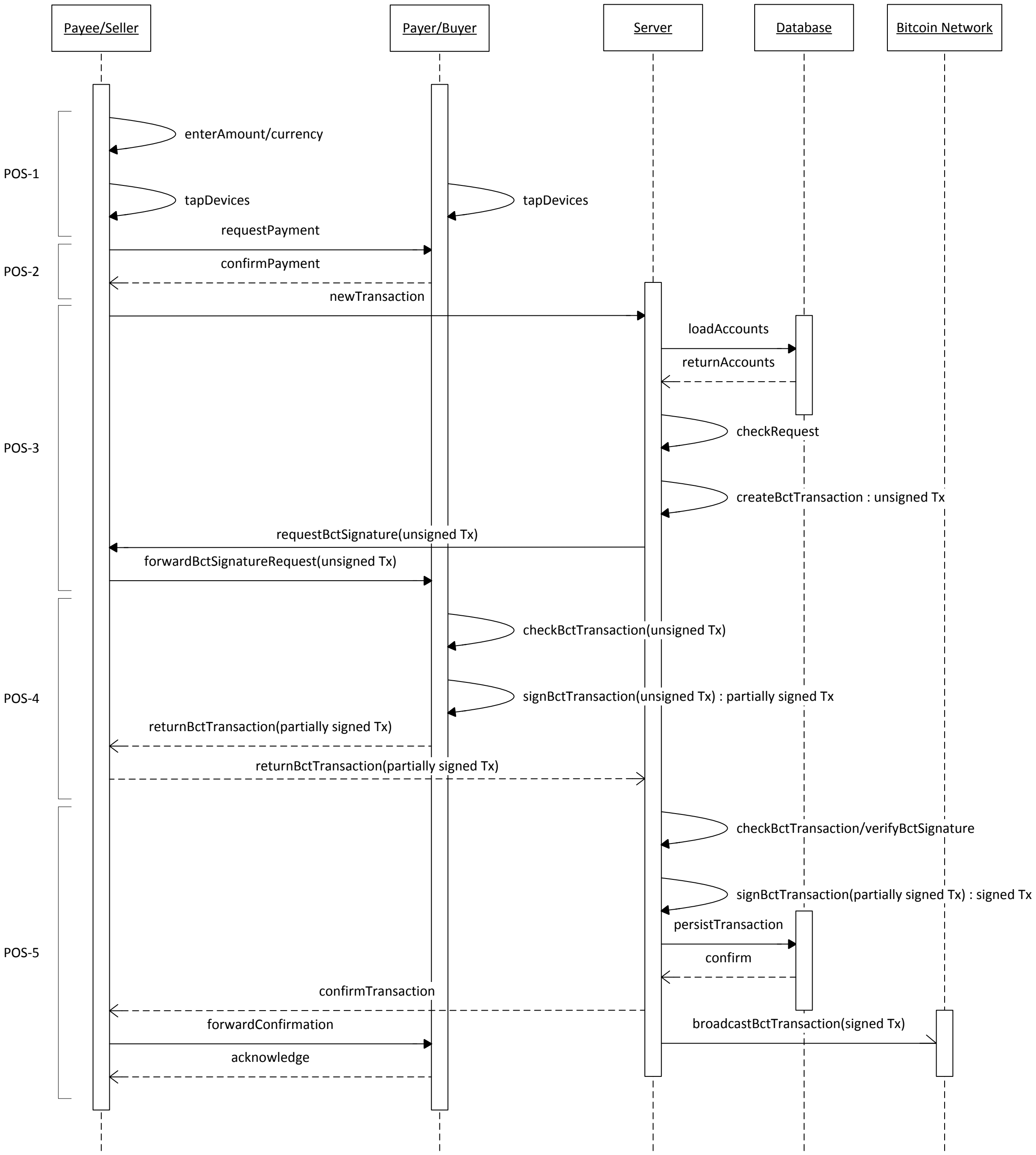

Payee/Seller
Payer/Buyer
Server
Database
Bitcoin Network
POS-1
POS-2
POS-3
POS-4
POS-5
enterAmount/currency
tapDevices
tapDevices
requestPayment
confirmPayment
newTransaction
loadAccounts
returnAccounts
checkRequest
createBctTransaction : unsigned Tx
requestBctSignature(unsigned Tx)
forwardBctSignatureRequest(unsigned Tx)
checkBctTransaction(unsigned Tx)
signBctTransaction(unsigned Tx) : partially signed Tx
returnBctTransaction(partially signed Tx)
returnBctTransaction(partially signed Tx)
checkBctTransaction/verifyBctSignature
signBctTransaction(partially signed Tx) : signed Tx
persistTransaction
confirm
confirmTransaction
forwardConfirmation
broadcastBctTransaction(signed Tx)
acknowledge

Figure 2.5: Point of sale transaction: request money

disappears. This problem can be solved by making deposits expire after a predefined time period. The user gets either a full or partial refund after that time period depending on how much money was spent.

The design and technique behind this approach is discussed next. First, delayed transactions are introduced in Section 2.5.2.1. They are a means to implement refunding of Bitcoins. The integration into CoinBlesk follows in Section 2.5.2.2.

2.5.2.1 Delayed Transactions (locktime)

Transactions can have an additional optional attribute called locktime (`nLockTime`) [8, 10]. It specifies the earliest point in time where the transaction can be included in the blockchain where the time is given either by a number referring to a block or by a timestamp (Unix time). The transaction does not lock the inputs itself and a user can spend them by creating another transaction without a locktime. Since the second transaction

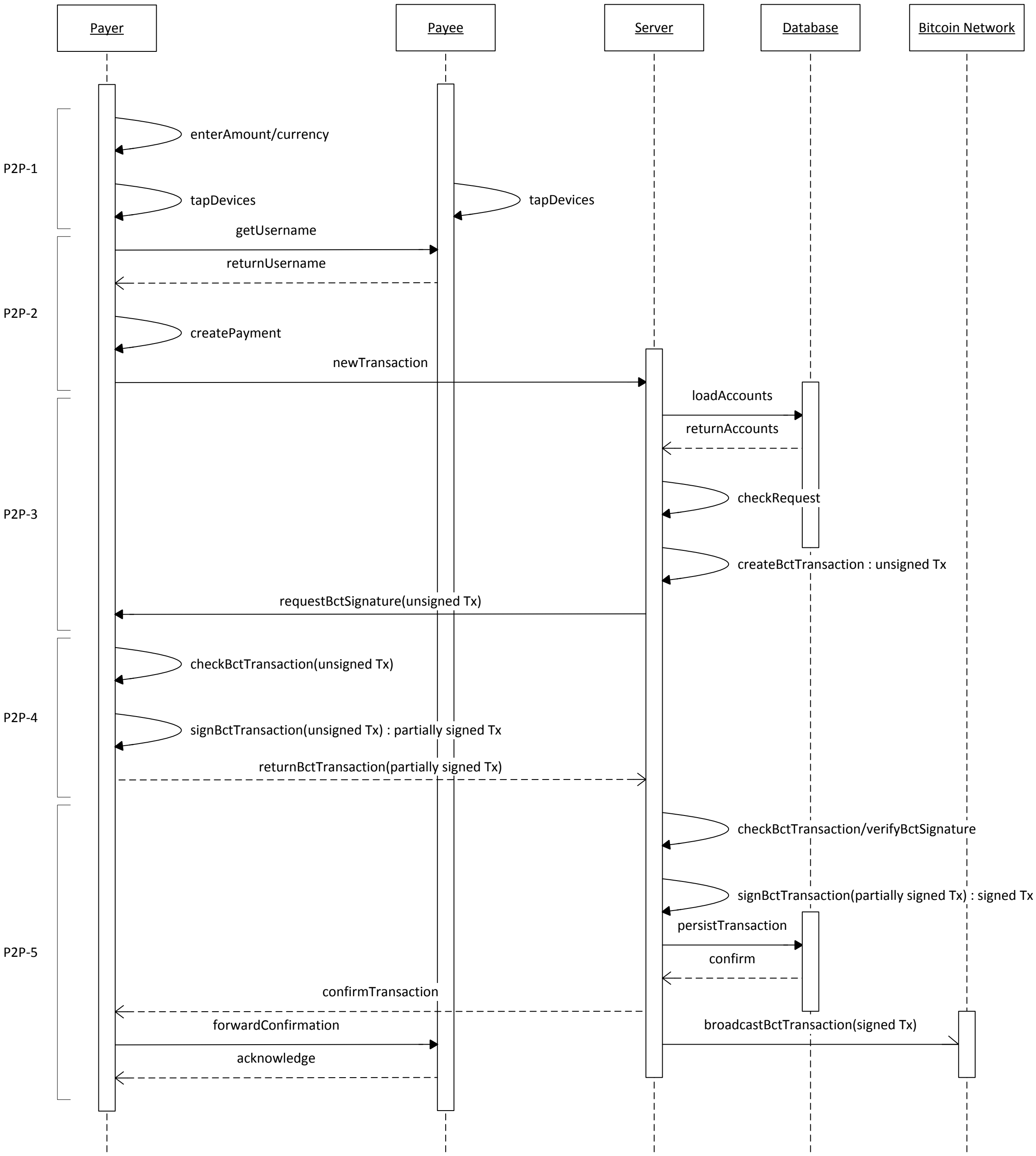

Payer
Payee
Server
Database
Bitcoin Network
P2P-1
enterAmount/currency
tapDevices
tapDevices
P2P-2
getUsername
returnUsername
createPayment
newTransaction
P2P-3
loadAccounts
returnAccounts
checkRequest
createBctTransaction : unsigned Tx
requestBctSignature(unsigned Tx)
P2P-4
checkBctTransaction(unsigned Tx)
signBctTransaction(unsigned Tx) : partially signed Tx
returnBctTransaction(partially signed Tx)
P2P-5
checkBctTransaction/verifyBctSignature
signBctTransaction(partially signed Tx) : signed Tx
persistTransaction
confirm
confirmTransaction
forwardConfirmation
broadcastBctTransaction(signed Tx)
acknowledge

Figure 2.6: P2P transaction: send money

does not have a locktime, it will be added to the blockchain in the near future. This invalidates the other transaction with a locktime. Since blocks are created in a non-predictable fashion, it is important to note that invalidating locktime transactions must happen a few hours before the specified point in time rather than minutes.

2.5.2.2 Payment Channel Setup

Using the locktime feature, a setup can be created where the user gets his deposit back from CoinBlesk in the future. Furthermore, the amount of the refund can be adjusted dynamically during the time until the expiry time. This is known as (micro-)payment channel in Bitcoin [8, 12].
To configure this setup, the deposit procedure has to be extended. Most steps are the same when compared with our proposal in section 2.5.1.1. Thus, the focus is on the part where the protocols differ which is mainly the last step of sending Bitcoins to the

CoinBlesk server. Figure 2.7 shows how to set up the payment channel with a locktime refund transaction.

(DEPR-1) The client creates a transaction that pays to the multisig address in control of the client and server. The transaction is not broadcasted or sent to the server yet, but stored on the client. Otherwise, the money would be already locked in the multisig address.

(DEPR-2) The client creates a refund transaction to an address in full control of the user. The input is the previously created deposit transaction, i.e. it sends coins to the user from the multisig account. As a consequence, two signatures are required. In addition, the transaction has a locktime in the future defining the earliest point in time where the refund may become valid. This is the expiry time of the deposit.

(DEPR-3) The client requests a signature for the refund transaction from the server. The server has to check that the transaction has a locktime in the future. In addition, the transaction is signed, persisted and returned to the client, who signs the transaction as well. Since the refund transaction refers to the deposit transaction, the latter is a precondition of the former to become valid. This makes it safe for the client to deposit money because he already is in possession of the refund transaction.

(DEPR-4) Until this point, the server does not know the deposit transaction because the refund has to be negotiated before. The server requests the deposit transaction from the client, who returns it. The transaction is compared with the refund transaction on the server and broadcasted to the Bitcoin network. The deposit transaction closes the gap between the refund and the users account. It shows that the user is the owner of the coins and that he sent it to the multisig address.

(DEPR-5) After some time, the deposit transaction will be confirmed by the network and the server can update the account balance accordingly. This completes the deposit procedure and the user can start spending.

Once the confirmation of the network is received by the server, the deposit is locked in the multisig account. Neither the server nor the client can spend without agreeing on the transaction. Both the server and client already agreed on one transaction: the refund transaction. It is fully signed and gives control over the money back to the user. The locktime specifies the earliest point in time where the refund can become valid and hence, temporarily locks the money in the multisig account. However, the server and the client can negotiate on a different refund during the time period until the expiry time.

2.5.2.3 Payments and Adjustment of Refund Amount

Payments work similar as discussed in section 2.5.1. Instead of creating a new transaction for every payment, a refund transaction is created and adapted for each payment. With each transaction, the user's account balance is split into two outputs until the expiry time of the deposit. One output goes to an address of the CoinBlesk server and the other output goes back to the user's refund address. The client signs each transaction but the server does not sign or broadcast any of these transactions except the last one just before the expiry time is up.

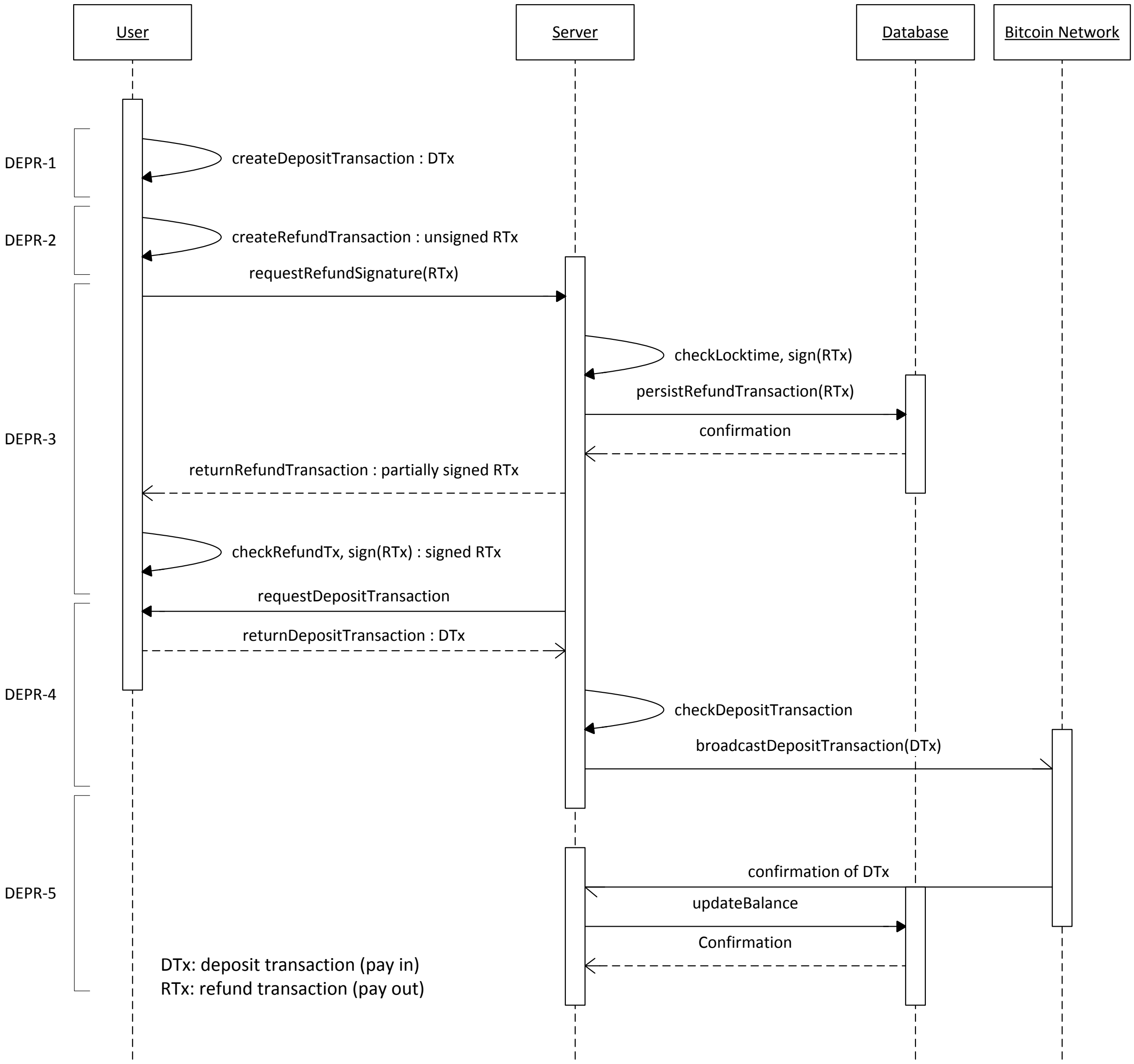

User
Server
Database
Bitcoin Network
DEPR-1
createDepositTransaction : DTx
DEPR-2
createRefundTransaction : unsigned RTx
requestRefundSignature(RTx)
checkLocktime, sign(RTx)
persistRefundTransaction(RTx)
confirmation
DEPR-3
returnRefundTransaction : partially signed RTx
checkRefundTx, sign(RTx) : signed RTx
requestDepositTransaction
returnDepositTransaction : DTx
DEPR-4
checkDepositTransaction
broadcastDepositTransaction(DTx)
confirmation of DTx
DEPR-5
updateBalance
Confirmation
DTx: deposit transaction (pay in)
RTx: refund transaction (pay out)

Figure 2.7: Deposit with Refund Transaction (Payment Channel Setup)

For instance, consider an account balance of 1 BTC (initial deposit). The user buys something for 0.1 BTC. Thus, the transaction spends 1 BTC by sending 0.9 BTC to the refund address of the user and 0.1 BTC to the server's address. The next day, the user makes another payment of 0.3 BTC which results in a transaction splitting 1 BTC into 0.6 BTC sent to the user and 0.4 BTC to the server.

The server can broadcast the most recent transaction any time which closes the payment channel and makes the accumulated payments final. Note that the intermediate transactions are not chained. Each intermediate transaction splits the deposit transaction into two parts and only the ratio or the amount of Bitcoins sent to the server and user is adjusted. Thus, only the most recent transaction has to be broadcasted. The amount that is sent to the server is never decreased (e.g. due to an incoming payment to the user). It certainly could be decreased but there would no way for the user to prevent that the server does not broadcast the transaction with the higher amount anyways.

A consequence of this design is that the CoinBlesk server is an intermediate receiver of each payment because it is not possible to have intermediate transactions to multiple receivers. In theory, it would be possible to specify more than two outputs, i.e. for each payee an output. However, payees would not have access to the Bitcoins until the transaction is broadcasted right before the expiry time. Thus, the CoinBlesk service has to issue transactions from the server's account to the recipient. This means that the service has to provide a certain amount of spare funds that can be sent to the user.

2.5.2.4 Implications of the Refund Protocol

Refund Address An important aspect of this design is the refund address where refunds are sent to. It has to be ensured that the user is in control of the refund address and owns the associated private key. This becomes more interesting when considering cases where the user uses third party services to make a deposit. For instance, a Bitcoin ATM may send Bitcoins to an address of the user's choice. However, the user certainly does not want that the refund is sent back to the Bitcoin ATM (respectively an address of the ATM). Thus, it is probably not advisable to send the refund back to an associated address of the inputs of the deposit transaction. Similar concerns apply for exchange services such as Bitstamp.

A simple solution would be that (1) the user specifies the address or (2) CoinBlesk offers a refund address included in the application. The address can, for instance, be derived from the public key that is already used in the multisig address.

Automatic Deposit A user may still want to use CoinBlesk after a refund. Thus, a convenient feature is to allow for automatic deposits after a refund. If a refund address is used that is under control of the CoinBlesk application, the user can configure an automatic deposit after a refund. Moreover, this prevents the case where the user wants to pay for something but first has to deposit money again and wait due to a very recent refund. This is in particular relevant because certain implementations of the payment channel have limits regarding the expiry period in place. For instance, BitcoinJ [12] currently uses an expiry period of one day.

Time Dimension In this design time becomes relevant for the server because the server has to claim the money spent by the user before the expiry time. If the server does not broadcast the last transaction before the expiry, the user receives a full refund and the service loses money. Thus, a reliable Internet connection and service is very important in this scenario. Furthermore, the locktime is not a hard limit but rather a soft limit because the exact point in time cannot be predicted in advance. As a result, the server has to broadcast the transaction hours before the locktime such that the network can confirm the transaction before the full refund.

2.5.3 Multisig with More than Two Keys

Even though 2-of-2 multisig may be the most feasible approach in terms of resolving issues in CoinBlesk and offering desired functionality, there are other solutions that involve more than two keys.

A central question with more than two keys is: who gets the additional keys? To guarantee the CoinBlesk functionality, the user should never have full control over the funds because instant confirmation relies on having at least partial control over a transaction. Furthermore, it should not be possible to spend money without the service noticing it in order to keep track of the account balances.

2.5.3.1 2-of-3 Multisig

In a 2-of-3 scenario, the user and CoinBlesk each have one key. The third key serves as a backup key that can be stored by a trusted third party such as an escrow service or notarial service which can mediate in case of a dispute. This avoids that CoinBlesk can hold user funds hostage and prevent access to it or that funds are lost if the service is shut down. Escrow services are already known from traditional contracts and multisig enables this in the Bitcoin world using cryptography and the blockchain.

Escrow services are a key feature of multisig and mentioned frequently by the Bitcoin community [18, 51]. However, systems will get more complex because an additional party is introduced and setting up accounts and exchanging keys requires some coordination between all participants. Furthermore, the reputation of escrow services becomes relevant for users since they need to decide whether they trust a service or not.

2.5.3.2 Even More Keys

The 2-of-3 scenario already shows how third parties can provide services that allow implementing additional functionality. For instance, other functionality could be fraud detection where an oracle analyzes transaction and computes a probability or some risk score based on past behavior of the user and other data such as the location. Depending on the outcome the transaction is signed or not and hence, can be broadcasted and become valid or not. CryptoCorp [20] is a company that proposes such a solution where they provide similar functionality as credit card companies that try to detect and prevent fraudulent usage of credit cards. This can be achieved with 3 required signatures out of 4 where the user, CoinBlesk and an oracle have to sign a transaction. The fourth key would be a backup key that allows spending without the oracle if desired.

2.6 Related Work

GreenAddress [27] is an online wallet service targeting mobile devices and desktops (browser). They focus on security and use P2SH wallets to secure the funds of the users. In its default configuration, GreenAddress locks the funds in 2-of-2 addresses that require a signature of the user and the service provider. Furthermore, they offer 2-of-3 addresses to give users more control over their account. A key feature of GreenAddress is *instant confirmation* which eliminates the need to wait until transactions are confirmed by the network. 2-of-2 addresses are used for this purpose and the service denies signing if it detects a double spending attempt [36]. The service is similar to CoinBlesk, which offers instant payments without any waiting time. The current CoinBlesk implementation controls the private keys to avoid double spending and confirmations. To further protect user funds locked in a 2-of-2 address, GreenAddress mails pre-signed transactions to the user that can be redeemed after a user-defined time [28]. The pre-signed transactions send the funds from the multisig address to a regular address that only requires the user's signature. As a consequence, users have full control over their funds even if the service would be shut down. However, users have to keep and backup these pre-signed transactions. Users have to sign and broadcast these transactions if they want to withdraw funds without the service's signature.

The Bitcoin Improvement Proposal (BIP) 70 [1] outlines a protocol for communication between merchants and customers. The aim is to improve the user experience and protect the payments against attacks. Lawrence Nahum [36] from GreenAddress proposed an extension to BIP 70 which enables instant confirmations using multisig addresses. It uses the aforementioned concept and even aims at interoperability between services as long as they trust each other. It proposes a mechanism that allows merchants to ask for a signature from the trusted service that the transaction actually was issued. CoinBlesk tries to achieve a similar goal but currently only within service boundaries and not beyond multiple services. The extension is purely informational and not standardized or accepted as an official part of Bitcoin by the community.

Sigsafe [41] is a project with the aim to secure Bitcoin transactions. The general idea is that users have an NFC enabled hardware token that can be used to sign transactions. This integrates nicely with multisig where the hardware token provides one signature and a computer or mobile device the other one. It targets end-users and their personal wallets and is not a service itself that acts as mediator between payer and payee. Sigsafe is not a wallet itself but a device to sign transactions over a channel that is difficult to exploit (short air distance of NFC).

BitPay [15] is a service provider for merchants that want to get paid in Bitcoins. It does not target end-users in particular but rather point of sale businesses with direct customer interaction. In addition, it offers instant confirmations by detecting double spend attempts of payers. This is done without cryptographic techniques such as multisig. Transactions are accepted with zero confirmations from the network if the service is confident that there is no double spending. Furthermore, BitPay offers mobile payments using NFC by transmitting the payment details to the customer who then uses his regular Bitcoin wallet to finish the purchase. This is different compared to CoinBlesk, where the payee and payer use the same software or service and there is a custom communication protocol in place to negotiate payment details. BitPay tries to be more general which means that they cannot implement custom protocols between client software as the BitPay terminals need to be compatible with existing wallets. CoinBlesk, on the other hand, trades generality and in turn offers additional functionality such as offline payments. Due to the way BitPay works, their solution requires an Internet connection.

Goldfeder et al. [26] proposed threshold signatures as an alternative to the multisig mechanism of Bitcoin. They demonstrated that this technique works with Bitcoin private keys and is beneficial for many use case scenarios. Threshold signatures can be used to achieve joint control over Bitcoins by requiring multiple private key shares to create a signature for a transaction.

2.7 Summary and Conclusions

A few years have passed since the introduction of the Bitcoin cash system in 2008, during which an ecosystem of services and users evolved driven by enthusiastic early adopters and startup companies. In general, the technical foundation of Bitcoin is able to cope with the standard use case of a payment system: sending and receiving money. However, users face many challenges which are increasingly important now that Bitcoin has become much more popular. One crucial aspect is the safety of Bitcoins because many users and services were and still are targets of criminals, victims of fraud or have lost money due to missing wallet backups or theft.

The multisig concept is a technical answer to these issues. While traditional Bitcoin transactions need to be signed by one private key, multisig enables the use of more than one private key associated with a transaction. For instance, it is possible to create an address associated with three private keys out of which two are required to spend money. As a consequence, a lost key does not lead to loss of money and the level of security is increased because more than one key is required.

Furthermore, the multisig concept allows implementing customer protection by the use of escrow services. This is important because transactions are not reversible and there is no centralized authority that could solve disputes between a sender and a receiver of Bitcoins.

This paper discusses the feasibility of the multisig concept in CoinBlesk, a mobile payment solution for Bitcoins that allows payments with smartphones over NFC. CoinBlesk is built as a client-server architecture where private keys of clients are kept on the server. The motivation for multisig in CoinBlesk is manyfold. (1) It should not be possible to spend money without a user approving the transaction. With multisig, one key can be held by the user and one key can be stored on the server. Each transaction then needs to be signed by both parties in order to be valid. (2) A compromised server should not lead to lost money. This can be satisfied as soon as more than one key is used. However, depending on the number of keys that are used, a lost key immediately means that the coins are lost and cannot be retrieved anymore. (3) Since the private keys essentially correspond to assets of users, it may be required to comply with financial regulations and law. In Switzerland, a banking license is required as soon as 20 customers deposit money. Whether multisig helps is not clear as specific laws are not in place. On the one hand, it may not avoid the requirement of a license because no matter how keys are distributed, the customer is not able to withdraw money without the consent of the CoinBlesk server operator. On the other hand, automatic refunds allow locking deposits temporarily and give control over the assets back to the user after a certain time which may avoid a license. (4) Customer protection is not of great concern in CoinBlesk because users usually meet in person for each transaction.

Bibliography

[1] G. Andresen, M. Hearn: *Payment Protocol*; Bitcoin Improvement Proposal 70, July 2013, https://github.com/bitcoin/bips/blob/master/bip-0070.mediawiki.

[2] G. Andresen: *M-of-N Standard Transactions*; Bitcoin Improvement Proposal 11, October, 2011, https://github.com/bitcoin/bips/blob/master/bip-0011.mediawiki.

[3] G. Andresen: *Pay to Script Hash*; Bitcoin Improvement Proposal 16, January, 2012, https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki.

[4] G. Andresen: *Address Format for pay-to-script-hash*; Bitcoin Improvement Proposal 13, October, 2011, https://github.com/bitcoin/bips/blob/master/bip-0013.mediawiki.

[5] Armory Bitcoin Wallet; https://bitcoinarmory.com, November, 2014.

[6] bitaddress.org; https://www.bitaddress.org, November, 2014.

[7] Bitcoin; https://bitcoin.org, October, 2014.

[8] Bitcoin Developer Guide; https://bitcoin.org/en/developer-guide, November, 2014.

[9] Bitcoin Developer Reference; https://bitcoin.org/en/developer-reference, December, 2014.

[10] Bitcoin Protocol Specification; https://en.bitcoin.it/wiki/Protocol_specification, November, 2014.

[11] Bitcoin Transaction; https://en.bitcoin.it/wiki/Transaction, November, 2014.

[12] BitcoinJ: *Working with micropayment channels*; https://bitcoinj.github.io/working-with-micropayments, November, 2014.

[13] Bitcointalk.org: *List of Major Bitcoin Heists, Thefts, Hacks, Scams, and Losses*; https://bitcointalk.org/index.php?topic=576337, November, 2014.

[14] BitGo; https://www.bitgo.com/, November, 2014.

[15] BitPay Payment Service; https://bitpay.com/, November, 2014.

[16] Bitstamp Bitcoin Exchange; https://www.bitstamp.net/, November, 2014.

[17] Blockchain.info; http://blockchain.info/, November, 2014.

[18] V. Buterin: *Multisig: The Future of Bitcoin*; Bitcoin Magazine, March, 2014, http://bitcoinmagazine.com/11108/multisig-future-bitcoin/.

[19] V. Buterin: *Multisig: A Revolution Incomplete*; Bitcoin Magazine, July, 2014, `http://bitcoinmagazine.com/15290/multisig-revolution-incomplete/`.

[20] M. Cuperman, R. Singer, L. Saar: *Securing wallets by integrating a third-party Oracle*; CryptoCorp, February, 2014, `https://cryptocorp.co/technology.htm`.

[21] Coinbase; `https://www.coinbase.com/`, November, 2014.

[22] CoinBlesk; `https://bitcoin.csg.uzh.ch/`, October, 2014.

[23] Crypto-Currency Market Capitalizations; `http://coinmarketcap.com/`, November, 2014.

[24] Electrum; `https://electrum.org/`, November, 2014.

[25] Federal Council of Switzerland, Federal Department of Finance of Switzerland: *Federal Council report on virtual currencies in response to the Schwaab (13.3687) and Weibel (13.4070) postulates*; Report, June, 2014, `http://www.news.admin.ch/NSBSubscriber/message/attachments/35355.pdf`.

[26] S. Goldfeder, J. Bonneau, E. W. Felten, J. A. Kroll, A. Narayanan: *Securing Bitcoin wallets via threshold signatures*; Preprint, Security & Privacy Research Group, Princeton University, March, 2014, `http://www.cs.princeton.edu/~stevenag/bitcoin_threshold_signatures.pdf`.

[27] GreenAddress.it: *Free and secure online Bitcoin wallet*; `https://greenaddress.it/`, November, 2014.

[28] GreenAddress.it: *P2SH 2-of-2 Address Wallet Service Implementation*; Whitepaper, April, 2014, `http://ghgreenaddress.files.wordpress.com/2014/04/greenaddressp2sh2of2hd-61.pdf`.

[29] A. Greenberg: *Another Bitcoin Startup Tanks After $600,000 Theft*; Forbes magazine, March, 2014, `http://www.forbes.com/sites/andygreenberg/2014/03/04/another-bitcoin-startup-tanks-after-600000-theft/`.

[30] M. H. Ibrahim, I. A. Ali, I. I. Ibrahim, A. H. El-sawi: *A robust threshold elliptic curve digital signature providing a new verifiable secret sharing scheme*; Circuits and Systems, 2003 IEEE 46th Midwest Symposium on, Volume 1, p. 276-280, December, 2003, `http://dx.doi.org/10.1109/MWSCAS.2003.1562272`.

[31] S. Kaeser: *Improving the Mobile Bitcoin Payment System (MBPS)*; Master Thesis, University of Zurich, September, 2014.

[32] T. Kerin: *The Year of Multisig: How is it Doing So Far?*; May, 2014, `http://www.coindesk.com/year-multisig-so-far/`.

[33] R. McMillan: *$1.2M Hack Shows Why You Should Never Store Bitcoins on the Internet*; Wired magazine, November, 2013, `http://www.wired.com/2013/11/inputs/`.

[34] R. McMillan: *The Inside Story of Mt. Gox, Bitcoin's $460 Million Disaster*; Wired magazine, March, 2014, `http://www.wired.com/2014/03/bitcoin-exchange/`.

[35] J. Memeti: *Protocol Design and Implementation for a Fast and Reliable Mobile Bitcoin Payment System (MBPS) with two-way NFC*; Master Thesis, University of Zurich, August, 2014.

[36] L. Nahum: *Payment Request Instant Confirmations*; June, 2014, https://github.com/greenaddress/bips/blob/bip-payment-request-instant-confirmations/bip-payment-request-instant-confirmations.mediawiki.

[37] S. Nakamoto: *Bitcoin: A Peer-to-Peer Electronic Cash System*; https://bitcoin.org/bitcoin.pdf.

[38] Offline Bitcoin Wallet; https://www.offlineaddress.com/, November, 2014.

[39] p2sh.info; http://p2sh.info/, November, 2014.

[40] PayPal: *Safety and Security*; https://www.paypal.com/webapps/mpp/paypal-safety-and-security, November, 2014.

[41] Peter R: *Sigsafe: An electronic key tag for signing bitcoin transactions*; October, 2014, http://www.sigsafe.org/sigsafe.pdf.

[42] S. Ressler: *I Sign, You Sign, We All Sign: Explanation of Multi-signature Transactions*; Bitcoin Magazine, April, 2014, http://bitcoinmagazine.com/11848/multisig-explained/.

[43] A. Shamir: *How to share a secret*; Magazine Communications of the ACM, Volume 22 Issue 11, November, 1979, http://dl.acm.org/citation.cfm?id=359176

[44] R. Shea: *SecretSharing*; https://github.com/onenameio/secret-sharing, November, 2014.

[45] Swiss Financial Market Supervisory Authority (FINMA): *How investors can protect themselves against unauthorised financial market providers*; Report, April, 2014, http://www.finma.ch/e/privatpersonen/documents/kundenschutz-e.pdf.

[46] Swiss Financial Market Supervisory Authority (FINMA): *Bitcoins*; Factsheet, June, 2014, http://www.finma.ch/e/finma/publikationen/faktenblaetter/documents/fb-bitcoins-e.pdf

[47] Swiss Parliament: *Bitcoins und Geldwäschereigesetz* (German); September, 2013, http://www.parlament.ch/d/suche/seiten/geschaefte.aspx?gesch_id=20133854.

[48] Swiss Parliament: *Die Risiken der Online-Währung Bitcoin evaluieren* (German); September, 2013, http://www.parlament.ch/D/Suche/Seiten/geschaefte.aspx?gesch_id=20133687.

[49] Swiss Parliament: *Rechtssicherheit für Bitcoin schaffen* (German); December, 2013, http://www.parlament.ch/d/suche/seiten/geschaefte.aspx?gesch_id=20134070

[50] Trezor Bitcoin Safe: *Bitcoin Hardware Wallet*; https://www.bitcointrezor.com/, November, 2014.

[51] J. Villasenor: *Could 'Multisig' Help Bring Consumer Protection To Bitcoin Transactions?*; March, 2014, http://www.forbes.com/sites/johnvillasenor/2014/03/28/could-multisig-help-bring-consumer-protection-to-bitcoin-transactions/.

[52] P. Wuille: *Hierarchical Deterministic Wallets*; Bitcoin Improvement Proposal 32, February, 2012, https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki.

Chapter 3

Cloud-based Services: To Move or Not To Move

Cristian Anastasiu, Taya Goubran

The emerging paradigm of Cloud Computing and services in the recent years is very popular among analysts, software, hardware vendors, and organizations using IT. Analyst reports show that growth rate of Cloud Computing is five times higher than the overall IT growth rate, with an estimated worldwide market of $107 billion in 2017 [27]. Some years ago, this market was only dominated by a few players, the most popular of which were Amazon, Microsoft, and Salesforce.com. Currently, most of the big hardware and enterprise software vendors, such as IBM, Oracle, SAP have also started to change and adapt their development, marketing strategy and offerings, adding a rich portfolio of granular services in the cloud in addition to their legacy product offerings. The cloud model may seem tempting from a flexibility and costs perspective with the pay-as-you-go model, promising a lower total cost of ownership and a transformation from capital costs to operational costs. However, there are many other technical, economic and organizational factors that play a role in the decision of investing or moving the current legacy or parts of it to a cloud service model. This paper tries to identify and summarize factors that may influence such decisions in enterprises based on different surveys and research done in this field. These factors are analyzed and categorized according to three different dimensions: type of cloud service, importance of the application and type of factor – technical, economical, and organizational

Contents

3.1 Introduction

The idea of computing being delivered as a public utility dates back to sixties by computer scientist John McCarthy [9]. Due to the limited Internet bandwidth until the nineties, the development of such a vision was delayed [9]. The term "Cloud Computing" was coined by some executives in Compaq Computer in 1996 [11]. Nowadays, the cloud is everywhere. While more than 50% Americans claim to have never used the cloud, 95% of them are using the cloud without knowing it [22]. Whether it is online banking, online shopping, or social networking, these are cloud-based applications. Using the cloud has its upsides as well as its downsides. Let us consider the following example. Pinterest [32] is a web and mobile application company for social media. It managed to expand from 50,000 users to 17 million users in only nine months with currently in 2014 around 48 Million users. Such immense growth with only 12 employees was due to adopting cloud-based services to manage the data centers and store around 400 terabytes of data [21]. While this is a great success, there are many factors to be considered. Relying on the cloud makes the organization dependent on the services, offerings, and limitations of the service providers. Due to the momentary lack of standards in the cloud-based services, the user is strongly dependant on the chosen service provider. Once the user chose a service, that is suitable for the organization's requirements, the user is contractually bound to the services provider. If the vendor performs changes in the service, the user has to adapt or otherwise change the provider, which could be rather costly depending on the contracts and SLAs. The users are able to configure the services as required within the provided framework of the service provider. If the provider has availability issues the organizations could experience outage causing their servers to be inaccessible. This report discusses such and other factors that should be considered when deciding whether to adopt cloud-based services. Although there are many other factors such as environmental, social or technological factors, the report concentrates on the three main dimensions: technical, economic and organizational factors.

The structure of this report is as follows. Firstly, related work that has been researched in this area is mentioned, indicating the increasing interest and concerns in the topic. Secondly, Cloud Computing is defined, describing its features and characteristics while showing the clouds features and types. Thirdly, the market is analyzed showing some statistical figures about its growth and the top service providers in the respective area are mentioned. In the main section the individual factors are analyzed from a technical, economical and organizational perspective, in which each factor is explained, showing its importance to cloud adoption decisions in an organization. Finally, the findings are discussed and a conclusion is summarized.

3.2 Related Work

The question of migrating to cloud or to continue using on-premise resources has been and still is a big question and a main concern in the IT industry. Much research has been conducted discussing arguments that are for cloud as well as against cloud. While some look at the costs benefits, others worry about technical failures. In [18] key factors are analyzed, which influence the cost of a deployment choice indicating that the cloud has many great features, but discusses whether it is suitable for the organizations own use. A research conducted by Ovum compares on-premise and cloud costs over a five year period with regard to small, midsized and large companies using small, medium and large infrastructure technology for contact centers [5]. In [1] technical (*e.g.* availability, performance) as well as non-technical(*e.g.* vendor lock-in, data transfer) obstacles and opportunities of using the cloud are identified and methods for avoiding such obstacles were mentioned.

There has also been research made, asking customers about the perceived benefits and disadvantages of the cloud with regards to technological factors, non-technical factors, vendor customer relationship, duration and the extent of the adoption [24] [4] [34] [23]. The studies and researches conducted observe cloud with regards to the same characteristic from different viewpoints. While [1] ,for instance, mentions the fear of vendor lock-in, another study by International Data Corporation (IDC) showed that customers prefer to establish a relationship with a single trusted vendor with a wide variety of services instead of scattering their data across multiple providers [34].

3.3 What is Cloud Computing?

Over the years Cloud Computing became more famous than ever and is the new hype in the IT-industry, but confusion remains on the clear definition of Cloud Computing[14]. In [1] Cloud Computing was referred to simply as the applications being delivered as a service as well as the hardware and systems on which the service is provided. The National Institute of Systems and Technology (NIST) defined Cloud Computing as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (*e.g.*, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models” [7], which will be discussed in the remainder of this section.

3.3.1 Characteristics of a Cloud Service

The cloud offers users what may seem like an infinite pool of resources (*e.g.*, data storage, server time, bandwidth) [1]. These resources are offered on-demand with a pay-as-you-go model. Whenever the user is in need of certain resource capacities, these capacities are available for immediate usage with high elasticity and without interference from the service provider. Once resources are no longer needed, they are released by the user and is then charged only for the consumption of used resources.

Note the difference to the renting model (as opposed to pay-as-you-go), in which the resources are provided over a certain period of time and the consumer is charged for the rent of these resources, whether they are used or not [1]. Cloud-based services are charged to resource usage. The different payment models and possible disadvantages are discussed later in Section 3.5.2.1. Using a multi-tenant model, the cloud vendor is able to supply multiple consumers on the same platform, based on their needs of resources. The users do not need to control, manage or maintain the provided service. This is done by the service provider.

From a hardware perspective, Cloud Computing offers flexibility due to the infinite resource pool, elimination of up-front costs and a suitable payment model, which allows consumers to use the resources by the day, by the hour or another negotiated payment model. Companies using Cloud Computing can rely on the elasticity of the services in order to serve peak times, unlike the traditional data centers where companies had to deal with over-provisioning or under-provisioning. So, instead of investing in servers, storage and networking when starting a business, by using cloud-based services companies can transfers Capital Expenses (CapEx) into Operational Expenses (OpEx) [1]. This can be a decisive cost saving factor, allowing companies to direct their investments into other areas, such as innovation.

3.3.2 Deployment Models

The cloud-based services can be used and deployed using different models, which offers consumers the flexibility of different management, control, and security options. The two extremes are private cloud and the public cloud. Depending on the organizational requirements and use cases, both models offer advantages and disadvantages. The public may appear more convenient for the consumer in terms of operations and costs, because the provider maintains, updates and controls the resources for the users. However, some consumers may question the integrity and security of their data stored on a cloud platform. Therefore, private cloud offers the consumers the control over their private data and operations by keeping the infrastructure and the service on premise of the organization. Another deployment model, which tries to combine the two models mentioned above attempting to combine the best of both worlds, is the hybrid cloud approach. A hybrid cloud is a Cloud Computing environment, in which an organization provides and manages some resources in-house using a private cloud and has others provided externally through a public cloud [13].

For an overview, the fastest growing business services with regards to the public and private cloud are shown in Figure 3.1.

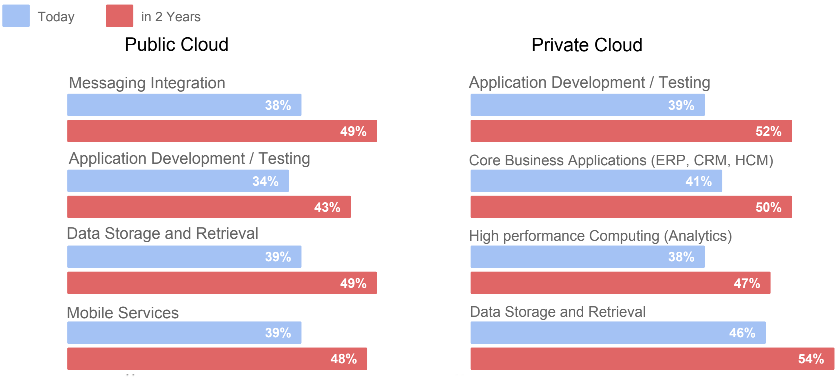


Figure 3.1: Fastest Growing Business Services in the Cloud [23]

3.3.2.1 Public Cloud

The public cloud is the most popular and accessible cloud deployment model. While the term “Public Cloud” is relatively new and used excessively only in the last years, public cloud-based services, such as web email or collaboration platforms have been offered for a long period of time. Hosted by a third party service provider and accessed through Internet, public cloud-based services are designed to serve multiple consumers by sharing the computing resources and infrastructure amongst them. In such a model, consumers benefit from the economy of scale due to shared costs among the different users and allows the usage of statistical multiplexing, which increases resource utilization compared to on-premise environments [1]. The cost of using the service is in most cases either based on a pay-per-use model or on a monthly subscription model. This can be seen as an advantage from an economic perspective, because compared to traditional IT projects, no initial investment in hardware such as servers and storage space or qualified staff with the essential know-how are required for the cloud-based services. The costs become operational expenses and enable organizations to optimize their cash flow and their IT budgets.

By using multi-tenancy and separating each customer account with the required resources into isolated virtual spaces, cloud service providers are able to serve millions of customers.

This is also a way to isolate customer data from the data of other customers. Administration and monitoring of the service, if enabled, is also done though the web in a self-service mode by accessing a management console offered by the hosting provider. The maintenance of the datacenters with the required infrastructure, as well as patching and upgrades of the different cloud components, is done by the provider. This reduces the complexity and costs on the consumer side, but also raises security concerns in regards to the confidential data stored in the cloud. For many organizations, using the public cloud for its core applications involves a high risk. This is due to the high security and performance requirements of these applications, which might not always be fulfilled by the public providers. So in order to make use of the advantages of cloud provisioning, but be in full control of the used infrastructure, consumers can choose the private cloud model.

3.3.2.2 Private Cloud

In order to address concerns regarding the shared access of the resources and lack of control over their data, consumers can choose a private cloud model. The private cloud is hosted mostly on the premise of the consumer. Depending on the requirements of the cloud software, consumers can use their own commodity hardware or buy special hardware from the cloud vendors tuned and pre-configured for running cloud-based services. This type of cloud model is more expensive for the consumer than the public cloud due to the sole occupation of the resources and the additional management and administration costs, but solves the security, control and compliance issues. Management of the private cloud can be done by the customer, or, if the customers lack the required experience in doing so, it can opt for a managed private cloud approach. In the latter case, the cloud infrastructure is hosted in the customer's data center site but managed by the cloud vendor either on site or remotely through a secure connection. Having the cloud infrastructure managed by an external service provider through a remote connection can raise concerns about the security of the data, and in some cases be inconsistent with internal compliance regulations. This is why in some domains such as the banking or insurance industries, these kind of methods are avoided or prohibited.

3.3.2.3 Hybrid Cloud

The third Cloud Computing model, which has increased in popularity in the recent years, is the hybrid model. This model implies that the consumer uses and manages some resources on-premise, using a private cloud, and has others provided externally in a public cloud; it tries to combine the better of the two worlds. Depending on the situation, according to the study of [18] "the most cost-effective approach for an organization might, in fact, involve a combination of cloud and in-house resources rather than choosing one over the other". One example would be to use the public cloud for development and testing with anonymized data, and use the internal private cloud for production. However, one key requirement for the scenario above is portability between public and private cloud.

3.3.3 Offerings and Service Models

In this section we will look at the different cloud categories from a solution and offering perspective. Although cloud offerings are getting more granular every year, with offerings tailored for particular use cases or applications, such as Integration-as-a-Service, Mobile-as-a-Service, Data-as-a-Service, Process-as-a-Service and many other services, it can still be divided in three major Cloud Computing categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). Each of the three categories is explained in more detail in the following sections.

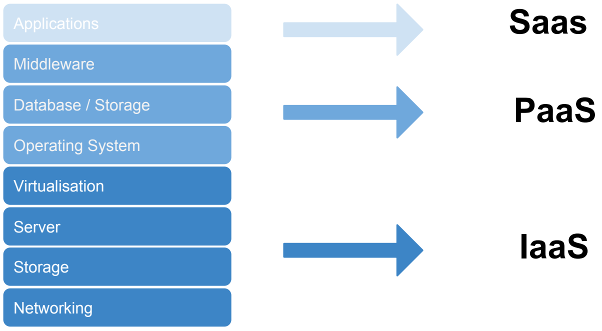


Figure 3.2: Cloud Offerings based on System Layers [27]

3.3.3.1 Software-as-a-Service

SaaS is the most common cloud service deployment model and refers to applications delivered remotely as services over the Internet. The most popular examples are web email clients such as Gmail, Hotmail or Yahoo, and sales tools such as Salesforce.com, which until recently defined itself as the leader of the Cloud Computing revolution [3]. According to the definition by Garner, "the provider delivers software based on one set of common code and data definitions that is consumed in a one-to-many model by all contracted customers at anytime on a pay-for-use basis or as a subscription based on use metrics" [26]. Standard charging model for this type of service is either a pay-as-you-use model (also called pay-as-you-go or pay-for-use) or a user-based subscription model. Although, the majority of the early SaaS services were mostly in the email and collaboration area, all major enterprise software vendors, including SAP, Oracle, Microsoft and Salesforce.com, are starting to offer all their core business applications such as Enterprise Resource Planning (ERP), Human Capital Management (HCM), or Customer Relationship Management (CRM) as cloud-based services.

3.3.3.2 Platform-as-a-Service

Another offering, which from a layer perspective lies between SaaS and IaaS (see Figure 3.2) and which will experience a rapid growth in the years to come according the IDC [27], is Platform-as-a-Service. Platform-as-a-Service enables consumers to develop, configure and deploy applications in the cloud using infrastructure, tools and frameworks (platform) provided by the service provider. Network, storage, and other additional services required to run the application are also provided by the cloud provider. One example would be a Tomcat application server provided as a deployment platform by the cloud provider, where consumers can deploy and manage their custom Java applications. A similar example is Database-as-a-Service, where consumers get access to a database scheme which can be used for their applications.

3.3.3.3 Infrastructure-as-a-Service

The third main type of cloud-based service is Infrastructure-as-a-Service (IaaS), which allows the customer to use computing resources such as servers, networking components, or storage in order to run their own applications, software, and possibly operation systems without having to manage or maintain the underlying infrastructure. This deployment model is suitable for customers who do not require the platform or software services, but only the infrastructure to their own build products. Customers access the infrastructure, using a Web-based graphical user interface, which serves as an IT operations management console for the overall environment [26]. Among the many offering, some of the most famous infrastructure services are Dropbox, Cisco and Amazon Web Services. Amazon Web Services is not only restricted to IaaS, it offers a wide range of services, such as the Amazon Elastic Compute Cloud (EC2) for computing services, Storage and Content Delivery Services, Database Services, Analytics Services, Deployment and Management services and many more [19].

3.4 Cloud Service Market

When tracking the term Cloud Computing from 2004 until 2014 in Google Trends and Google Scholar (see Figure 3.3), a strongly increasing interest in the topic can be recognized. Ed Anderson, research director at Gartner said "The continued growth of the cloud-based services market will result from the adoption of cloud-based services for production systems and workloads, in addition to the development and testing scenarios that have led as the most prominent use case for public cloud services to date" [26].

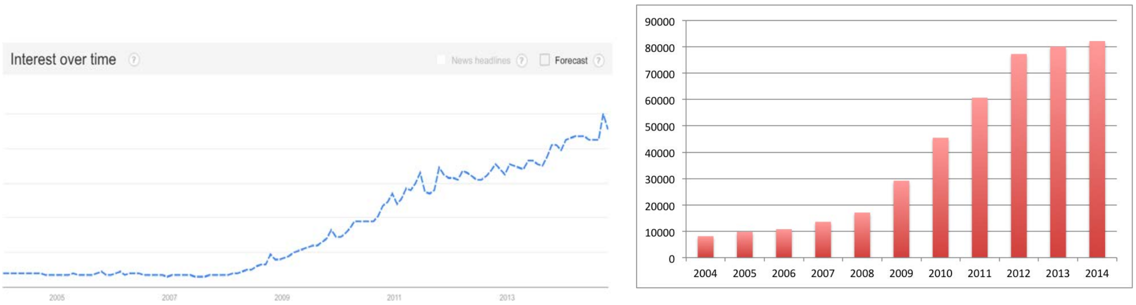


Figure 3.3: Google Trends (left) and Google Scholar (right) on the search of the Term Cloud Computing

Companies use software without having to test it, use storage without having to maintain it and use networking components without needing to configure it. Servers are acquired when needed during peak hours and released when they are no longer required. The rapid growth of cloud-based services is a response to the increasing demand of the market in terms of flexibility, elasticity and agility among others. The scalability of the cloud lets the companies avoid buying and maintaining resources in case of overload and avoid the risk of not having these resources when needed. Although continuing to adopt cloud-based services, companies still fear outages and down times. The factors that influence such a decision are discussed later in Section 3.5. In Figure 3.4, the enterprise cloud adaption trends are shown.

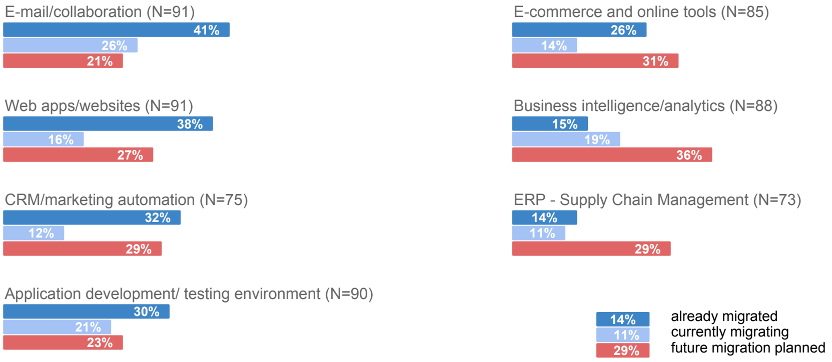


Figure 3.4: Enterprise Cloud Adoption Trends by Workload [24]

3.4.1 Growth Projection

In recent years, many companies have shifted to cloud-based services. Over the years the demand has risen and the offerings increased of variety and kind. In 2017 IT spending for cloud-based services is estimated at more than \$107 billion as reported in the new forecast from the IDC [27]. Its expected to have compound annual growth rate of 23.5% from 2013 until 2017 is five times as much growth as expected of the whole IT industry [27]. While SaaS remains the most dominant service segment, the platform and infrastructure services are growing to double and triple until 2017 respectively (see Figure 3.5).

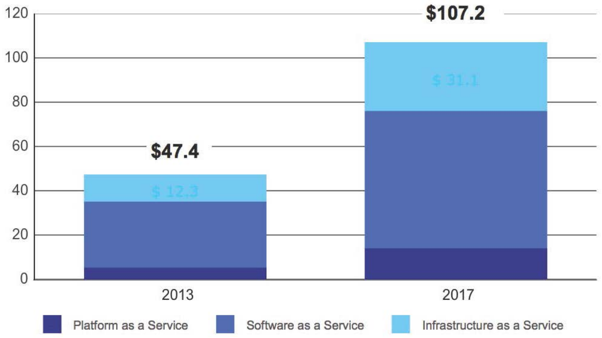


Figure 3.5: Worldwide Public IT Cloud Services Spending by Segment (in \$ billion) [27]

3.4.2 Cloud Service Providers

Shifting form only a few key players in the Cloud Computing market, today many cloud providers exist branching out in kinds of services and types of offerings. As software services, there are business management software, vertical apps, software security or some domain specific software in areas such as CRM. Platform services such as development and testing, integration, database, or application platforms are also widely offered by companies such as Microsoft, Oracle or Google. Other vendors such as Dropbox, Amazon, or Cisco offer cloud management, storage, virtualization, networking or computing services. The top and most known service providers are IBM, Salesforce.com, Amazon, Microsoft, Oracle, SAP, and Google. The vendors not only differ in the services they offer, but also in the way these services are offered, especially when we look at the management tools provided. Each cloud provider has strengths in a different manner. While some are simplistic easy to use others offer a more detailed and deeper granularity for more complex needs. For instance Amazon Web Services offers IaaS as well as PaaS and has a big set of other services, but if the business workflows and services grow in an organization, it becomes more difficult to manage all the services. To the contrary, Microsoft Azure has easy to use administration tools, but this may be insufficient for experienced users, which require a more granular control over the service. Google with its rich infrastructure is designed to scale, but is not easy to administrate [16]. Therefore, the customer has to decide, according to requirements and capabilities, which providers offer the suitable service for his or her business need.

3.5 Dimensions and Factors

Having analyzed the different cloud service models and offerings, as well as the adoption projections for the next years, two main trends are observed: the cloud offerings are getting more and more granular, tailored and designed for very specific use cases and, secondly, the adoption rate is growing even in business areas seen as too critical for cloud deployment until now, such as core business applications or business intelligence. To understand this trend, one must also understand the factors, which lead to such decisions. Although most studies contain charts showing only the top 5 to 10 reasons, which lead to cloud adoption, this paper tries to find and analyze all the decision factors which can have an impact on the decision process and categorizes them from a technical, an economical and an organizational perspective. In the second part of the analysis, we will also try to prioritize the factors based on the cloud service model (SaaS, PaaS, or IaaS).

3.5.1 Technical Factors

This subsection will give an overview of the technical factors identified as important by different research studies and analytical reports.

3.5.1.1 Scalability

Scalability is the property of a system to handle an increasing amount of workload or requests by being able to scale both horizontally (by adding new nodes system or computers to a distributed system) and/or vertically (by adding new resources to a single node of a system, *i.e.* adding more CPU's to a node) [35]. This is one of the most important requirement for cloud-based services and should, if possible, take place in an (semi) automatic demand-driven way, fully transparent for the end consumer. Although the term "unlimited" scalability is not exactly accurate from a technical perspective because data center have a physical limit in regards to the scalability of the resources, cloud systems are designed to handle almost any future usage peak the consumer might have.

To accomplish scalability, cloud vendors need to use a virtualization layer on top of the hardware layer, which is non-homogeneous in most of the cases. When services running on top of it demand more computing power to scale up, the virtualization layer adds new resources or pool of resources to respond to the increasing demand in a fully transparent way for the services. The service is only aware of the virtualized system it is running on, not of the different hardware used in the back-end. A lot of proprietary solutions were available in market (*i.e.* VMware), however, in recent years, a new community was founded to solve the issue of managing and controlling cloud resources from a non-homogeneous environment. With the start of the collaboration between NASA and Rackspace Hosting in 2010 [30], the OpenStack open source community was founded. Since then, a lot of major software vendors joined the community and are actively contributing to the creation of new modules and implementation of the standard in their own software stack. Another example is OpenNebula, an open source Cloud Computing platform for managing heterogeneous distributed data center infrastructures [31].

3.5.1.2 Elasticity

Although quite similar to scalability, elasticity defines ability to fit the resources needed to handle loads dynamically, normally through a scale out operation. When the workload increases, the system scales out by adding new resources, and when the load decreases, the system shrinks back and decouples the unutilized resources. This is especially important in a cloud environment, which implements a pay-as-you-use charging model, where consumers do not pay for resources which are not consumed as per the load requirements.

3.5.1.3 Availability and Reliability

Availability and reliability are two systems properties, which measure the system uptime. More specifically, availability is the probability that a service is operating properly when it is requested for use. Reliability on the other hand, "represents the probability of

components, parts and systems to perform their required functions for a desired period of time without failure in specified environments with a desired confidence"[33]. Although both seem to measure the same thing, a high availability does not necessary mean a high reliability. Imagine following example: A cloud service which is available 363 out of 365 days has an availability of 99.4%. If the system however crashes every 10 minutes for 1 second, it results in reliability less than 10 minutes, which is not high at all.
Because reliability is harder to measure and to predict, most of the cloud vendors focus more on the availability measure as a metric in their SLAs. According to the report [23], 82% of the questioned enterprise customers seek for service-level guarantee capabilities when it comes to availability, for both public and private cloud. A standard today in cloud business is to provide availability between 99% and 99.9%. Nevertheless, reliability should be part of a SLA, this is also reflected by the Cloud Service standardization guidelines published by the European Commission in 2014 [6].

3.5.1.4 Performance

Performance is rather a generic technical factor, but looking at recent reports [23] [24], it can observed that one of the top rated fears in the consumers perception is that cloud solutions are not able to support mission critical applications. Even if it is not decisive, performance, together with resource isolation, play a key role in running mission critical applications such as Enterprise Resource Planning (ERP) systems, which require a lot of tuning and configuration both on the software and hardware side.

3.5.1.5 Portability

While Cloud Computing might be attractive from a lot of viewpoints, one critical factor, which is not necessarily visible at a first glance is portability. Portability is the ability to move applications and data from one Cloud Computing environment to another with minimal disruption[12]. It can be further divided into data, application and platform portability. Consumers of cloud-based services may seek cloud portability so that they can migrate services to a new provider in response to a price increase or a breached service-level agreement[12]. In the context of hybrid cloud environments, portability is defined as the ability to move between a public to a public cloud solution. This requires interoperability among the different cloud providers. As previously mentioned, a lot of cloud providers started building up on a common set of standards, the most notable being OpenStack initiative.
Another important aspect in regards to portability, which is often overseen, is the portability of the licenses. While it might be technical feasible to port an application or service from one cloud environment to another, this might cause a substantial effort on the license side. Software licenses are often bound to users or to the underlying hardware, so porting an application, such as a database for example, from one cloud environment to another can cause substantial additional license costs. In extreme cases, the software licenses have to be purchased again.

3.5.1.6 Provisioning Flexibility

Strongly related to the factor above, provisioning defines the ability to prepare and deliver a new service or resource to be used by an end user. This can range from creating a new user account to delivering a new pre-configured system for software development. A system can also consists of multiple services, such as computing power, storage, network, databases etc. Having the right provisioning tools to provision and combine different cloud offerings (*e.g.* IaaS combined with PaaS and Data-as-a-Service), as well as being able to perform such operations in a shorter period of time can prove itself as a real advantage when it comes to optimize operational IT costs.

3.5.1.7 Security and Data Privacy

Maybe the most disputed fact, especially in the recent years since the NSA scandal, is security and data privacy. While security is also covering the authentication and authorization aspect of services, which can be accomplished through integration with identity management tools, this section will focus more on data privacy. Data privacy can be viewed from a technical and a legal perspective. From a technical perspective security and privacy can be ensured using a wide set security mechanisms on the transport and storage layers, such as (strong) authentication and authorization, data encryption, use of SSL and virtual private networks (VPN).

Ensuring data privacy on the legal level however can be more challenging, because cloud providers must comply with the legislation of the country where they operate and/or with the one in which they are legally registered. The overall opinion about privacy is that there is no real protection of privacy against powerful intelligence organizations such as the NSA, or as Darlene Storm from ComputerWorld is writing “If NSA wants your cloud data ‘be big boys about it’”[15]. Additional factors, such as the recent ongoing process of the US government against Microsoft to release data held outside of the US in Ireland [29], add more insecurity to the consumers. This is why for many government agencies and financial institutions, going public cloud is a no-go for systems containing sensitive data. An alternative in such situations is to use a private cloud solution. A growth trend towards private cloud solution is also reflected in the forecast done by IDC [27].

3.5.1.8 Geographic Location

Related to the previous factor, the geographic location of the cloud provide can also have a impact on the decision regarding the migration to cloud-based services. Geographic location of the cloud service provider or its data-center can bring additional assurance on the data privacy aspect (*i.e.* having a data center in Switzerland is being seen as more safe for Swiss companies against US extraterritorial jurisdiction because of the current data protection legislation [25]) and harmonize compliance issues the organization may have, depending on the industry the organization is active, such as the pharmaceutical, financial or insurance industry.

3.5.1.9 APIs, Openness of Service and Use of Standards

Application Programming Interfaces (API) are a set of routines, protocols, and tools [37] for building, integrating and monitoring software applications. As consumers often don’t have unlimited (root) access to the cloud system on all levels (hardware, operating system, network, storage etc.), exposing APIs for enabling integration with other on-premise or cloud solution and for monitoring and control purposes is crucial for every project deployed in the cloud. Since organizations tend to keep their most sensitive customer data on premise, integration of cloud systems with on-premise systems is unavoidable and the APIs enabled by the cloud provider play a central role. However, exposing unwanted or unneeded APIs, which could result in remote connections to the system, can also lead to compliance issues for organizations.

Another important factor in this context is openness. One of the factors seen as a barrier to enterprise cloud adoption is data or vendor lock-in [24]. This can happen, if the cloud provider does not provide any means to extract or migrate the data after the end of the contract or if the costs and complexity associated with this process are very high. This can be avoided through the use of industry standards (*e.g.* OpenStack), both on the provider and consumer side.

3.5.1.10 Data Transfer

In many cases, when large amount of computing power is needed for a short period of time (such as running a large Hadoop job for a couple of days on a big data set), one crucial factor is the ability to transfer the data on the cloud storage servers. Let us suppose that for example a big media company wants to run a Hadoop job in a cloud on a data set of 100TB. If the transfer rate is only between 1 and 2 MB per second, it will cause more time and costs to transfer the data and run it on the cloud than locally. This has been also analyzed by the study of [18], which states, that depending on the use case and the complexity, the data transfer rate can turn into a bottleneck and it can be costly to run the service in cloud than running the service on premise.

3.5.1.11 Tools

When using cloud-based services, tools are always needed when it comes to application deployment or configuration, administration and monitoring of the services. Most of the cloud-based services today are self-service. This is why the ease of use, the stability and performance of these tools play a central role in the cloud adoption.

3.5.1.12 Other Technical Factors

Other identified technical factors that are not covered in detail by this paper but play a role in the decision process, are application functionality, upgrade and patching frequency, multi-tenancy data isolation among others.

3.5.2 Economical Factors

Our second category of analyzed factors is the economical factors. While most technical factors can be represented as checklists that can be answered with “yes” or “no”, economical factors can be quantified, measured, and form the basis for creating business plans and costs analysis over specific period of time. These factors, along with the “fine- grained economic models enabled by Cloud Computing make trade-off decisions more fluid” according to [1].

3.5.2.1 Charging Model

Pricing and costs have become the strongest argument in favor of using Cloud Computing. This is reflected in the Everest Group Cloud Adoption Survey in 2013 [24], which positions the Total Cost of Ownership (TCO) reduction as the most important driver for cloud adoption. Adopting a cloud strategy however does not always lead to a better TCO, as shown by the study done by OVUM [5]. Nevertheless, the study illustrates that the costs of moving to the cloud are in many cases lower than running the system on-premise, depending on the organization’s size and the complexity of the used software. Some of the factors, which lead to a lower TCO, are the elimination of the upfront investments in hardware and software and the use of a dynamic, pay-per-use charging model. Combining such a charging model with the resource elasticity offered by cloud-based services makes Cloud Computing a lot more advantageous because organizations do not need to over-provision resources for handling unpredicted demand and costs will be calculated on a per usage metric. But as we look at the offering in the market, we observe that the pay-per-use model is mostly used when it comes to IaaS or computing power. For PaaS or SaaS vendors often charge per user subscription, per environment or through so called "credits", virtual currencies that can be used on the different cloud offerings the vendor has. This can result in overall higher costs for organizations.

3.5.2.2 TCO - Total Costs of Ownership

Part of every business case, TCO defines total costs of an investment over a certain period of time. In IT, the TCO includes hardware and software acquisition, management and support, communications, end-user expenses and the opportunity cost of downtime, training, and other productivity losses [36]. We will not go into the details of calculating a TCO, but more on the role that the cloud service model plays on the TCO. In some cases, depending on the organization's size and the complexity of the application, investments in cloud-based services can lead to a lower TCO than owning the hardware and software in house [5]. As the study shows, companies, which require applications with less complexity and a small technology footprint, achieve a better TCO when using cloud-based services than owning and managing the applications in-house. This changes when organizations require more complex applications or services with a larger technology footprint. More complex functionality translates in higher subscription and training fees, and, depending on the number of subscriptions, can lead to a higher TCO compared to in-house hosting. Therefore, a TCO calculation has to be made on a case-by-case basis.

3.5.2.3 CapEx to OpEx

As described in [1], "The economic appeal of Cloud Computing is often described as 'converting capital expenses (CapEx) to operating expenses (OpEx)'". To better understand this, the cost structure of IT projects has to be analysed. IT projects often require a high initial investment, such as software licenses, hardware investments and consulting services for setting up the environment. This is what we call capital expenditures on fix items. In addition to this, a project also includes expenses consisting of implementation costs, such as development, configuration, or testing, maintenance and annual support fees for hardware, software, and services for period of years, until the solution is retired or replaced. Operating costs should also include indirect costs, such as salaries. These costs are calculated on yearly basis and normally go into operational expenses (OpEx). Because cloud-based services are charged on a pay-per-use or user subscription model, initial CapEx is much lower. The absence of up-front capital expense allows capital to be redirected to core business investments [1]. The costs associated with running a service will be mostly operational monthly costs and this will enable organizations either to reduce total TCO for the investment or have a better spread of the costs across a period of time. According to studies, only 33% of the IT budgets are spent for innovation and business opportunities [17], the rest is used only for integration and maintenance of ongoing services, so this can be a decisive factor in a lot of cases where the organizations don't have a enough IT budget for investments.

3.5.2.4 License Costs

As organizations buy software from different vendors, they end up owning big amounts of software licenses. However, there is no standard in regards to the usage of a license. Each vendor has its own license model, metrics and restrictions, licenses are normally bound either to users, to hardware or to a specific period of time. Things can get more complicated as organizations try to move their existing licenses from their on-premise infrastructure to a cloud environment, public or private. In some cases there are some license migration paths offered when migrating the licenses to a cloud environment of the same vendor. If this is not the case, it can either cause consistent additional costs or a compliance inconsistency, which can lead to legal repercussions.

3.5.2.5 Contracts and Service Level Agreements (SLAs)

In every contract related to a service between two or more entities, service level agreements play a major role. SLAs, in general, define the service scope, service quality and responsibilities between the service consumer and the service provider and are enforced through rewards or penalties. During a transaction, such as a sales contract, between two parties, information asymmetry can exists. “In contract theory and economics, information asymmetry deals with the study of decisions in transactions where one party has more or better information than the other. This creates an imbalance of power in transactions, which can sometimes cause the transactions to go awry, a kind of market failure in the worst case” [28]. Assuming there is no information asymmetry, because cloud service contracts are relative new, often both the cloud providers and consumers lack the experience in defining the proper SLAs. “The structure and negotiation of an IS outsourcing contract cannot be based on a zero-sum game philosophy; a win-win philosophy is more appropriate” [10].As described in the research of [10], “estimates of the outsourcer’s business value for the different performance levels and the corresponding vendor cost are often not known by either outsourcer or vendor”. The study offers one approach with several steps to calculate and define proper SLAs for reducing the risks on both the consumer and vendor side. The steps defined by the study are: definition of the business value analysis, definition of the performance levels, estimating the value of each performance level and estimating the costs of the each performance level.

3.5.3 Organizational Factors

The last category of factors analysed by this paper are organizational factors. This category is also not measurable or quantifiable, but shows the impact on the organization when deciding to shift to cloud-based services. Such factors shift the organization dynamic strongly enabling them to change the way IT related decisions are made.

3.5.3.1 Technical In-house Resources

Organizations no longer require networking components, storage or processing power in the measures that was required before. Since the resources are constantly required and the services are costly, some organizations only own the minimal required resources in-house and use the cloud in order to scale. Depending on their needs, they can either not have these resources at all in cases where the software is delivered as SaaS choose certain services being delivered in house for security and integrity reasons.

3.5.3.2 Organization Size

Due to the lack of required development, maintenance, and configuration of resources, organizations do not need the staff to undergo these procedures. If we consider the Pinterest example that has been mentioned before. With only 12 employees they managed to scale their business without wasting resources on staff to maintain the data centers or develop and test their applications [21].

3.5.3.3 Decision Factor

Since the organizations lack the need to staff their departments with IT specialist and consultants, IT related decisions involve more and more non-technical and business units.

The chief marketing officer (CMO) is a key stakeholder in decisions regarding cloud adoption since many cloud-based services are related to marketing and sales operation such as ERP, CRM, BI and E-Commerce (see Figure 3.6) [24]. According to a Gartner analyst, by 2017, CMOs would spend more on IT than CIOs. [2] IT teams embedded in business units are yielding more decision power regarding such choices. Such involvement of business users forces service providers change their orientation of how they perform and market their offerings [24].

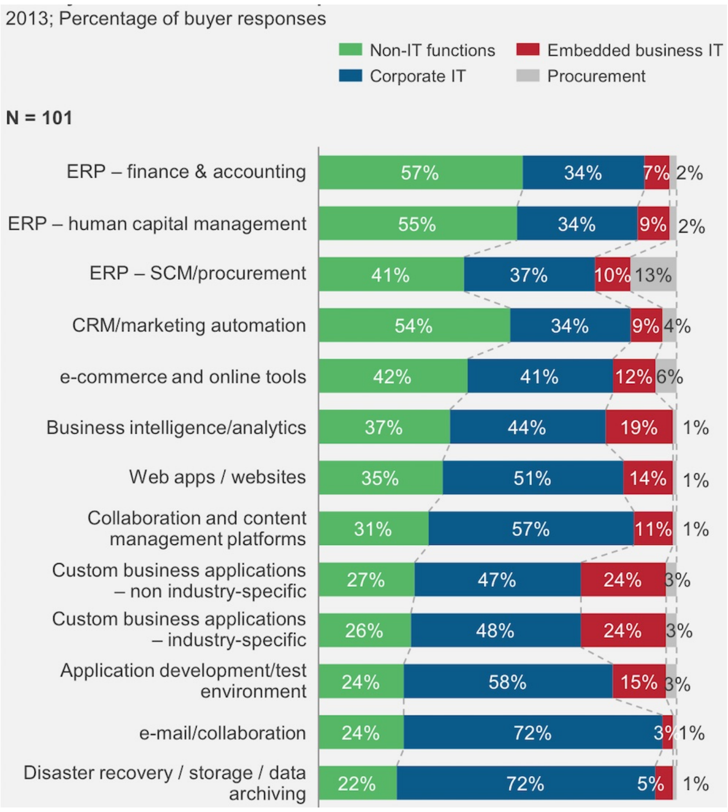


Figure 3.6: Primary Decision Makers for Enterprise Workloads/Solutions [24]

3.5.3.4 Organizational Agility

While there are multiple definitions of agility, we would define it as the ability of an organization to rapidly react and adapt to new changes coming from inside the company or from the outer environment. With the development of new technologies to an unprecedented pace, (re-)acting fast with new products and offerings can be critical for the organization's survival. Cloud-based services enable a faster provisioning of services and at the same time require a much lower workforce for managing the IT infrastructure. This can reduce the implementation time for projects, giving companies a greater agility towards the unknown.

3.6 Evaluation and Discussion

After identifying all the factors, which play a role in the decision process, the next step was to analyze which offerings are most affected by the different factors. The results are displayed in the Table 3.1. Although some of the factors do not apply on all the offering models, we could not find a notable difference based on this.
An interesting fact that has been noticed while conducting this report was that the technical factors were mostly seen as fears, whereas most of the organizational, and economical

factors are seen as opportunities or benefits. This, however, shows that the most perceived disadvantages of the cloud-based services now are of technical nature. Although it may represent an issue on the short term, this may turn into an opportunity or into a business differentiator for the different cloud providers on the long run. With technological advancements and new achievements in the field of Cloud Computing, such technical factors may be overcome in the future. For the time being, some technical factors have workarounds, that could be used until a better solution is presented [1]. In contrast to such factors, economical and organizational factors are not limited by technology, but rather by business administration. Looking at this from a long-term perspective, technical issues and bugs are rather easy to fix and could later on turn into benefits. Since Cloud Computing is still considered as a new technology, the dynamic of the field is shifting. While some problems could be solved, others emerge and the importance and priorities of the factors shift. The change in the importance of the different factors over a longer period of time can be analyzed in future work.

The reason why economical and organizational factors are seen as opportunities is more complex. Firstly, these are factors which are mostly weighted by non-technical people (*e.g.*finance, marketing), meaning that budget cuts may be considered more important than data safety. Secondly, they enable organizational efficiency both in human and in financial resources by downsizing the organization with regards to employees as well as required in-house resources, which my be used through the cloud.

The factors mentioned in the report have been described, prioritized, and categorized in to three levels of importance from an adoption point of view (see. Table 3.1). The digits 1,2, and 3 denote high, medium, and low priority, respectively. They have also been differentiated into fears and benefits, annotating if the factor is considered as a disadvantage or an advantage. Looking at the rank, the technical factors with the highest priority are those related to scalability, portability, integration, and data security/isolation. On the economical side, the factors with highest priority are those related to the costs: OpEx, SLAs and TCO. This means that the decision whether to move or not to move to the cloud has to be done, or more precisely, calculated, on a case by case study and that there is no rule of thumb.

3.7 Summary and Conclusion

In this paper the main factors, which play a role in the decision process of moving to the cloud were identified and listed. They were also categorized in technical, economical and organizational factors. These three types of factors were chosen, because they cause the most impact on the cloud adoption decision. Nevertheless there are other influential factors like social and environmental factors, which are not mentioned in the report(see [20]). After analyzing the different factors and reading related work, the report concludes that there is no one single correct answer indicating whether to move to the cloud or not. Every case has a different situation, different attitude towards the mentioned factors and different business limitations to consider for a cloud adoption.

Although there is no right answer, the mentioned factors listed in this report help organizations to give an overview of the most important factors to consider while moving to the cloud. It thereby helps them determine their priorities and evaluate accordingly whether to use cloud-based services or not. Since the disadvantages and threats of Cloud Computing are mostly technical factors, cloud users can be assured, that these factors could be more or less overcome with technological advancement. As of the question "To Move or Not To Move", there is no definitive answer, but a set of factors that offer an overview to help the organizations with such a decision.

Table 3.1: Summary of Technical, Economical, and Organizational Factors

Type	Factor	Description	Benefit	Fear	Priority	IaaS	PaaS	SaaS
Technical	Scalability	Ability to scale horizontally and vertically (should actually be transparent). On demand scalability based on the workload.	x	x	1	x	x	x
	Availability & Reliability	High availability (Application Clustering) Networking, redundancy of power, Internet connection, cooling systems, fire suppression systems, servers, storage, and security systems)	x	x	1	x	x	x
	Security, Data confidentiality and Identity Management	Communication Security (SSL) etc., Data encryption and security etc.		x	1			x
	Elasticity	Flexible Infrastructure/Resource Capacity	x		1	x	x	
	Performance	Achieve the performances same as on premise, support mission critical applications	x	x	2	x	x	x
	Interoperability and Portability	Ability to deploy application from private to public cloud and back, use of same software on both public and private cloud. Ability to integrate cloud solutions		x	1	x	x	
	APIs, openness of service and use of standard	Ability to integrate the applications in the cloud with other cloud software or on premise applications, customization, ability to export data after end of cloud subscription etc.		x	1	x	x	x
	Tools to deploy the applications in the cloud	Tools to deploy, administer and monitor applications in the cloud.		x	1	x	x	
	Provisioning flexibility	Time to provision new applications / servers, flexibility in combining the different cloud offerings from a vendor, templating etc.		x	2	x	x	x
	Geographic Location	Location of the data servers		x	2	x	x	x
	Application functionality	Moving from on premise / desktop applications to cloud applications often comes with a decrease of functionality		x	2		x	x
	Automatic Upgrades & Patches	How many times per year are upgrades performed (including security upgrades), how does it affect customizing?	x		3	x	x	x
	Data Transfer Bottlenecks	How fast can be data transferred / migrated to the cloud.		x	3	x	x	
	Complete isolation & multi-tenancy	Isolation on application / security and infrastructure level	x	x	1	x	x	x
Economical	Charge model	Pay per use, user subscription	x		2	x	x	x
	License costs	How to use and/or migrate existing licenses on cloud infrastructure	x		2	x	x	
	CapEx to OpEx	"Including direct - cloud service costs, software licenses, migration costs, data transfer (networking) costs, Indirect costs - salaries, usability, quality "	x		1	x	x	x
	Contracts and SLAs	Penalties and incentives	x	x	1	x	x	x
	TCO reduction	TCO over 3 to 5 years	x		1	x	x	x
Organizational	Decision factors inside organizations	CIO vs. CxO's	x		1	x	x	x
	Size of organization	Number of user licenses/subscription	x		2	x	x	x
	Organizational agility		x		1	x	x	x
	In-house resources	Resources for managing internal IT	x		2	x	x	x

Bibliography

[1] M. Armburst, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia: *Above the Clouds: A Berkeley View of Cloud Computing*, EECS Department, University of California, Berkeley, Technical Report, Electrical Engineering and Computer Sciences, University of Berkeley, February 2009.

[2] L. Arthur, *Five Years From Now, CMOs Will Spend More on IT Than CIOs Do*, Forbes, February 8, 2012 `http://www.forbes.com/sites/lisaarthur/2012/02/08/five-years-from-now-cmos-will-spend-more-on-it-than-cios-do/`, last accessed November 2014.

[3] M. R. Benioff, Behind the Cloud, Salesforce.com, `http://www.salesforce.com/behindthecloud/`, last accessed on December 2014

[4] G. Cattaneo, *The Demand of Cloud Computing in Europe: Drivers, Barriers, Market Estimates*, IDC, Research In Future Cloud Computing workshop, Bruxelles, May 2012.

[5] K. Dawson, *The Total Cost of Ownership of Cloud and Premise-Based Contact Center System*, OVUM, January 2013.

[6] N. Editor, *Cloud Service Level Agreement Standardization Guidelines* , European Commission, Brussels 2014, `http://ec.europa.eu/digital-agenda/en/news/cloud-service-level-agreement-standardisation-guidelines` last accessed December 2014.

[7] P. Mell and T. Grance: *The NIST Definition of Cloud Computing*, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, United State, September 2011.

[8] B. Mitchell, *VPN Tunneling*, abouttech, `http://compnetworking.about.com/od/vpn/a/vpn_tunneling.htm`, last accessed December 2014

[9] A. Mohammed, *A History of Cloud Computing*, Computer Weekly, March 2009, `http://www.computerweekly.com/feature/A-history-of-cloud-computing` , last accessed December 2014

[10] K. Osei-Bryson and O. K. Ngwenyama, *Managing Risks in Information Systems Outsourcing: An Approach to Analyzing Outsourcing Risks and Structuring Incentive Contracts*, European Journal of Operational Research, 174, 245-264, January 2014.

[11] A. Regalado, *Who Coined Cloud Computing*, MIT Technology Review, October 2011 `http://www.technologyreview.com/news/425970/who-coined-cloud-computing/` last accessed November 2014.

[12] M. Rouse, *Cloud Portability*, TechTarget, http://searchcloudprovider.techtarget.com/definition/Cloud-portability, last accessed December 2014.

[13] M. Rouse, *Search Cloud Computing - Hybrid Cloud*, TechTarget, http://searchcloudcomputing.techtarget.com/definition/hybrid-cloud, last accessed December 2014.

[14] D. M. Smith, *Hype Cycle for Cloud Computing, 2014*, Gartner, July 2017, https://www.gartner.com/doc/2807621/hype-cycle-cloud-computing-, last accessed December 2014

[15] D. Storm, *No cloud privacy or security: If NSA wants your cloud data 'be big boys about it'*, ComputerWorld, June 16 2014, http://www.computerworld.com/article/2476331/cloud-security/no-cloud-privacy-or-security--if-nsa-wants-your-cloud-data--be-big-\\boys-about-it-.html, last accessed December 2014.

[16] D. Sullivan, *IaaS Providers List: 2014 Comparison And Guide*, tom'sIT PRO, February 14, 2014 http://www.tomsitpro.com/articles/iaas-providers,1-1560.html last accessed December 2014.

[17] S. Ranger, *Here's what your tech budget is being spent on*, ZDNET, November 2014, http://www.zdnet.com/article/heres-what-your-tech-budget-is-being-spent-on/, last accessed on December 2014

[18] B. C. Tak, B. Urgaonkar and A. Sivasubramaniam *To Move or Not to Move: The Economics of Cloud Computing*, Proceedings of the 3rd USENIX Conference on Hot Topics in Cloud Computing, HotCloud'11, USENIX Association, Portland, 2011.

[19] J. Varia and S. Mathew, *Overview of Amazon Web Services*, Page 10, January 2014 https://media.amazonwebservices.com/AWS_Overview.pdf last accessed November 2014.

[20] E. O. Yeboah-Boateng and K. A. Essandoh, *Factors Influencing the Adoption of Cloud Computing by Small and Medium Enterprises in Developing Economies*, International Journal of Emerging Science and Engineering (IJESE), February 2014.

[21] AWS Case Study: Pinterest http://aws.amazon.com/solutions/case-studies/pinterest/, last accessed December 2014.

[22] Citrix Cloud Survey Guide, AWS, August 2012, https://s3.amazonaws.com/legacy.icmp/additional/citrix-cloud-survey-guide.pdf, last accessed December 2014.

[23] Cloud Computing - Customers Speak: Cloud Needs Guarantees, ComputerWorld Research Report sponsored by Oracle, September 2014.

[24] Everst Group Research, Enterprise Cloud Adoption Survey 2013: Summary of Results, March 2013.

[25] Guide to Cloud Computing, Federal Data Protection and Information Commissioner (FDPIC) - 2014, http://www.edoeb.admin.ch/datenschutz/00626/00876/01203/index.html?lang=en, last accessed December 2014.

[26] Gartner IT Glossary, http://www.gartner.com/it-glossary, last accessed December 2014.

[27] IDC Forecasts Worldwide Public IT Cloud Services Spending to Reach Nearly 108 USD Billion by 2017 as Focus Shifts from Savings to Innovation, Press Release, http://www.idc.com/getdoc.jsp?containerId=prUS24298013, last accessed December 2014.

[28] Information Asymmetry, Wikipedia, http://en.wikipedia.org/wiki/Information_asymmetry, Last accessed on December 2014

[29] Microsoft 'must release' data held on Dublin server, 29 April 2004, BBC online, http://www.bbc.com/news/technology-27191500, last accessed December 2014.

[30] OpenStack, Wikipedia http://en.wikipedia.org/wiki/OpenStack, last accessed December 2014.

[31] OpenNebula, Wikipedia http://en.wikipedia.org/wiki/OpenNebula, last accessed December 2014.

[32] Pinterest, http://pinterest.com/, last accessed December 2014.

[33] Reliability Basic, http://www.weibull.com/hotwire/issue26/relbasics26.htm, weibull.com, April 2003, last accessed December 2014.

[34] SuccessfulCloud Partners HIGHER, FASTER, STRONGER, An IDC InfoDoc Sponsored by Microsoft, July 5 2013.

[35] Scalability, Horizontal and Vertical Scaling, Wikipedia, http://en.wikipedia.org/wiki/Scalability#Horizontal_and_vertical_scaling, last accessed on December 2014

[36] Total Cost of Ownership (TCO), Gartner, http://www.gartner.com/it-glossary/total-cost-of-ownership-tco, last accessed December 2014

[37] Wikipedia API definition, Wikipedia, http://en.wikipedia.org/wiki/Application_programming_interface, last accessed November 2014.

Chapter 4

Applicability of Cryptographic Protocols to Support Service Level Agreements

Mohit Narang

This chapter analyses the problems faced by SLAs, in context of cloud services. It is based on the blockchain technology implemented as a part of crypto currency protocols. First part explains fundamental concepts of SLAs and inherent problems. Then, blockchain and Ethereum concepts are introduced. Next, also proposing how some of the problems being faced by SLAs can be solved with the presented technologies. Finally, some directions about future work.

Contents

4.1 Service Level Agreements

This section defines a general SLA in terms of a customer and a service provider where service being provided is one of the cloud services. An SLA is an agreement between the service provider and the customer for a cloud service. It defines terms of agreement between the user and service provider of a Web-based utility service. Customer and service provider both agree on some common parameters based on the contract.

4.1.1 Introduction

Current web services and applications, which rely on cloud infrastructure and services, require the measurement and monitoring of quality and performance of these core cloud services. A fair amount of effort had been made by organizations in the past to define parameters to monitor services and how to measure those parameters. These parameters are important for performance and quality of core infrastructure services on top of which most IT service providers build applications. One of the most comprehensive definitions of an SLA in domain of computing as defined by Farrell [6] makes it clear that there have to be a set of norms or parameters which define the key performance indicators for the quality and performance of a service.

This report makes some assumptions about SLAs considering a use case with Amazon EC2 (Elastic Compute Cloud) [7], an infrastructure as a service offering by Amazon. EC2 is a web service, where one can request virtual hardware machines for computing. It has different sizing options where customers can configure the offerings according to their requirements. One of the key parts of an SLA as defined by Amazon EC2 [8] is virtual machines' uptime. Amazon guarantees a 99.95 % uptime and anything between 99 and 99.95 % gives the customer 10 % credit back and anything between 95 % and 99 % gives 30 % credit back as part of the SLA between customer and Amazon. The customers can use this credit in future to pay for EC2 usage.

4.1.2 Compensation

Whenever two companies agree over the fulfilment of a contract, they agree upon a set of rules which define the constraints of an SLA. In the same agreement the companies also agree on how to deal in a situation when the agreed rules are not met and there is an SLA breach. In cloud computing scenarios, most SLAs are defined in terms of percentage credit back or bonus credit for future. This fits well in the current internet economy model, as the providers can keep the customers in future for little costs, the customers can drive their costs down. The most challenging problem is the way SLA compensations are dealt with by the companies. For example, in the event of an Amazon EC2's SLA breach, the customer should open a support case with the customer service desk. The case is then forwarded to verification and enters a long workflow until the customer service provides back an answer. Moreover, it is up to the service provider to provide compensation, and there is no alternative for the customer in case of a disagreement. This entire model is thus time consuming and cumbersome, both for the customers and service providers.

4.1.3 Enforcement

Enforcement of SLA pertains to fulfilment of agreement between parties over breach of an SLA. It takes into account the parameters and measurements agreed upon at the inception of SLA. After the evaluation of measurements, payments or necessary steps are taken by responsible party to fulfil the conditions of SLA. If the parties don't reach an agreement the case is forwarded to a third party or court for evaluation.

4.1.3.1 Problems of Enforcement

Enforcement of SLAs are subject to issues like disagreement between parties or undefined issues. An example of such a case from 2010 [9] shows how cumbersome enforcement can be. It was a case of SLA breach between an online retailer American Eagle using a hosting provider IBM. There was a database outage for 8 days during which American Eagle lost more business than the compensation from the SLA penalty.

4.1.4 Monitoring

Whenever two parties agree on an SLA there need to be valid facts they can both use to determine whether an SLA is breached. Monitoring of the SLA parameters is one of the most challenging tasks. Leaving any technical complexities aside and sticking with a simple example of an uptime SLA:

4.1.4.1 Possible Responsible Stakeholders to Perform Monitoring

- Customer
- Provider
- Third Party

4.1.4.2 Problems Of Monitoring

Monitoring of web services is a challenging task due to factors like network connectivity, response time, hardware architecture of service provider/customer, etc. Whenever a parameter is chosen to be monitored for an SLA, these factors kick in to affect the readings depending upon which party monitors. For an example [10] Netflix uses Amazon Web Services as network and compute service provider to distribute media content to its end users. If an end user is seeing a lag while watching a movie there can be different reasons:

1. Compute servers at Amazon Web Services are not able to process the request.
2. Network service at Amazon Web Services is congested
3. Network service at end user is congested
4. Network hardware at the end user is not capable to sustain streaming requirements

Netflix as a customer needs to make sure the problem is not due to its service provider, Amazon. It is difficult for Netflix, as a customer of Amazon's cloud service, to decide how to monitor the performance of services provided by Amazon.
Generalizing the above example, monitoring at each end have different problems:

- **Customer end:** If monitoring is set up at the customer's end it can misreport as misreporting is better strategy for the Customer from an economic point of view.
- **Service Provider end:** If monitoring is set up at the Service Provider's end it suffers from the similar strategic exploitation.
- **Trusted third party:** Possible solution, but can be swayed by Customer and/or Service Provider.

Modeling the above as a case of game theory, each party wins the game if it mis-reports. Both the parties will always misreport as there is no loss of misreporting and positive economic gain in result of winning the game. Hence from an Economics viewpoint it is a dominant strategy to mis-report the results of monitoring and hence no party can be trusted.
A viable solution to tackle all possible cases is challenging and not yet widely implemented.

4.1.5 Motivation

Given these problem spaces the report identifies the need for new solutions:

1. Solve long time lags in Compensation of SLAs due to administrative overhead.
2. Reduce disagreements during enforcements.
3. More acceptable ways to monitor SLAs where parties do not need to trust each other and can trust the monitoring technology.

4.2 Cryptographic Protocols

The term Cryptographic Protocols describes a group of protocols that were initially developed for digital communication security. Basic guiding principles for these protocols were:

4.2.1 Smart Contracts

An application of Cryptographic Protocols that was already widely discussed before the recent introduction of Bitcoin are smart contracts. This report describes two examples of Cryptographic Protocols that are later on applied to support SLAs. A wide variety of new cryptographic protocols have emerged in recent years. The most traditional kind of cryptography is secret key, in which Alice and Bob (our exemplar parties to a smart contract) use a single shared, prearranged key to encrypt messages between them.
With the publication of the white paper on the Bitcoin protocol in 2009 there has been a surge of new ideas and applications of the blockchain technology that bitcoin is based on. In the following sections the report dives deeper into this technology that Bitcoin introduced and then also looks into Ethereum, another concept of a cryptographic protocol that expands on some ideas that would allow for more sophisticated smart contracts. Bitcoin introduces a solution to the Byzantine Generals Problem [5].
List of other Cryptographic Protocols that we are aware of but didn't look into:

1. ShellingCoin
2. Counterparty
3. MasterCoin

4.2.2 Bitcoin

In this section the blockchain technology is explained, which enables the usage of Bitcoin for services like Escrow and Multisig transactions. Next, Ethereum platform is explained which is built on top of the fundamental concepts of blockchain technology.

4.2.2.1 Blockchain Technology

Blockchain is a public ledger of transactions with time stamps. It can be considered as the backbone of cryptographic currencies like bitcoin. It makes sure that no user can double spend the coins. The bitcoin network is a distributed set of nodes, where each of the nodes contain a copy of the blockchain. There are a set of consensus rules, which are followed by the nodes and once some of the nodes have the same blocks in blockchain, they are said to be in consensus. A set of transactions are put in a block after hashing the transactions. On continuous hashing, this block gives a merkle root, which is part of the block header that contains the information of each block. The bitcoins are stored in transactions and not in wallets which means one can only spend if they receive bitcoins from previous transactions. Due to this structure of keeping the balance of accounts its impossible to double spend the coins, as whenever someone spends coins they are merely using the output of a previous transaction as an input of a new transaction to spend coins. Difference between the input and output usually goes to the miner, as a fee to include the transaction on a block. Proof of work is what makes the blockchain a trustworthy ledger since it becomes extremely difficult to modify a block by any dishonest node, as the number keeps increasing with time. Difficulty of calculating the hash is set dynamically based on the number of blocks added in two weeks time. If the time taken is lower, the difficulty is set higher, otherwise its lowered proportionally. Any miner can add a block to the blockchain, if the target threshold can be hashed. Blocks are addressed based on their distance from the block 0 or genesis block. Each block contains at least one transaction. First transaction is a coinbase transaction or generation transaction, which is responsible for the block reward. A condition for a coinbase transaction is that it cannot be spent in the next 100 blocks, which restricts the spending of coinbase transactions from stale blocks. A transaction is comprised of following parts:

1. Version: Four byte number which links the set of consensus rules to be used, to validate a particular transaction, so that if in future the rules change, the transaction should still be valid.

2. Inputs: Output of previous transactions to use to spend existing unspent transactions.

3. Outputs: Amount of bitcoins to spend.

4. Locktime: Unix timestamp or block number at which the transaction is locked.

Transactions use the standard Pay to public key hash (P2PKH). This type of transaction can be used by a user to send bitcoins to a Bitcoin address. To generate public-private key pair Bitcoin uses Elliptic Curve Digital Signature Algorithm (ECDSA). The public key is hashed and provided to spending party as a bitcoin address, which are usually base58-encoded strings. The user can now create a transaction with the decoded bitcoin address such that anyone with the private key can spend the bitcoins associated with that particular transaction. After the transaction has been broadcasted to the network it appears as an unspent transaction in the receiver's wallet software. Now whenever the receiver wants to spend the above unspent transactions he needs to use the signature script in any new transactions, signing those transactions with his private key. These new transactions are validated by miners and once validated, become part of the blockchain. P2PKH scripts are validated using the evaluation of both pubkey and signature scripts. Another way of transactions using public private key pairs is using pay to script hash (P2SH) transactions where a spender can also add a redeem script. They are more simple and equally secure as P2PKH hashes.

Standard transactions Transactions which use believed to be safe templates for the pubkey scripts and signature scripts and pass a test IsStandard() are standard transactions. For core version 0.9 the standard transactions can be

1. Pay to public key hash (P2PKH): used by most of transactions
2. Pay to script hash (P2SH): used mostly for multisig
3. Multisig: advantages over P2SH as can have minimum m of n number of public keys to match.
4. Signatures must be in same order as public keys.
5. Pubkey: it is an abridged version of P2PKH, but less secure.
6. Null Data: used to create transactions only to add data to the block chain.

Non-standard transactions Any transaction which is not using the standard transaction types. By default these transactions result in error unless the script hash is added as a redeemable script. In which case it will result in an unspendable transaction. Following conditions must also be true for valid transactions :

1. Locktime must be in past or sequence numbers must be 0xffffffff.
2. Size must be less than 100KB
3. Inputs must be less than 500 Bytes
4. Signature script must push just the data to evaluation stack.
5. No outputs should have bitcoins less than minimum defined.

A signatory can sign any part of the transaction. Depending on what is to be signed, there are three kinds of SIGHASH available for a signatory:

1. SIGHASH_ALL : signs everything except signature scripts.
2. SIGHASH_NONE : signs inputs but not the outputs
3. SIGHASH_SINGLE : signs input and single output so that other signers can control their outputs.

The above base types can be modified with flags: SIGHASH_ANYONECANPAY:
When combined with SIGHASH_ALL : means anyone can change who contributes to the transaction but no one can edit where the output goes and how much of bitcoins are sent in the transaction.
When combined with SIGHASH_NONE : anyone can spend however they like.
When combined with SIGHASH_SINGLE : anyone can spend however they like.

The locktime is an important part of every transaction. It defines when a transaction can be spent or made part of the block chain. It helps users to create future transactions such that users are allowed to change a transaction later on. These transactions can be added two hours before the actual locktime expiry, due to which a cancellation of such a transaction is not possible after 2 hours before expiry time. If the locktime is less than 500 million then its considered as a block height where transaction have to be broadcast. Otherwise it is used as a unix timestamp.

Transaction Fees Transaction fee of a transaction depends on the size of the signed transaction and is given by the miner. It is a decision of the miner to choose their minimum fee to include a transaction to the blockchain. 50KB of each block is reserved by the peers for high priority transactions which spend coins which did not circulate for long time. The remaining block is filled according to the preferential filling of the block, based on the amount of fee per transaction in the rest of the transactions. There is also a minimum fee, a transaction needs to pay to be broadcast. The minimum fee only applies to the transactions which are not high priority. Since each unspent transaction, which is being spent as a part of a new transaction, have to be spent completely in a transaction, there are usually two outputs. One output to the receiver and the other as the change output which sends the coins back to the sender.
Since in a transaction both the sender and receiver can see the public keys of each other, it is possible to see any other transactions using the same public keys from the blockchain. If a user doesn't change their public keys often, everyone else can see the spending patterns from different public keys and any unspent amounts. To avoid this users can adopt a policy to use one public key only twice, once to receive and once to spend coins.

Malleability Signature scripts are susceptible to a denial of service attack known as transaction malleability. Since the scripts can't sign themselves it can be used by an attacker to modify the transaction and it will be still a valid transaction. These attacks can change computed hash of transactions and hence transactions which depend on previous transactions should be avoided. It also adversely affects the transaction tracking because the transaction with a particular transaction id will not be visible due to the modification. Also transaction tracking is mostly done by tracking the outputs as they can't be modified by this type of attack.

4.2.2.2 Multisig Escrow services

Escrow is a service where two parties entering into a contract, appoint an external party to hold the money or documents, and release them according to the rules of contract. A traditional example of an escrow service is a hospital and a medical supplier getting in a contract. The supplier agrees to supply the medical equipment to the hospital over the next year and every request for an equipment should be met within one week of request. At the end of year, if the number of requests which have not been met in a week exceeds 10, then the supplier gives back 10 percent of the amount of contract to the hospital. They both agree on a third party to monitor this agreement and hold 10 percent of the value of contract and in the end release it to the party which should be paid back depending upon the number of times a request has not been fulfilled within a week from request.

Bitcoin Contracts Transactions which use Bitcoin to act as financial instruments, which enforce agreements. If a user wants to buy something from a seller, the user can send a transaction which will only pay the seller if the user receives the item. But in case of a dispute this simple contract can't be resolved easily. This problem can be resolved by using an intermediary authority which decides whoever should receive the payment back from the above contract.

4.2.3 Ethereum

Ethereum [1] is an open source platform which enables the creation and distribution of decentralized applications. The driving force behind it is a group of computer scientists and programmers that envision a cryptographic platform with a built-in programming language attempting to generalize concepts such as multi-signature escrow, bets, contracts for difference, etc. This all-encompassing concept of "contracts" would allow users to come up with arbitrarily complex combinations of arithmetic formulas and nested if-then clauses to set up conditions for how funds could be spent. Vitalik Buterin, one of the founding members of the Ethereum team describes the goal of Ethereum in the White Paper as follows:

> ”What Ethereum intends to provide is a blockchain with a built-in fully fledged Turing-complete programming language that can be used to create "contract" that can be used to encode arbitrary state transition functions, allowing users to create any of the systems described above, as well as many others that we have not yet imagined, simply by writing up the logic in a few lines of code." [1]

4.2.3.1 Timeline

It was introduced by Buterin in early 2013. It is work in progress and has not been applied to a real world case. This project is still active and is expected to arrive in early 2015 according to the founders and the website. The first proof of concept was released in February 2014. The current roadmap [2] shows all the expected features and phases of development which have been planned.

4.2.3.2 Introduction to Ethereum

It is a distributed, decentralized platform for publishing digital contracts on top of a turing complete language. It uses ether to power itself and also to execute and maintain state of the contracts. A contract may execute any code depending upon the amount of ether its willing to spend. Nodes choose any contract to be executed and added to the blockchain similar to Bitcoin, but contracts can be much more complex in this case. Contracts can run any program until they run out of the ether. These contracts have applications, like running computations on distributed networks or creating financial derivatives based on external factors, which can be used to calculate and dispense the value to the owners of the contracts, based on exact calculations. According to the official whitepaper following are the major expected applications of Ethereum:

1. *Token Systems*: these are sub-currencies which can be implemented on top of exisiting crypto currencies. They can be easily implemented using Ethereum in the form of contracts.

2. *Financial Derivatives*: they are implemented using external price monitoring services. They can be used as hedge against other basic financial instruments with a transparent pricing of the instrument rather than a third party bank fixing it for the public.

3. *Identity and Reputations Systems*: DNS name resolution systems and Email authentication which involve a simple read only data based contract. Since the data cannot be modified in future it can serve the purpose of trust chains for web authentication and certificate authorities.

4. *Decentralized file storage*: the model of this application is that nodes can store data for a contract in return for getting ether. Now the user can encrypt the data and put different blocks in different contracts for the data to be stored with the ether being spent on each access.

4.2.3.3 Blockchain Extension

Ethereum expands on this idea by not only making simple states of key value pairs where the key is always a public address and the value a certain amount of bitcoin but extends this to any possible keys and values. This means that blockchain can be used to store data as well alongwith transactions. Blocks do not hash the data and only contain pointers to the data, which ensures efficiecy of the network.

4.3 Applications of Cryptographic Protocols

In this section the applications of cryptographic protocols are explained to solve the probelms related to the compensation, enforcement and monitoring of the SLAs using the technologies introduced in previous sections.

4.3.1 Application of Bitcoin to Compensation of SLAs

As discussed in Escrow section above, bitcoin comes as a very suitable solution for SLA compensation. It is a form of compensation which can be exchanged for money on a bitcoin exchange by both the parties, which makes sure that neither customer nor the service provider are bound to each other in future, in contrast to what we saw in the example with Amazon, usually the service providers return only credits to the customers accounts to use their services.
The major ways Bitcoin can be used for SLA compensation are with :

1. Multisig transactions: Transactions that have multiple signatories to make the transfers. Here, the third party acts as a deciding factor whether to make the transaction or not. For example, there are two parties Customer and Service provider. They create a 2 of 3 multisig address which requires atleast signatures of two parties. Customers sends reserve money to this address and if the SLA is fulfilled then both Customer and Service provider sign a transaction to initiate transfer to the Service provider. If the SLA is not fulfilled the trusted party verifies and signs transacation to initiate transfer to Customer.

2. Future transactions: Multisig transactions dated in future are a perfect way to make SLA compensations. It works exactly as Multisig transactions except the fact that they become active in future. In case SLA is not fulfilled any two parties can void the transaction.

4.3.2 Application of Smart contracts to Enforcement of SLAs

Smart contracts are supported both in Bitcoin and Ethereum. The basic architecture which supports enforcement of SLAs using smart contracts is as follows:

Parties Involved:

1. Service Provider

2. Customer

3. Third Party

In case of smart contracts, the third party is a set of oracles which monitor the SLA as defined during the agreement between the parties. Ethereum provides a set of nodes which can run simultaneous contracts and get a consensus based monitoring. SchellingCoin even goes another step further by making consensus based distributed monitoring architecture where nodes get paid to monitor and truthful reporting is based on the median reports.

4.3.3 Application of Ethereum for Monitoring of SLAs

Since Ethereum also supports a turing complete language, complex contracts can be created to monitor services like sending data chunks and calculating returned hashes which depend on latency etc. This enables SLAs to use Ethereum as a network to monitor even the most complex SLAs in the domain of computing. Supporting a turing complete language as the backbone for contracts ensures a standard way of monitoring and computing in the whole network. This is also in line with using Ethereum as a distributed cloud service provider with no single node as a service provider.

4.4 Discussion

Service level agreements are not evolving at the same pace as the services in the past few years. This gap has been increasing because the Services are becoming more and more complex due to which monitoring of the parameters had been a challenge. But most difficult part of monitoring had been the old way of being done by either the Customer or Service provider or a third party. This approach failed most of the times because it is prone to fraud by any of the parties involved. This report discusses the applicability of different cryptographic protocols to solve those problems as discussed above.
There are some issues which were raised during the presentation of this report during the seminar as well such as:

1. Automated transfers: some participants argued that they don't feel safe with automated transfers, using bitcoin contracts as a way of SLA compensation. It is a very valid point and there is further work that needs to be done regarding how to make such transfers more secure and fraud proof.

2. Monitoring using distributed oracles: distributed oracles still face the same issues as a third party based monitoring solution. Another important drawback also says that for parameters like reachability there are number of external factors involved which are independant of the service provider of the cloud service but might be dependent upon an end user's reachability.

4.5 Summary and Final Considerations

Recent Bitcoin advances made it possible to have a solution where a network of nodes can reach consensus on defined states. The focus here lies on consensus of computationally provable facts. With the rising interest in the space of cryptographic protocols there are however also advancements in extending this to facts outside of the network.
This report showed how Bitcoin as a payment mechanism already could be of use to the area of SLAs as a means of compensation in the case of a SLA violation.
When discussing how smart contracts could be applied to SLAs we discussed how a smart contract always had to be based on a data point outside of the consensus network.
Some of the concepts we discussed are highly hypothetical and their viability in the real world still needs to be proven. However, if decentralized oracles become widely used, it will certainly open opportunities for many new applications in a wide range of domains that would profit from a trustless automated contract resolution process.

Bibliography

[1] Vitalik Buterin: *Ethereum White Paper*, November 2014. https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf

[2] Ethereum Dev Plan preview website, November 2014. https://www.ethereum.org/pdfs/Ethereum-Dev-Plan-preview.pdf

[3] Bitcoin.org: *Developer reference block-chain* https://bitcoin.org/en/developer-reference#block-chain

[4] reddit.com: *SchellingCoin for consensus based monitoring* http://www.reddit.com/r/ethereum/comments/25vmno/schellingcoin_for_oneoff_discrete_outcomes/

[5] *Byzanitine Generals Problem Website* http://www.mail-archive.com/cryptography@metzdowd.com/msg09997.html

[6] Andrew D.H. Farrell, Marek J. Sergot, Mathias Salle, Claudio Bartolini, David Trastour, Athena Christodoulou *Performance Monitoring of Service Level Agreements for Utility Computing Using the Event Calculus* http://www.hpl.hp.com/techreports/2004/HPL-2004-20R1.pdf

[7] *Amazon Elastic Compute cloud website*, November 2014 https://aws.amazon.com/ec2/

[8] *Amazon EC2 SLA Website*, November 2014 https://aws.amazon.com/ec2/sla/

[9] *American Eagle 8 day outage website*, December 2014 http://www.availabilitydigest.com/public_articles/0509/american_eagle.pdf

[10] *Amazon Netflix Use Case website*, November 2014 http://aws.amazon.com/solutions/case-studies/netflix/

Chapter 5

QoE Based Charging

Sandro Boccuzzo

Over the past decades when it came to define and charge for computer relevant tasks such as packet loss rate, delay, bandwidth etc. they where mostly described in terms of Quality-of-Service (QoS). Resent academic research and industrial solutions propose a newer rather user centric quality concept. They focus on the user experience and define a Quality-of-Experience (QoE) as an important measurement to describe the 'overall acceptability of an application or service, as perceived subjectively by the end-user'[1] With QoE research tries to quantify this perception of a content by the end-user and evaluate its value. This can be used to allocate resources more efficient and charge more accurately to the endusers willingness to pay for a particular experience. In this work we focus on current work done in the field of QoE Based Charging and discuss particularly approaches in combination with network, multimedia and mobile services [2].

Contents

5.1 Introduction

User experience is a major concern of IT-Services and user interaction. Without efficient communication services are not perceived as wished from the end user. To control the quality of a service traditionally measurements of computer relevant tasks such as bandwidth, delay or packet loss rate are defined. With service level agreements (SLA) customers and clients than specify what penalties apply if the defined targets are not achieved. IT-services that are charged based on such measurable parameters are mostly described in terms of Quality-of-Service (QoS). The end users perceived service experience in regard to its expectations and its willingness to pay are not taken into consideration. These levels of different perceived experiences in that are crucial. As a dropped film sequence during an action movie can have a negative impact to the overall experience, a movie added with higher definition or surround sound can enlarge the experience.
Still the price a user expecting high definition and surround sound is willing to pay is different than the price a user is willing to pay for high definition if he for example is already happy with a lower resolution. But how can a user be charged accurately based on his personal preferences. In this paper we address current research done in that field and open discussion for possible solutions.
The remainder of this paper is organized as follows. In Section 5.2 we address the general differences of quality of services and quality of experience. The section 5.3 provides an overview of research done in the field of quality of experience. In Section 5.4 propose a possible real live scenario for QoE based charging. We summarize with our conclusions in Section 5.5.

5.2 QoS versus QoE

What is the difference between charging for a service quality in contrast to charging for an experience? And why most current charging scheme are rather based on quality of service? If we formally look at how charging is done in a quality of service approach, we se that the functions all depend from each other (Figure 5.1). For any given price function p(x), we can get the corresponding demand function d(p) and from that the quality function q(d) results.

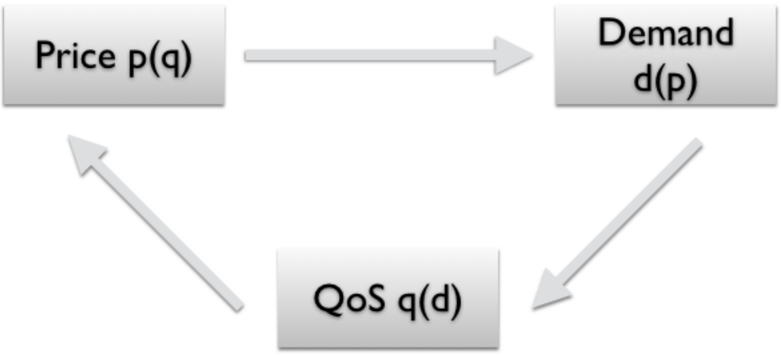


Figure 5.1: Functions involved in QoS

If we consider the variables involved in quality of experience however it is not so trivial. The function of the quality of experience is dependent from the price function p(x) and of the quality function q(d). The price function on the other side is dependent from the experience function x(p,q). This reciprocal dependency means that for one particular quality of a service two enduser might have a different price. Formally the situation is disclosable but what makes it so complicated to measure the quality of experience and charge accurately? In a real situation the experience function is characterized by the personal preference of the end user, it is therefore not a function that is clearly exposed. Furthermore the experience function can extremely differ among the end users and varying even from their daily sensations. That means that an end user might what to pay more to see a movie during a relaxing evening, than see the same movie during lunch.

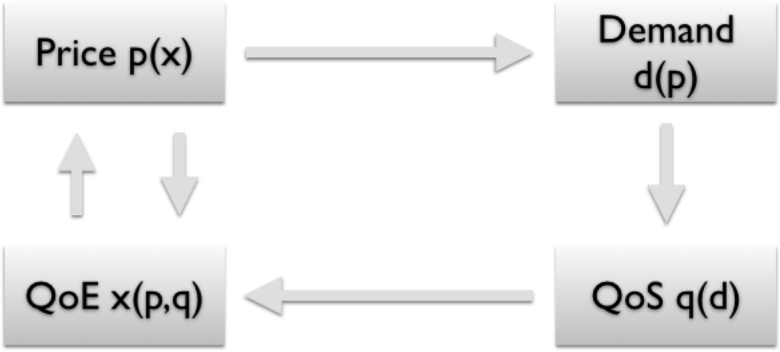


Figure 5.2: Functions involved in QoE

5.3 Related Work

The goal in charging for quality of experience is to explore ways to measure the personal preferences of a end user in any given time and to adjust the quality accordingly. In the following we summaries how other research addressed this goal.

5.3.1 QoE in the home cinema environment

In their work on QoE-based Charging Reichl *et al.* addressed the issue of measurement with a user study [3]. Their setup provided a TV screen to the test user on which a film in various qualities was shown (Figure 5.3). The quality was adjusted based on a logarithmically scaled bitrate and on two general film sources. The logarithmically scaled bitrate together with the standard definition (SD) and the high definition (HD) files allowed a distinction of 17 quality levels. On top of that they added three other virtual quality levels that would show the same best quality but to a different price. As an incentive the test users could no spend 10 euros in quality enhancements and take the remaining home afterwards. Depending on the assigned group the film price range was between zero and 2, 3 or 4 euros. The test user watched three film sequences of 20 minutes each. With a jog wheel the user can freely adjust the quality during the first 5 minutes after that the last selected quality is set and charged with the previously shown price.
The work of Reichl *et al.* in our opinion shows an interesting approach towards quality of experience measurement. However, we find as well some critical notes towards the work results. In their work the used 17 quality levels plus three virtual levels that in the end classification where summarized into four groups. In these four groups the three virtual quality levels where combined together. With a focus on charging for quality of experience, in our opinion clustering the virtual groups together results in changing the whole test scenario to measure only charging for different quality of services rather than charging for the quality of experience. Let us explain our point. As we discussed before charging for quality of experience is defined in the end users willingness to pay for a perceived experience. This assumes, that one user is willing to pay more for the same quality than an other user. In the previous scenario, the users had time to test

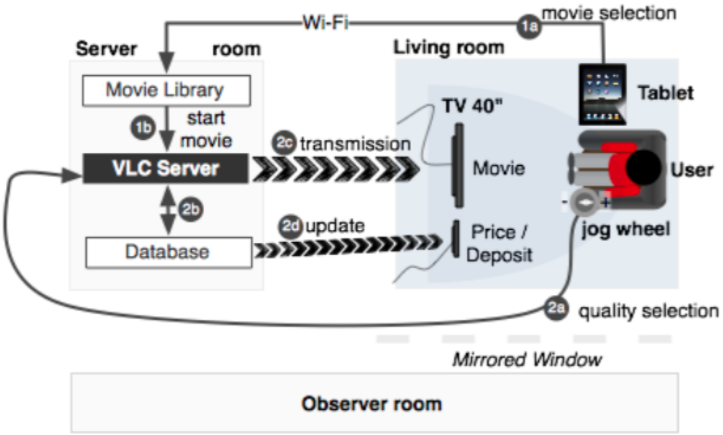


Figure 5.3: Technical Setup of the Reichl *et al.* User Study

the available qualities and than pick one desired one. This is rather similar to a simple i search for my preferred mobile phone subscription. Different service levels are offered to me with a transparent price and I can choose the one suiting my best and close the deal. This is clearly rather choosing a quality of service than an experience. Secondly a users experience can changes in any moment, in the mentioned scenario however after the first 5 free testing minutes the deal is closed and the rest of the time the user consumes the chosen quality. If *e.g.,* during the movie a phone call would come in and disturb the user he might not need the same quality anymore. The situation has changed and with it the users experience for which he is willing to pay. We therefore argue, that in regard to charging for quality of experience Reichl *et al.* in the mentioned work should have focused more on the virtual quality levels and eventually added some virtual quality levels as well in the lower and middle price ranges. With that they could distinguish the individual users and their perceived preferences and also see if the price volatility would be stronger in the lower, middle or higher price range.

5.3.2 QoE for multimedia content evaluation

Chen *et al.* with their work addressed how multimedia content can be evaluated with users quality of experience[4]. To evaluate the multimedia content they build a crowdsourceable framework. In their setup they used an original audio or video clip and created a set of differently encoded versions. During the case study a participant would than being presented an audio or visual mulitmedia content. Two qualities of the same clip where randomly selected and one of them played. The participant can switch between the two selected clips by pressing the space key. If the user is ready to take a decision towards one clip he prefers he would than we pressed the left key to indicate the clip with the released space key is better and press the right key it the clip with the pressed space key is better (Firgure 5.4). With their framework Chen *et al.* where able to outsource QoE evalutaion experiments to an internet crowd, reaching a wider participant diversity while preserving the quality of the results.

5.3.3 QoE in the mobile network environment

Qiao in his work addresses the collection of QoE data across the mobile network [5]. In his work he proposes to process QoE measurements such as the speech conversation quality related method ITU-T P.563 [6] directly on the smart phone and submit only the final QoE results back to the network servers. This would help to collect QoE relevant data directly in the smart phone and reduces the transmission load of the network. The advantage of

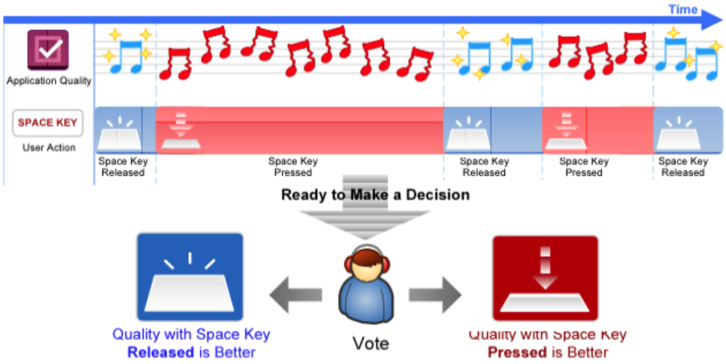


Figure 5.4: Experiment setup of a participant in an acoustic paired comparison.

this proposal is that an offline QoE measurement is possible, which is specially useful on remote locations where network lost becomes an issue to take into account for QoE. Stankiewicz and Jajszczyk in their survey [7] describe QoE as strongly dependent on intrinsic network features and performance. For them apart of the quality of service (QoS), the quality of resilience addressing the recovery time and availability (QoR) as well as the grade of service (GoS) are such intrinsic features (Figure 5.5). Still there are no simple mapping between these and QoE and that efforts in finding mathematical relationships between QoS parameters and QoE, like the mean opinion score (MOS) are always derived for a particular use case and under several assumptions. Furthermore in their work the address the various network technologies towards their typical range, downlink data rates and handover supports and focus on the convergence between fixed and wireless networks, as well as within wireless networks based on different technologies. The state that with growing user expectations a harmonization between different standards and mappings can help fulfilling these user expecations.

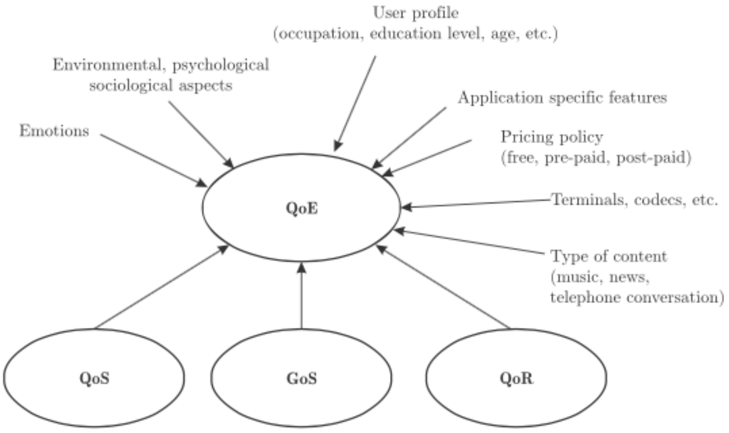


Figure 5.5: Factors influencing QoE.

5.3.4 QoE in the gaming environment

Jarschel *et al.* addressed QoE from the perspective of cloud gaming [8]. In cloud gaming the player is no longer depending on a specific gaming hardware. Cloud computing is handling that. A cloud game on the other hand becomes heavily dependent on the underlining network in terms of latency and bandwidth. Jarschel *et al.* designed a survey where a WAN-Simulator would allow to configure specific test scenarios and the test users would answer to a QoE related poll (Figure 5.6). The handle it the way that the test user would sit in front of two monitors and answer the poll questions on the first monitor while gaming on the second one. From their study the were able to identify key influence factors impacting QoE of the test users. According to their test scenario a delay of 80ms would represent a threshold where player start to notice the delay. However, they also found out that the perception of the delay varies according to the speed of the game in general. A delay in *e.g.*, role play game would be perceived different than the same delay in a soccer game. They conclude that the slower the game is the less a delay influences the users QoE. The influence of such key quality indicators is particularly useful for a service providers to ensure a minimal level of QoE at all time. Last but not least Jarschel *et al.* showed also that a package lost on the downstream side has a significantly higher impact on QoE than a packet lost on the upstream.

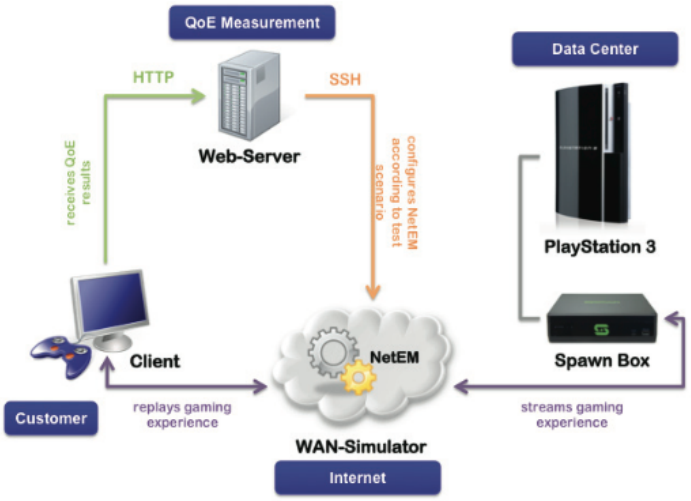


Figure 5.6: Testbed Setup - Logical View of the cloud gaming study

5.3.5 QoE in commercial products

As an example of a commercial service that incorporates in a way a quality of experience based charging approach we mention flattr[13]. The idea behind flattr is to support creators of web content financially based on the own experience. If a had a particularly good experience in consuming the content he can klick on the flattr button on the content providers website (Figure 5.7). Flattr subsequenceially gets informed of the users action and records the click in the users profile. The user at the beginning of the month has specified what amount (flatrate) he is willing to spend on preferred content. At the end of the month this amount is then divided among the content providers that received a flattr click from the user during this month.

Figure 5.7: left: flattr button on content websites right: browser extension to flattr website

Even-though this approach uses quality of experience principals to divide the amount the user is willing to pay for content he had a particularly good experience, there are also some downsides in this approach. One is for sure that it does not take into consideration how much a user preferred one content in respect of the other. The other is that a content provider in one month might be the only one that particularly pleased a user and subsequenceially gets the full share, while on the other month needs to divide the share with hundreds of other content providers. With flattr the cost to produce the content are not honored in any way.

5.4 Towards a real QoE based charging

As we have seen in related work, it is not so simple to create a real quality of experience scenario, without the risk of falling back into a simple charge for a quality of service approach. Dring our survey on related work we found that most of the QoE focused research aims rather into finding some sort of a prediction model for QoE based on some key influence factors of the underlining QoS. But how can we build a real live scenario allowing QoE based charging? In the following we propose a possible real live scenario for QoE based charging.

5.4.1 A real live QoE scenario

For the purpose of our discussions we need some simple real live QoE scenario. Lets think of a hotspot event such as new years eve on New Yorks Time Square. Millions of individuals that at midnight all like to use the same mobile service for sending a short message with their best new year wishes to their relatives. In this scenario the underlying service is simple and clear. Deliver the short message as instantly as possible to the relatives. Because of the resource restriction some of the individuals experience lack of the sort message service, while other can send the message but the message would reach the relatives only hours later. Depending on the importance of the message in this use case there is a clear difference in the willingness to pay for the short message service by any individual. Some might be able to wait, while others absolutely like to have their message delivered before one minute past midnight. In this scenario the quality of experience of having the message delivered instantaneously is clearly worth more to some individuals. But how can we use our limited service resources to provide an adequate service and charge the individuals differently according to their experience preferences. We think that such a use case is best addressed in the same way as a stocks on a market.

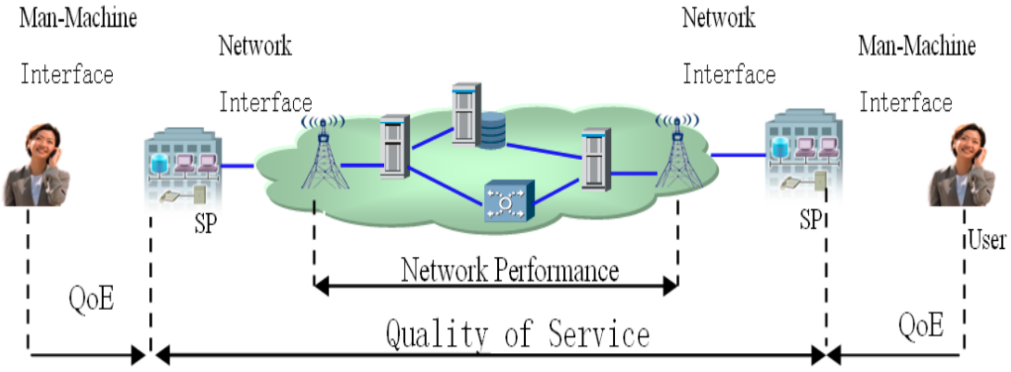
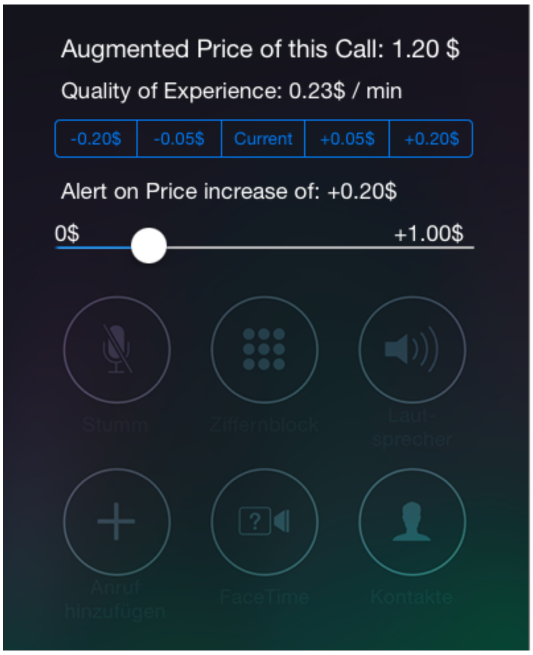


Figure 5.8: Relationship between end to end QoE and network level QoS[5]

5.4.2 A QoE oriented marketplace

We base our stock market inspired QoE approach on a network where every ip-package has it price or were alternatively in a specified time interval a package would deliver the new summarized current price for a requested or still used service. The research community has already addressed this issue in various cases such as Elovici *et al.* in [9] or Chen *et al.* in [10]. An overview of pricing concepts for IP networks is provided as well by Falkner *et al.* [11] or by Da Silva [12].The important part of the network part is a transparent way to adjust the price based on the route a packet has to take and eventually some fixe cost a service is charged for (Figure 5.8).

In a user interface for the end user requesting a service such as sending a short message a box with the current price can be presented and accepted or declined by the user. With a simple service such as a short message where the message size is small and know up on the request of a service that does not opens many issues. However, if in our scenario we think of a phone call instead of a short message a lot more issues arise. What happens if a call is started with a low service price and the price rises? Or what if a user with a mobile device moves to a hot spot or out of a network area feasible for the requested service? For these situations the user interface needs an alert functionality to warns the user before reaching an edge of a network area, to specify a maximal price increase, a minimal expected QoE or to back propagate his current QoE. This issues can be resolved in offering a user interface that after a phone call has started presents a user a set of sliders to adjust his current QoE and some other boundaries based on the current available limitations (Figure 5.9). With a user interface similar to the shown the quality of experience can be adjusted at any time based on the users current preferences and wiliness to pay. Furthermore the current QoE is stored and used as the expected QoE of the user when initiating a new phone call.


Augmented Price of this Call: 1.20 $

Quality of Experience: 0.23$ / min

Alert on Price increase of: +0.20$

Figure 5.9: Suggested UI for QoE based charged phone call

5.5 Conclusions

In this paper we presented various approaches available in respect to QoE based charging. We showed what issues arise in trying to measure the individuals quality of experience, showed we researchers might have struggled in confounding QoE with traditional QoS approaches, and summarized some studies that intended to statistically approximate QoE with the help of adequate key influence factors. We rounded this work up by presenting our own approach towards a real world scenario implementing QoE based charging.

Bibliography

[1] ITU-T Rec. P.10:: *Vocabulary for performance and quality of service, Amendment 2: New definitions for inclusion in Recommendation ITU-T P.10/G.100.*, International Telecommunications Union. Geneva, Switzerland, 2008

[2] Mohseni, S.: *Driving Quality of Experience in mobile content value chain*, In 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST), pages 320-325, April 2010.

[3] Reichl, Peter and Maille, Patrick and Zwickl, Patrick and Sackl, Andreas: *A fixed-point model for QoE-based charging*, In FhMN ?13 Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking, pages 33-38. ACM, 2013.

[4] Chen, Kuan-Ta and Wu, Chen-Chi and Chang, Yu-Chun and Lei, Chin-Laung: *A Crowdsourceable QoE Evaluation Framework for Multimedia Content*, In Proceedings of the 17th ACM International Conference on Multimedia, MM ?09, pages 491-500, ACM, 2009.

[5] Zizhi Qiao: *Smarter Phone based Live QoE Measurement*, In 15th International Conference on Intelligence in Next Generation Networks (ICIN), pages 64-68, Oct 2011.

[6] ITU-T Rec. P.563:: *Single Ended Method for Objective Speech Quality Assessment in Narrow-Band Telephony Applications ITU-T Recommendation P.563*, International Telecommunications Union. Geneva, Switzerland, 2004.

[7] Stankiewicz, Rafal and Jajszczyk, Andrzej: *A survey of QoE assurance in converged networks*, Computer Networks, 55(7):1459-1473, 2011.

[8] Jarschel, Michael and Schlosser, Daniel and Scheuring, Sven and Hoßfeld, Tobias: *An evaluation of QoE in cloud gaming based on subjective tests*, In Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pages 330-335. IEEE, 2011.

[9] Elovici, Y and Ben-Shimol, Y and Shabtai, A: *Per-packet pricing scheme for IP networks*, In 10th International Conference on Telecommunications, 2003. ICT 2003., volume 2, pages 1494-1500. IEEE, 2003.

[10] Chen, Yih-Farn Robin and Jana, Rittwik: *SpeedGate: A smart data pricing testbed based on speed tiers*, In INFOCOM, pages 3195-3200. IEEE, 2013.

[11] Falkner, Matthias and Devetsikiotis, Michael and Lambadaris, Ioannis: *An overview of pricing concepts for broadband IP networks*, Communications Surveys & Tutorials, IEEE, 3(2):2-13, 2000.

[12] DaSilva, Luiz A: *Pricing for QoS-enabled networks: A survey*, A survey. Communications Surveys & Tutorials, IEEE, 3(2):2-8, 2000.

[13] Flattr; `www.flattr.com`, december, 2014.

Chapter 6

QoS in Mobile Ad-hoc Networks (MANET)

Mark Bosshard

Mobile Ad-hoc Networks (MANET) are networks consisting of only mobile participants without any fixed infrastructure. Applications of MANET can be found in many areas, such as in military communication systems, sensor networks or on the road between cars (Vehicle ad-hoc Networks or VANET). All of these applications depend on a high quality of service (QoS) in different means. Today a lot of Quality of Service optimization has been done for our Internet - this is still missing though for MANETs and not solvable with the same ease either. This work shows problems in this field and what solutions or approaches in terms of routing protocols are available today.

Contents

6.1 Introduction

The two adjectives differentiating a MANET from every other network are mobile and Ad-hoc. Ad-hoc refers to the temporariness of a network structure. Mobile on the other hand stands for moving participants (nodes), and even per definition wireless nodes [5]. A wired backbone infrastructure (*i.e.* central hubs or routers) does not exist, since MANETs communicate from network participant to network participant only, without any installation on premise. Network topology between these nodes is subject to constant change since all mobile nodes are allowed to move freely. The communication between nodes is happening through multi-hops [5]. This means a sender's signal strength might not directly reach the destination node, but other nodes in between can forward pakets and thereby establish a path. A MANET has to be rapidly deployable and work immediately [17]. Routing protocols have to deal with these challenges. Since MANET is a class of Wireless Network, the physical link's bandwidth and delay are unpredictable as well [16]. First applications of MANET will be shown in Section 6.1.1, followed by a general introduction to QoS (Section 6.1.2). In Section 6.2 this paper outlines QoS goals and metrics specific to MANET. Finally present design solutions in terms of routing protocols are looked at in Section 6.3. Section 6.4 summarizes all previous sections and in the end a discussion including future prospects and economic aspects is being held.

6.1.1 Applications and Commercial Use of MANETs

Nowadays MANETs are applied in several specific fields. One example is temporary networks created for military purposes, K. Wu et al. mention "battlefield communications, disaster recovery etc." [20]. In a battlefield one cannot rely on a pre-installed backbone infrastructure. The areas are not civilized enough to build such a set-up or often do not even belong to the party in need of a communication system. In disaster recovery situations static communication infrastructure is broken or not available. Circumstances therefore are very similar to these in a battlefield.

Vehicle Ad-hoc Networks (VANETs), are another type of MANETs. The participating nodes form networks between vehicles (often automobiles) [12]. They mainly differ from MANETs in a way that power supply is not an issue. In addition, hardware size and costs are looser constraints. Challenges for this network type are frequent topology changes (*i.e.* relative speeds up to 500km/h) and either dense or sparse network configurations (*i.e.* a few up to several hundred meters of distance). Their key application nowadays is road safety (such as avoiding accidents), however missing economic incentives for a roll-out and missing international communication standards are reasons for VANETs not to be widely used by today [15].

Another example that differs from traditional Mobile Ad-hoc Networks is mobile networks in the sky between Unmanned Aerial Vehicles (UAVs). These vehicles would fly close to the ground, serving e.g. in "disaster assessment and recovery, emergency communications, infrastructure protection and surveillance", as stated by K. Namuduri et al. [12]. There are also UAVs that enabled cellular communication in Japanese earthquacke-regions in 2013 or agricultural UAVs spraying chemicals over fields from the air. Here mainly bandwidth could be improved through creating a Network in the air instead of connecting each node directly to the endpoint on the earth or satellite [12].

Tonnesen also mentioned Sensor Networks as a commerical use [17]. "An ad hoc sensor network is a collection of sensor nodes forming a temporary network without the aid of any central administration or support services. In other words, there is no stationary infrastructure such as base-stations." as defined by M. Tubaishat et al. [18]. Even though their definition matches the MANET quite well, sensor networks show several unique differences to MANETs. Firstly "the number of ... nodes in a sensor network can be several orders of magnitude higher than the nodes in an ad hoc network" [1]. Secondly the nodes are deployed more densely and their topology may change much more rapidly. The oftentimes tiny nodes are error-prone, and limited in power, computational capacities and memory. The large amount of overhead does not allow to use a Unique ID for each node. The protocols most used today are based on a broadcast approach rather than point-to-point communication [1]. These are just a few applications, various more are expected to be coming up in future.

6.1.2 Quality of Service

Quality of Service (QoS) is a guarantee by the network to satisfy service performance within predefined upper and/or lower bounds. End-to-end delay statistics, available bandwidth, probability of packet loss are relevant factors [4]. Commonly on the end-user's side QoS is most known as a set of guarantee measures for the transportation of audio/video over the network, such as IP TV. Meeting QoS Standards is an end-to-end issue, all involved elements have to work in unison in order to achive a desired application level behaviour [2]. It is a challenge to not only offer QoS for individual architectural components and make these configurable, but also design an overall QoS Architecture for multimedia communications over the whole system [2]. However in MANET the topic of QoS is not only an implementation but rather still a question of unsolved network design. Further discussion is provided in the Section 6.2.

6.2 Challenges for QoS in MANETs

Currently, many implementations for QoS are supporting the Internet's common applications (e.g. packet prioritization in routers of IP TV providers). In MANET these topics of Internet QoS are basically present as well. Additionally, there are "resource constraints (e.g. computing power, energy, bandwidth time) ... and dynamics (e.g. topolgy changes, node mobility, node failure, propagation channel conditions)" [5] present as additional QoS challenges, that make guaranteed QoS impracticable in MANETs with their mobile nodes. Only "soft QoS" is feasible, and even that only if topology change does not appear faster than the time window needed for updating parameters to propagate to the entire network [16]. The restrictions of MANETs in detail are described as follows:
Energy - Mobile nodes often are powered by a battery, with the exception of VANETs, where a fuel generator is available. For example in UAVs, a lightweight electricity supply is often crucial. This limits communication distance of the physical wireless links strongly.
Computing Power and Memory - In many applications of MANETs, nodes are tiny or lightweight devices. A good example is sensor networks, where this is mostly the case. Usually neither much computing power nor memory available and MANETs have to deal with these restrictions.
Node Failure - Nodes can disappear suddenly without announcement in advance due to position changes, technical defects and also an empty battery. A fully functioning MANET in that sense should be able to work without a single node at any time.
Rapid Topology Changes - The rapid change in structure of a MANET brings hard challenges with it. Distances and even the reachability of nodes are able to change constantly. If these changes happen too frequently, finding a loop-free path between source and destination may become impossible. This is the case, when nodes are moving again, before the previous topology update or routing information has been propagated to all pertinent nodes. A network is called "combinatorially stable", if changes happen slow enough for topological update information to be propagated in time to all nodes [4].

In the following sections different routing protocols and their trade-offs are discussed. Thereby we will evaluate the QoS measures scalability, availability, reconfigurability, reliability and security [5], which are mainly showing what a MANET set-up should follow.

6.2.1 Scalability

Scalability in a MANET should allow for significant growth in the number of nodes. With wireless links and restricted bandwidth, interference can be a limitation. Regulating the transmission power can mitigate this problem [7]. R. Rathee and R. Pahwa show that to a certain extent, adding more nodes generates smaller distances and improves communication. On the other hand adding too many nodes augments their collision rate, which in turn worsens poor performance in communication [13].

6.2.2 Availability

In mobile networks, nodes move freely, what oftentimes leads to link failures. Furthermore, node failures are expected to appear (nodes that stop working, *e.g.* in networks with many participating nodes a single node can be very primitively manufactured and of low quality). This leads to frequent connection losses and separate network partitions. Hence, mobile nodes in one partition are not able to access data in the other partition any more [22]. Data availability can be ensured through replication which stands in a trade-off to query time, node availability is hard to be fully ensured. Availability can be measured in percentage points stating how much time all participants of a network are reachable [10].

6.2.3 Reconfigurability

Reconfigurable hardware components ”can be reprogrammed after fabrication to achieve flexibility and customizability.” [21]. For MANETs this also means that topology restructurings are firstly propagated and secondly actively applied to all nodes.

6.2.4 Reliability

A reliable network ensures no data loss, no duplication and no out of order delivery of packets. Reliability is a vital necessity for all application programs in all communication networks [19]. In other words, reliability means a successful delivery of all data packets sent. The term ”fault-tolerant” states how reliable a network in case of failure of one component still is [10].

6.2.5 Security

Transmission of MANETs must be secure to prevent eavesdropping [7]. Apart from simply eavesdropping there are various different possible attacks, an unfinite list by Jin-Hee Cho follows here [5]:

- *Routing Loop Attacks*: A malicious node generates an eternal loop for packets. These packets will then not be able to exit that loop anymore and never reach destination.
- *Wormhole Attacks*: A group of malicious nodes can pretend to connect to very distant points with low latency and thereby disrupt normal traffic flow.
- *Blackhole Attacks*: A node always responds positively to route requests and then drops all packets. Similarly, Grayhole Attack and Sinkhole Attack nodes drop packets selectively.

- *DoS Attacks*: Nodes can cause excessive resource consumption and thereby block the normal use or management of communication facilities.
- *False information or recommendation*: Through providing false information a malicious node can exclude a good node.
- *Incomplete information*: A malicious node may provide improper or incomplete information. In MANETs this phenomenon could also appear due to node mobility or link failure.
- *Packet modification/insertion*: The modification of packets or insertion of malicious packets *e.g.* incorrect routing information can disrupt a network.
- *Newcomer Attacks*: If a node has a bad reputation, it can discard this reputation by registering as a new user.
- *Sybil attacks*: Topology maintenance or fault tolerant schemes such as multi-path routing can be disrupted if a malicious node uses multiple identities.
- *Blackmailing*: Significant amount of traffic and disruption of the entire network can be reached through a node disseminating false information *i.e.* which states that another node is malicious.
- *Replay Attacks*: A malicious node may replay earlier information, which is not a big trouble with data but can disrupt a network when routing requests are replayed and routing table informations become erroneous.
- *Selective misbehaving attacks*: A malicious node only behaves badly to several nodes.
- *On-Off Attacks*: In order to stay undetected a malicious node may alternatively behave well or bad.
- *Conflicting behaviour Attack*: A malicious node may behave different to two groups of nodes to generate a mutual bad reputation and ultimately non-trusted relationships.

6.2.6 Quality of Service Metrics

In MANET networking, the different possible routes for packets are compared with the help of their numerical values associated, which are called ”metrics” [16]. These metrics also specify the QoS of a network as a whole. In general, we have three different metric types, where x(ni, nj) is a metric for link (ni, nj) and p(n1, n2, ..., nm) denotes a path from n1 over n2 and more nodes until node nm [16]:

- **Additive Metrics**
  x(p) = x(n1, n2) + x(n2, n3) + ... + x(nm-1, nm)
- **Multiplicative Metrics**
  x(p) = x(n1, n2) * x(n2, n3) * ... * x(nm-1, nm)
- **Concave Metrics**
  x(p) = min((n1, n2), x(n2, n3), ..., x(nm-1, nm))

In order to get a reasonable QoS over the whole network, choosing the right path is crucial. The commonly used metrics to achieve QoS are bandwidth and delay, where bandwidth is an example for a concave metric and delay for an additive metric. Additionally, delay jitter, energy or number of hops are other additive metrics that should be considered. If there are two or more additive metrics involved, finding an optimal path can be a NP-complete problem (i.e. solvable in nondeterministic polynomial time or put in simple words: almost unsolvably complex for a computer) to solve [20]. Hence, many of the present routing algorithms proposed in the literature are looking for paths that satisfy multiple constraints instead of complete optimal routes [16]. An example for a multiplicative metric would be reliability or packet loss [9].

6.3 Existing routing protocols in MANET

There is no central router in a MANET, this makes routing not as simple to implement as in the structure of our commonly used Internet. It is the multi-hop operation, that requires a routing mechanism designed for mobile nodes and with loop-free paths. Dynamic topology changes and rapid convergence, but also assuring a minimal network traffic overhead and scalability for extension are topics that have to be dealt with at one the same time [17].
Even though there is still research going on looking for new routing protocols, so far there are the following three types of protocols being distinguished:

- **Re-Active Protocols**
  Re-Active Protocols do not take initiative for finding routes actively. When a target aim is looked for, a broadcast request is flooded through the whole network and a unicast response from the target aim confirms the path [17]. An Advantage of re-active routing protocols is no presence of overhead traffic when not communicating. The downside at the same time being vast traffic when searching a node (i.e. performing a broadcast through the whole network). Due to those facts, a delay occurs in the communication, until the communication path is established [16].

- **Pro-Active Protocols**
  Pro-Active Protocols maintain all routes permanently and set them up initially. In order to achieve that, control traffic packets are sent over all established paths. As a downside proactive protocols are constantly generating overhead traffic for maintaining all routes, the advantage is no need of a broadcast and especially a lot faster establishment of connections, which means a shorter delay [17].

- **Hybrid Protocols**
  Hybrid protocols combine properties of pro-active and re-active protocols [16].

The above one is the most popular distinction applied to distinguish MANET protocols. However, Sundar et al. also propose a distinction of multiple versus single metrics, where one category is establishing routes based on a single metric and the other one multiple metrics [16]. As discussed above, comprising multiple metrics can be especially hard (up to NP-complete problems), when these are additive. In these cases calculation is achieved through specific heuristics [9].
A third possible distinction can be made on an either hierarchical or flat network structure. An example for a hierarchical structure would be core extraction, where a dominating set of nodes represents core nodes and all the other nodes choose one of their core node neighbours as its dominator [20, p. 17].
There are many different ways to let packets flow in a MANET with freely moving participants. These ways are called routing protocols. The following ones are the most important routing protocols, that also guarantee QoS. Based on the above structure, QoS-affine routing protocols and their way to ensure QoS will briefly be discussed in the following sections.

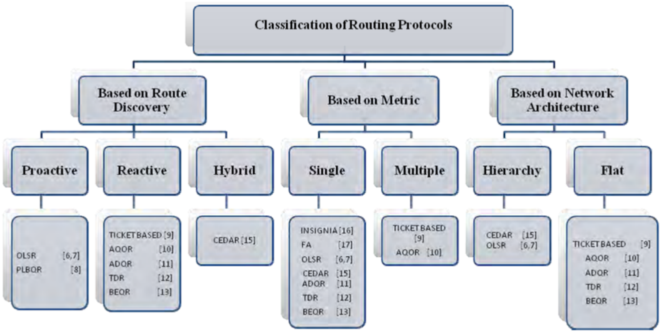


Figure 6.1: A possible structuring of QoS aware Routing Protocols as in [16]

6.3.1 Optimized Link State Routing

The Link State Routing Protocol (LSR) saves all the links present in a mobile ad-hoc network with the neighbour nodes and broadcasts (floods) this information in the entire network [8]. Hence, it is a pro-active routing protocol [16]. Its optimization OLSR reduces the size of control packets. This is achieved by using just a necessary subset of all possible links (multipoint relay selectors). Further traffic minimization and thereby improved QoS is reached by using just the selected nodes (multipoint relays) for retransmitting broadcast messages [8].

6.3.2 Predictive Location-Based Routing

The Predictive Location-Based QoS Routing algorithm tries to make routing more efficient by predicting the position of a node in future [16, p. 2079]. This is achieved by measuring the delay of two particular factors, extracting a trend how the node is moving. In a next step this trend is being applied to the node's current position to get the future position [14]. This mechanism requires nodes to be able to obtain their current position via GPS or another positioning mechanism [14]. No resources are reserved along the path from source to destination [16], but the knowledge of every node n of the position and path delay of its destination node m makes calculation of an optimal path and thereby network-wide QoS possible [14].

6.3.3 Ticket Based QoS Routing

The basic idea here is to use tickets, in order to reduce the number of candidate paths. Probe messages containing several tickets are issued by a source node when establishing connection to a receiver. Each intermediate node then splits the tickets if two or more paths are available, where the link with more residual bandwith gets more tickets [20]. So-called yellow tickets are able to determine both optimal delay and bandwith paths. Green tickets determine low-cost routes. A drawback of this algorithm is that the routing requires every node to keep track of all resource availabilities of its neighbour nodes, which requires enough memory [16].

6.3.4 Ad-hoc QoS Routing

Ad-hoc QoS On demand Routing is one of the QoS Routing Protocols listed here, that works with multiple QoS metrics at the same time. These are namely both bandwith and delay. These metrics are set as constraints when sending out a request. At route discovery time only nodes that satisfy both metrics will forward the packet and create an entry with expiration time. If there is no reply received within that time, the entry will be deleted. Path break requires a new route discovery initiated by the source [16].

6.3.5 Adaptive QoS Routing

Adaptive QoS routing (ADQR) is designed for the establishment of routes with a longer life time. There is a set minimal value for signal strength of the wireless links predefined. After a broadcast is sent out, the receiver chooses the best path. If signal strength falls below a threshold, a new route is being looked for [16]. That way QoS is assured by choosing the path with best signal strength.

6.3.6 Trigger-based Distributed Routing

Trigger-based Distributed QoS Routing tries to minimize the local memory needed in every node, by only saving their direct neighbours [6]. The protocol is location-based and nodes know about their neighbour's location as well as their power level. A route maintenance similar to the previously discussed ADQR protocol based on power level is sustained [16].

6.3.7 Core Extraction Distributed Ad-hoc Routing

Core Extraction Distributed Ad-hoc Routing Protocol (CEDAR) defines a dominating set (DS) of hosts (i.e. a ”core”), so that every host in the network is either in DS or a neighbour of a node in DS. In this hierarchical protocol every non-core host choses its nearest core host as dominator, core hosts are their own dominators [20]. Only core hosts are responsible for route computation, route maintenance and also for QoS provisioning [16]. CEDAR proposes a broadcast mechanism, in which nearby core hosts do not broadcast to each other (nearby hosts have a distance no more than three). This broadcast has thereby very low overhead and is further very stable for topology changes [20].

6.3.8 INSIGINIA

INSIGNIA stores signaling control information in the IP option of every IP data packet, the INSIGNIA option. This information is a minimum QoS guarantee (e.g. minimum bandwith), with sufficient resources available however it can be extended to support more QoS metrics [20].

6.3.9 Forward Algorithm

Forward Algorithm also takes bandwidth as its QoS parameter. Bandwidth is measured by calculating local maxima for adjacent links when discovering routes, and forwarding these values. It is a QoS extension to yet existing algorithms like AODV or TORA [16].

6.4 Summary

In the introduction it has been showed, that MANET stands for networks that are both between mobile participants and ad-hoc. With that comes no present network infrastructure and frequent topology changes. A focus was then first laid on applications and commercial use-cases. Namely temporary networks for military and disaster recovery purposes, Vehicle Ad-hoc Networks (VANET), networks between Unmanned Aerial Vehicles (UAVs) and sensor networks have been looked at. Many different applications of MANETs in specific fields have been discussed.
After that, quality of service (QoS) was introduced as a term standing for an end-to-end quality assurance over the whole system in terms of various different metrics.

Section 6.2 first outlined the problems commonly present in MANETs. It then went on to goals, which help to deal with these problems and should be reached for when designing a MANET. Scalability should allow for significant growth in the number of nodes. Availability should assure the possibility to reach all participants regardless of possible physical link failures. Reconfigurability guarantees the option for changes on the system even when fully set up. Reliability stands for no data loss or duplication, as well as no out of order delivery of packets. Security finally protects the system from malicious activities in various ways, where specific attacks for MANETs have been looked at in detail.

In the end of Section 6.2 common QoS metrics such as bandwidth, delay, energy usage, reliability or packet loss have been identified and classified into additive, multiplicative and concave QoS metrics.

Section 6.3 shows, that stable QoS in a MANET depends on the design of the right routing protocol. It then lists a categorization and provides a collection of common QoS-affine routing protocols, that research has come up with so far. It briefly discusses the way each routing protocol tries to deal with which QoS problem mentioned earlier for each routing protocol listed.

The following conclusion will establish links between the initially presented use-cases and Section 6.3's list of QoS affine routing protocols. It will further shortly outline economic aspects of MANETs and their potentials in general.

6.5 Discussion and Conclusion

The list of routing protocols in section 3 is neither complete nor terminal. Many routing protocols also combine single techniques of previous protocols in a new way. It is impossible to generally name a single outstanding routing protocol. Rather it clearly depends on the chosen application, which of the routing protocols brings the biggest benefit or even which single techniques should be included in a new optimal routing protocol. In that sense we will quickly go through the initially introduced use-cases:

Optimal bandwidth might not be a primary factor for communication between cars (i.e. a VANET), where the primary aim is that they message each other their presence. However, the establishment of this connection with speeds up to several hundred km/h might be one. VANETs have a lot of power - Energy saving algorithms might not be of a big importance either, there.

UAVs in the sky or also battlefield/disaster recovery networks have more of a tendency to rely on high bandwidth for transmitting much more detailed information or even video camera streams and similar bandwidth-intense applications. Bandwidth is therefore a higher priority, the speed the vehicles are moving can be slower *e.g.* for battlefield application on the ground, but just as well a challenge *e.g.* for UAV.

For sensor networks the low use of battery is important as well, moreover the extendability with more sensors and the error-recovery if one sensor fails are important factors to ensure communication. Since further their topology can change very fast, they bring a very high number with challenges with them and need specific routing protocols for these.

Even though end-users will always be interested to be connected to the Internet and will therefore whenever possible chose the classical internet infrastructure with the use of access points, future applications for MANET might show up in various specific fields similar to the present ones. It is very probable that these fields will also bring their new QoS challenges with them. For End-Users one might primarily think about battery lifetime or a high bandwidth for big data transfers.

What almost all MANET application have in common is the challange of moving participants and topology changes. Therefore the first and most efficient factor for improving QoS in MANET should always be minimizing the influence of the nodes' movements [20]. From an economic point of view a big problem is often, that every solution is individual and only proprietary implementations exist. Standards such as e.g. a car communication standard for a usable VANET are still missing. Further, as mentioned already, private end-users lack a need of using MANET with their notebooks or smartphones nowadays. IEEE 802.11 implements the ad-hoc standard, but *e.g.* windows is taking their function "create an ad-hoc network" out already. The replacing standard Wi-Fi direct is so far also not being used even though widely implemented on smartphones and notebooks. Future applications in smart homes are still possible to evolve, though [3].

Bibliography

[1] I. Akyildiz et al.: *A survey on sensor networks.*, Communications magazine, IEEE 40.8, 2002, p. 102-114.

[2] C. Aurrecoechea, et al.: *A survey of QoS architectures.*, Multimedia systems 6.3, 1998, p. 138-151.

[3] J. Carter: *Wi-Fi Direct has no need for the Internet*, http://www.scmp.com/lifestyle/technology/article/1644580/wi-fi-direct-has-no-need-internet, 20.11.2014

[4] S. Chakrabarti and A. Mishra: *QoS issues in ad hoc wireless networks.*, Communications Magazine, IEEE 39.2, 2001, P. 142-148.

[5] J.H. Cho, A. Swarmi, IR Chen: *A Survey on Trust Management for Mobile Ad-hoc Networks*, IEEE Communications Surveys and Tutorials, Vol. 13, No. 4, 2011.

[6] S. De et al: *Trigger-based distributed QoS routing in mobile ad hoc networks.*, ACM SIGMOBILE Mobile Computing and Communications Review 6.3, 2002, 22-35.

[7] E. Huang, W. Hu, J. Crowcroft, I. Wassell: *Towards Commercial Mobile Ad Hoc Network Application: A Radio Dispatch System*, MobiHoc'05, 6th ACM International Symposium on Mobile Ad-hoc Networking and computing, 2005, p. 355-365.

[8] P. Jacquet, et al: *Optimized link state routing protocol for ad hoc networks.*, Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International. IEEE, 2001.

[9] M. Jena and Ajay Rana: *Multi-Constrained QoS Routing Issues in High-Speed Multimedia Networks.*, International Journal of Computer Science and Information Technologies, Vol. 3, 2012, p. 4770-4773

[10] D. Medhi: *Network Reliability and Fault-Tolerance.* Wiley Encyclopedia of Electrical and Electronics Engineering, 1999.

[11] Microsoft: *General Association Operation Guidelines*, http://msdn.microsoft.com/en-us/library/windows/hardware/ff552451

[12] K. Namuduri, Y. Wan, M. Gomathisankaran: *Mobile Ad Hoc Networks in the Sky: State of the Art, Opportunities, and Challenges*, 2nd ACM MobiHoc Workshop on Airborne Networks and Communications, 2013, p. 24-28.

[13] R. Rathee, Mrs. R. Pahwa: *Scalability in QoS Analysis Of Mobile Ad-hoc Network*, Journal of Science and Advanced Information Technology, Vol. 2, No. 3, 2013.

[14] S. Shah and Klara Nahrstedt: *Predictive location-based QoS routing in mobile ad hoc networks.*, Communications, 2002. ICC 2002. IEEE International Conference on. Vol. 2. IEEE, 2002.

[15] E. Slottke: *Wireless Access Systems: Introduction to Vehicular Networks*, ETH Lecture slides for ”Wireless Access Systems”, October, 2014.

[16] S. Sundar, R Kumar, H.M. Kittur, M. Shanmugasundaram: *Manet Routing Protocols with QoS Support - A Survey*, International Journal of Engineering and Technology, Vol. 5 No. 3, 2013.

[17] A. Tonnesen: *Mobile ad-hoc networks*, 2014, `http://www.olsr.org/docs/wos3-olsr.pdf`

[18] M. Tubaishat, K. Sanjay: *Sensor networks: an overview.*, Potentials, IEEE 22.2, 2003, p. 20-23.

[19] M. Uddin and Azizah Abdul Rahman: *Reliability of mobile ad hoc networks through performance analysis of TCP variants over AODV.*, Journal of Applied Sciences Research 7.4, 2011, p. 437-446.

[20] K. Wu, J. Harms: *QoS Support in Mobile Ad Hoc networks*, GSA Journal 2001.

[21] S. Yau, and Fariaz Karim: *Reconfigurable context-sensitive middleware for ADS applications in mobile ad hoc network environments.*, Autonomous Decentralized Systems, 2001. Proceedings. 5th International Symposium on. IEEE, 2001.

[22] Y. Zhang, et al.: *Balancing the trade-offs between query delay and data availability in manets.*, Parallel and Distributed Systems, IEEE Transactions on 23.4, 2012, p. 643-650.

[23] C. Zhu, M. Corson: *QoS routing for mobile ad hoc networks*, Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Vol. 2, 2002, p. 958-967.

Chapter 7

Municipal Wireless Networks

Luis Gerardo Pena Perez

This work provides a comprehensive analysis of Municipal Wireless Networks and their economic impact. The subject is approached from 4 different angles. First, the main motivations behind the deployment of Municipal Wireless Networks are uncovered, and the technologies used to build them are analyzed. Then, several business models for Municipal Wireless Networks are reviewed and the reasons why many of these projects are struggling financially are discussed. In addition, this information is compared against a series of case studies from different cities around the world, which have already implanted Municipal Wireless Networks with different degrees of success. Finally, the paper concludes with predictions about the future of Municipal Wireless Networks.

Contents

7.1 Introduction

Municipal Wireless Networks have become increasingly popular in the last 11 years [2][14]. There are diverse reasons why cities are moving towards this trend [25][33], including the contribution of Municipal Wireless Networks to the quality of life in the municipality. A study performed by NOP Worldwide - Technology on behalf of Cisco Systems showed that 87 percent of users are convinced that access to a wireless connection improves their quality of life by providing flexibility increasing productivity and saving time [10].

Another reason for the increasing popularity of Municipal Wireless Networks is the rise of new technologies that facilitate the implementation, operation, and maintenance of such large scale projects [3]. Along with the technological advances, new business models are being created to run Municipal Wireless Networks and to try turning them into a profitable venture; which, as it will be shown in greater detail on this work, is not always an easy thing to achieve [15]. The aim of this work is to provide a general overview of the current state of Municipal Wireless Networks, their advantages, technical, and economical aspects, as well as presenting examples of previous and existing implementations, with the hope that awareness about the different aspects around of this technological trend can lead to more informed decisions that increase the chances of success during and after the implementation of Municipal Wireless Networks.

7.2 Common Motivations for Deployment

Municipal wireless networks are being adopted around the world for a variety of reasons. The reasons behind the adoption are often dependent on the municipality's needs and vision for the future. The contextual reality of each municipality brings it different challenges and areas of opportunity that can be conquered with the help of municipal wireless networks. In the remainder of this section the most common factors that motivate the deployment of these networks will be presented.

7.2.1 Economic Growth

In today's globalized world cities are in fierce competition against each other to attract businesses and tourism, which bring diversity, prestige, and a significant flow of cash into the municipality. The deployment of Municipal Wireless Networks can provide that extra edge to the municipality in its competition against other regions, making it more attractive to investors and franchises. It can also provide added value to local convention centers [3] and other facilities such as stadiums, theaters, etc. In the same way this networks can also help to attract tourists to the region who bring significant economic benefits to local businesses, restaurants, museums, amusement parks, etc. In addition, all this economic activity can help the municipal government to significantly expand its tax base [25].

7.2.2 Municipal Cost Reduction

Municipal Wireless Networks provide city employees with Internet connectivity thought the municipality, which allow them to support the internal operations and services to the community, such as utility monitoring, law enforcement, and fire protection [3]. Additionally, it increases the productivity of public servants by providing them with easy access to their schedules, email, office systems, and collaboration tools necessary to interact with colleges and external entities [25].

Governments are also increasingly providing online services to its citizens [3]. Paying bills and taxes online or consulting government information though the official website can significantly reduce the amount of resources necessary for the government to be able to provide these services, leading to a reduction in operational costs.

7.2.3 Enhancement of Public Safety

Municipal Wireless Networks can help law enforcement officers on the streets to have better communication with their base of operations, as well as providing them with access to security databases and other systems that increase their efficiency and effectiveness on the field.

The wireless connection makes more cost effective to deploy and manage surveillance systems, such as cameras and sensors, which can be installed in critical areas of the municipality to enhance public safety [25].

The benefits brought by the Municipal Wireless Network can make possible the creation of unified emergency networks used across the different emergency services (fire fighters, ambulances, police, etcetera). This unified service can significantly improve the response time during emergency situations [6].

7.2.4 Breaking the Digital Divide

Internet access can provide social, economic, educational, and cultural advantages to individual citizens, and Municipal Wireless Networks can help to bring these advantages to all members of the community regardless of their socio-economic status or location within the municipality [3][29][22].

A study performed in the USA and presented by Turner et al. [29] provides an idea of the dimensions of the digital divide existing in American households. The study shows that almost 60% of the households with an annual income of 150,000 USD have broadband internet access, in contrast to the households where the annual income is less than 25,000 USD from which only fewer than 10% have internet connection.

Turner [29] found that in 2005 the digital divide between urban and rural areas in the USA was significant. Where the rate of broadband penetration in urban and suburban areas was nearly the double as the penetration rate in their rural counterparts. Recent data from the United States department of commerce [21] shows a significant improvement in the fight against the rural vs urban digital divide in continental USA. However, big gaps still remain in other US territories, as well as in developing countries.

7.3 Technical Aspects

Currently Municipal Wireless Networks are mainly built using mesh Wi-Fi networks, and other supplemental fixed wireless technologies, such as 3G and 4G cellular networks, to take data from the user to a node in the mesh network [3][25].

7.3.1 Why Wi-Fi?

When talking about the technologies that make possible the deployment of Municipal Wireless Networks it is important to have in mind the three main aspects that contribute to the success of Wi-Fi technologies [2] which, as previously mentioned, are currently the base for most Municipal Wireless Networks. These three aspects are:

- 1. The 2.4 GHz and 5 GHz spectrum, in which WiFi operates, does not require a license.

- 2. Standardization of the technology through IEEE and WiFi alliance allows for great interoperability.

- 3. The large scale production of WiFi chipsets has reduced the costs and increased the dissemination of the technology.

These three advantages offered by Wi-Fi have a great weight in the decision of a municipality to choose Wi-Fi over other technologies such as cellular networks alone. For example, as it will be covered on the economic aspects section, reaching a critical mass may be vital for the survival of the municipal wireless network. In this case, using a standard like Wi-Fi which is compatible with virtually all computers, laptops, tablets, and cellphones, provides a significant advantage over cellular networks which require devices to have less widespread LTE or WiMAX chips in order to access the networks.

7.3.2 Wi-Fi Mesh Networks

A Mesh network is formed by several Wi-Fi access points that serve as nodes. Each one of these nodes acts as a repeater allowing data to be bounced from one Wi-Fi device to the other until it reaches its final destination.

A typical two-tier network contains an access tier and a backhaul tier. The access tier usually operates on the 2.5 GHz frequency using the 802.11 b/g mode, to provide connectivity between the mesh routing nodes and their clients. While the backhaul tier operates on the 802.11 a mode which operates on the 5 GHz band to support interconnections among mesh routers. Operating at two different bands mitigates the inter-tier interference [23].

Mesh architectures are known for providing great reliability since each node is connected to several others, in the event of node failure the data can still travel through alternative paths. Another advantage is the scalability of the Mesh networks whose capacity can be expanded by simply adding additional nodes. This feature also allows them to spam through long distances [25] making it possible to provide coverage for entire cities.

The only thing Mesh devices require to self-organize is a power source, which is really convenient for municipal governments since they traditionally control and have access to electric posts throughout the city where mesh devices can be installed, such as public lights, traffic signs, municipal buildings etc. All this points can be used as antennas once the Mesh wireless device has been installed [3]. Figure 7.1 illustrates the distribution of the mesh nodes across an area of the city.

7.3.3 Multi-Hop Cellular Networks

The first Municipal Wireless Networks using cellular technology where built upon 3G networks [7]. The arrival of the 4G is the next generation of cellular networks and is expected to be the replacement for 3G networks. Currently there are 2 high speed mobile technologies that are considered to be the main players in the 4G scene; these technologies are LTE and WiMAX [27]. Multi-Hop cellular networks present several advantages and disadvantages compared with Wi-Fi networks. Some of the most important are: Cellular networks can provide higher speeds (theoretically up to 100 Mbps), over greater distances (50km) and for a greater number of users. On the other hand building cellular towers comes at a very high cost, and WiMAX and LTE enabled clients are not as widespread as Wi-Fi clients. Figure 7.2 shows how a cellular tower station can connect directly to the Internet using a high-bandwidth, wired connection. It can also connect to another tower using a line-of-sight, microwave link. This connection to a second tower (often referred to as a backhaul), along with the ability of a single tower to cover up to 3,000 square miles, is what allows WiMAX to provide coverage to remote rural areas.

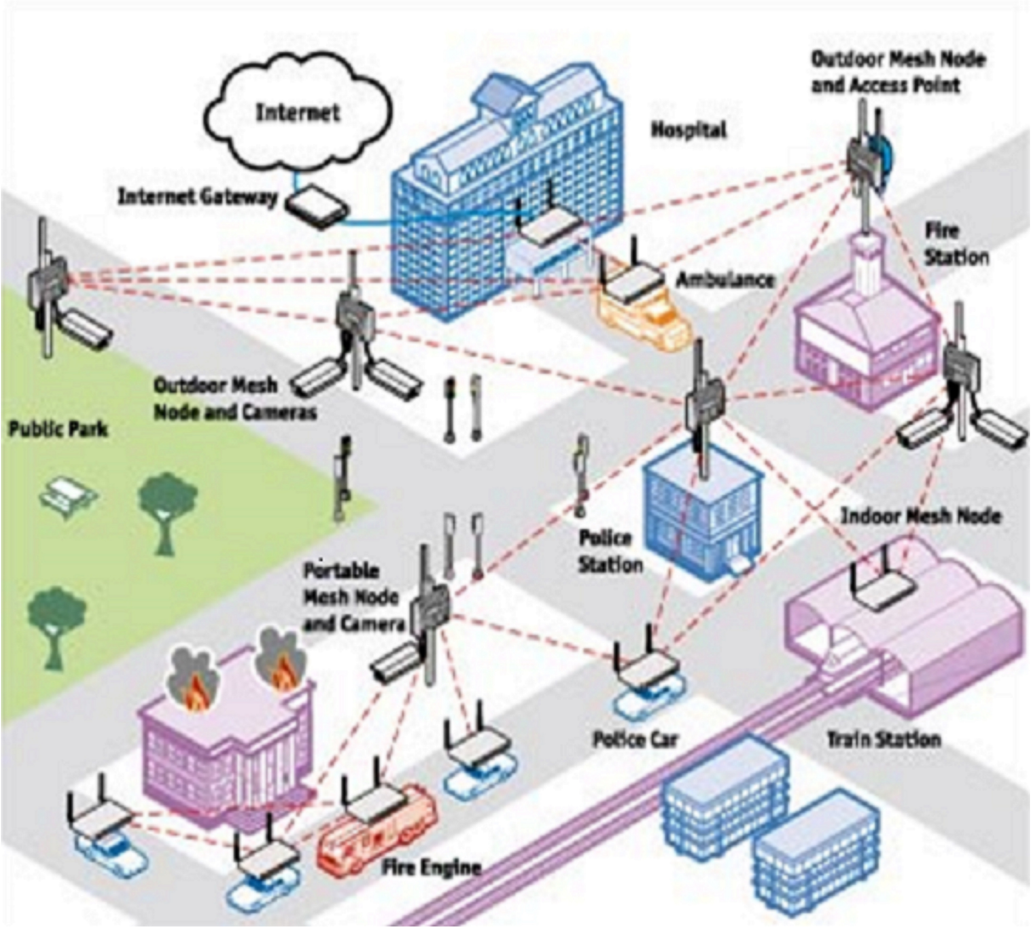


Figure 7.1: Municipal Wireless Network based on Wi-Fi mesh [16].

7.3.4 WiMAX vs LTE

Before talking about WiMAX and LTE it is important to point out that both of them share the same underlying technology. However, there is no interoperability among the two standards.

WiMAX stands for Worldwide Interoperability for Microwave Access and operates under the IEEE 802.16 standard, whose first draft for point-to-multipoint, line of sight with mobile users was proposed in 2004. A year later the standard was amended to include non-line of sight communication with mobile users. This amendment was later known as the IEEE 802.16e standard or Mobile WiMAX. Not much later an additional extension to the IEEE 802.16e standard was made to incorporate multi-hop relaying, this is now known as the IEEE 802.16j standard [27].

LTE on the other hand, stands for Long Term Evolution and consists of a series of standards by the 3GPP organization. Recently further extensions of the LTE standards to have led to the development of the 3GPP LTE-A or LTE-Advance standards [27], which gives devices implementing them the potential to meet the 4G standards for mobile systems set by the International Telecommunication Union [17], which are:

- On a high mobility environment (speed less than 350km/h) a peak data rate of 100 Mbps and an average case latency of 100ms.
- On a low mobility environment (speed less than 10km/h) a peak data rate of 1 Gbps and an average case latency of 10ms.

In recent years both technologies have co-existed in the 4G market, and in practice has been common to find competing carriers using networks based in one or the other technology. For example, in the USA the 4G LTE service from the company AT&T was in

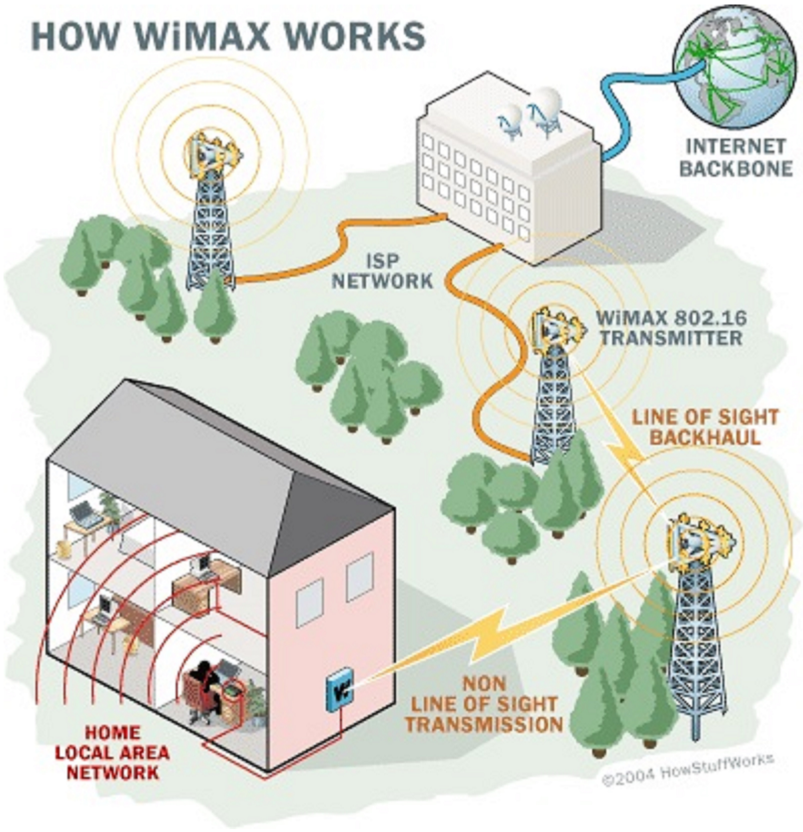


Figure 7.2: WiMAX networks use powerful cellular towers to connect local area networks to the internet backbone using microwave signals [5]

direct competition against the WiMAX service from Verizon and other major carriers [7]. A recent visit to the official websites of AT&T [1], Verizon [30], and Sprint [28] shows that not only they are all offering LTE service now, but is also the main cellular network advertised on their websites.

7.3.5 Hybrid Networks

A municipal wireless network can also be built by a combination of Wi-Fi mesh and cellular networks, to accommodate the needs of the municipality. Here a backhaul layer consisting of cellular towers and fiber connections would pass the data traffic from and to the Internet, while a capacity injection layer consisting of point to multipoint wireless access points would be used to provide connectivity to the mesh access layer comprised of mesh nodes placed on light poles and other places around the city would handle end user traffic [25]. Figure 7.3 illustrates how cellular towers provide connectivity to Wi-Fi mesh nodes through access points in the injection layer.

7.4 Economic Aspects

In this section different business models for Municipal Wireless Networks are discussed. In addition, several causes for financial struggles of Municipal Wireless Networks are uncovered, and recommendations for making these endeavours profitable are made.

7.4.1 Business Models

There are multiple business models that can be used when planning the deployment of a Municipal Wireless Network. Finding the correct business model is a hard task that

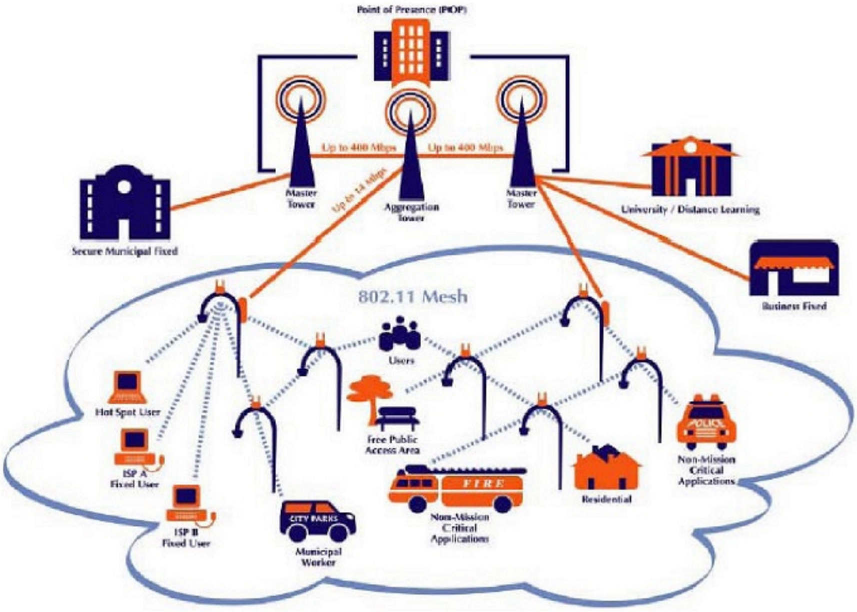
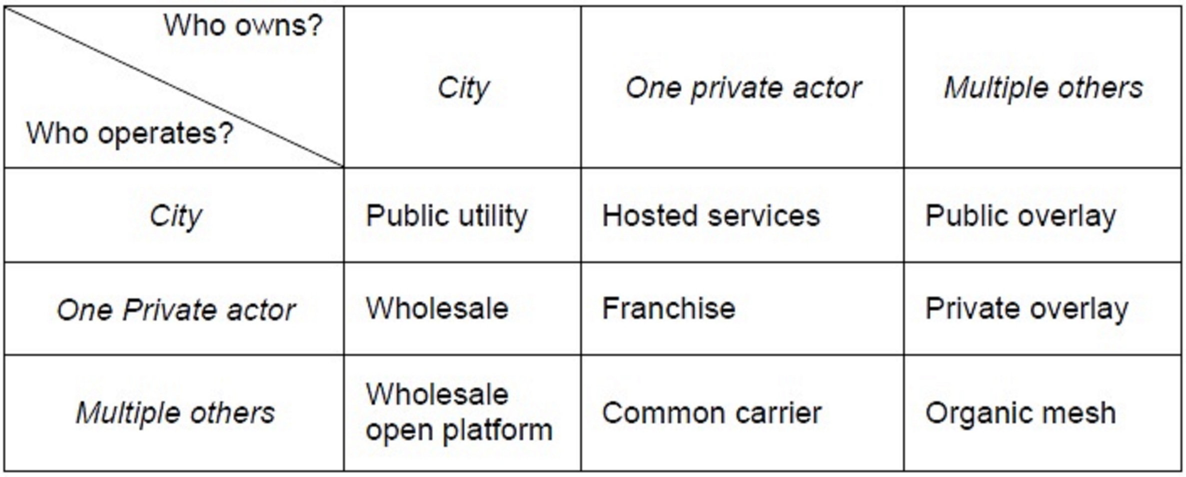


Figure 7.3: This illustration represents a hybrid architecture comprised by Mesh Wi-Fi and fixed wireless network [25].

implicates a lot of responsibility. Selecting the wrong business model can cause the whole project to collapse, depriving the municipality and its citizens from all the benefits that the Municipal Wireless Network could have offered them. Bar et al. [3] make an analysis of multiple existing business models categorizing them based on two dimensions: How owns the network? And who operates the network?. Figure 7.4 shows the nine possible business models for Municipal Wireless Networks.



Who owns? Who operates?	*City*	*One private actor*	*Multiple others*
City	Public utility	Hosted services	Public overlay
One Private actor	Wholesale	Franchise	Private overlay
Multiple others	Wholesale open platform	Common carrier	Organic mesh

Figure 7.4: Municipal Wi-Fi business models according to Bar et al.[3].

In the following subsections the details of each one of these business models will be explained.

7.4.1.1 Public Utility Model

In this model the city owns and operates the municipal wireless network offering the service as another public utility such as water or electricity [3][15].

7.4.1.2 Wholesale and Wholesale Open Platform Models

In the wholesale model the city builds and operates the network, but resells the excess capacity to a private operator (an ISP or telecom company), who retails the service to the general population. This relieves the municipality from all the retail tasks such as searching for customers, providing support, and billing. The wholesale open platform follows the exact same principle but resells the excess capacity of the network to several private operators [3][33].

7.4.1.3 Hosted Services

This option is possible in theory, however, it has never been implemented in practice. Here the municipal government acts as an ISP running its services on privately owned Wi-Fi networks [3].

7.4.1.4 Franchise Model

The private entity owns and operates the network and provides/sells the service directly to the citizens. The municipality often offers the access to antenna sites and uses it to negotiate subjects such as compensation to the city and technical specifications, such as coverage. It is a largely used model because it allows the government to take the role of organizer while delegating the deployment and operation to private entities [3][15].

7.4.1.5 Common Carrier

Here a private network owner would make his infrastructure available to multiple ISPs, or city services. Although, theoretically possible, if the government would demand it, it has not been applied in practice since it doesn't make too much business sense [3].

7.4.1.6 Public Overlay

The efforts from independent private and public entities such as shopping districts, independent businesses, community centers, etc. may give rise to independent uncoordinated networks. Here the government can use its authority to enforce greater coordination and consistency among the Wi-Fi coverage in the municipality. On this model the municipal government can offer a common public overlay to the multiple networks to promote homogeneity, this can be done by having features such as official city branding, common login and authentication service [3].

7.4.1.7 Private Overlay

Here multiple independent private and public network owners outsource the retail side and service provision to a private overlay, which is an operator such as iPass who will take care of the operation of the network [3].

7.4.1.8 Organic Mesh

This is a highly theoretical model in which multiple network owners self-organize into a mesh network, seeking collaboration and interconnection according to their specific needs. Here the participation of the municipality could be expected in the form of promoter of Wi-Fi networks in public facilities such as libraries, and other public places; as well as regulator of co-operation among different parties. For example, through access to antennas [3][4][24].

7.4.2 Financial Struggles

Regardless of the evident benefits of Municipal Wireless Networks, there are plenty of cases where municipalities experience financial struggles to keep their networks running. In the following sub-sections some of the common reasons for these financial struggles are explained.

7.4.2.1 Failure to Reach a Critical Mass

Reaching a critical mass of users is one of the most important aspects for the success of a Municipal Wireless Network, regardless of whether the network is financed through advertising, minimum fees, or taxpayer money. Achieving a pre-determined minimum number of users per day/month allows the network to stay profitable or justifiable, depending on the adopted business model. An example of a Municipal Wireless Network that suffered the consequences of not reaching a critical mass can be found in the city of Orlando, which deployed a free wireless network in one of its main parks expecting around 200 hundred users per day. Unfortunately, after 17 months the network had an average of only 27 users per day. This low number of users made it hard for the municipality to justify the cost of keeping the network running (about 1,800 USD per month), forcing the government to cancel the project [12].

7.4.2.2 Failure to Understand the Technology

There are plenty of examples in literature where the assumptions made about the capacity of the technology used in the Municipal Wireless Network lead to underestimations of the technological investment necessary to carry out the project [15]. As it will be covered in further detail in study cases, failure to understand the physical limitations and behaviour of the technology in the wild may result into cost rises that can put the project in serious financial struggles.

7.4.2.3 Adopting the Wrong Business Model

Choosing the wrong business model can prove to be fatal for the survival of the Municipal Wireless Network [15]. The business model plays a very important role on whether the network will bring financial benefits to the municipality or it will simply not generate the expected returns. Choosing a business model just because it has been successful somewhere else, but that doesn't necessarily accommodate to the municipalities contextual reality can have disastrous results for the Municipal Wireless Network project.

7.4.3 Making Profit

In the following subsections recommendations are suggested to avoid falling into financial struggles and to turn the Municipal Wireless Network into a profitable venture.

7.4.3.1 Understanding the Demand

In order to make a Municipal Wireless Network profitable is necessary to first have a clear understanding of the demand for the services offered by the wireless network [15]. In the same way that failure to reach a critical mass can bring down a Municipal Wireless Network; reaching or surpassing that critical mass can turn the network into a very profitable venture for the municipality.

7.4.3.2 Understanding Technology in the Wild

An understanding of the limitations of the technology used, as well as its behaviour in the wild, can help to save significant amounts of money in equipment deployment and maintenance. But knowing the technology is just one part of this effort. The topology of the municipality, its climate, extension, high building density, and any other factors that may interfere with the signal or any other technological function, should also be carefully taken into consideration when estimating costs to ensure the financial health of the project.

7.4.3.3 Business Models and Sustainability

In order to ensure the sustainability of the Municipal Wireless Network it is important to adopt a business model that is appropriate for the project, as we mentioned before, not all municipalities have the same needs, resources, and or contextual reality. The decision to adopt one business model over another should be tightly linked to how these factors align to create the favorable conditions that make the Municipal Wireless Network profitable in the long run. In addition, a business model can be changed as the contextual reality of the municipality evolves.

7.5 Case Studies

In this Section three study cases are presented, which illustrate the complexities of deploying Municipal Wireless Networks in the real world. These cases reveal the intricate reality of this kind of projects which combines politics, economics, technology, and good timing.

7.5.1 San Francisco, USA

In 2005 the city of San Francisco started the program techConnect, with the intention to provide all of its citizens, especially those in a low socio-economical situation, affordable access to the Internet and online services.

In December 2005 the city issued a request proposal stating that the network should be built, operated, and maintained at no cost to the city, that the entire city should be covered, and that the basic service should be offered free of charge (among other specifications).

The city received 6 proposal, one which was discarded due to incomplete specifications, and 5 more coming from MetroFi, NextWLAN, RedTAP, Seakay (a consortium headed by a non-profit company), and finally a consortium headed by Google and EarthLink.

The proposals were reviewed by 4 city employees holding different IT responsibilities within the municipality were in charge of the evaluation. After comparing proposals and further interviewing the 5 candidates, the consortium of EarthLink and Google were finally selected as the winners. After several negotiations the final contract was signed in January 2007.

After this the project was passed to the city's Public Utilities Commission and the Board of Supervisors, where it encountered several objections ranging from the ascetic changes to the city to user privacy. Finally the matter was subjected to popular vote on November 2007, but by this time EarthLink had already withdrawn from the project [15].

Currently the city of San Francisco counts with a more conservative service offering free public WiFi access only in selected areas and parks [26].

7.5.2 New York, USA

In May 2014 the city of New York issued a request for proposal (RFP) for a franchise contract lasting until June 2026, for the installation and operation of Wi-Fi hotspots, phone service, and advertising on more than 7300 pay phone distributed across the city. The franchise allows charging for phone calls, but the Wi-Fi service must be provided for free. The franchisee then will be allowed to make money through phone calls and advertising while providing free Wi-Fi access to the citizens. Also, the franchisee must compensate the city with a minimum annual amount of 17,500,000 USD or 50 percent of the percentage of the gross sales, whichever is higher [32]. Some of the technical specifications requested on the RFP include [19]:

- The Wi-Fi service must be provided 24 hours a day, 7 days a week
- Must provide a signal strong enough to reach a minimum of 85 feet across a busy street. The Wi-Fi hotspots should work together as a network.
- A user should be able to log in once and stay connected while within 85 feet of any hotspot.
- The user's device should be allowed to automatically re-connect after a connection has been severed and the user comes within the range of one of the network's hotspots.

7.5.3 Chihuahua, Mexico

In 2008 the city of Chihuahua launched a Municipal Wireless Network. The network initially covered two parks and a touristic corridor. However, this was part of a more ambitious project aiming to cover 95 percent of the urban area. By 2009, 474 wireless antennas had been installed [13]. To get an idea of the dimensions of this project, the municipal wireless network on the city of Geneva, Switzerland currently has 290 antennas distributed throughout the city [31].

One of the principal objectives of this network was to strengthen the capacity of the police forces fighting the alarming levels of criminality experienced by the city, and the intensive use of the network and telecommunication systems from private parties was turning to be very costly for the municipality. So, even though this was a multi-purpose network, just the savings in the security sector were enough to convince the government about the value of deploying a Municipal Wireless Network [13].

Once the network was in place the municipality rapidly found alternative uses for it, and multiple government run programs where built around it. Among these programs we can mention:

- Digitalization of municipal services [6].
- Free Internet services for schools, underprivileged neighborhoods and rural areas. Along with hardware donations and opening of training centers [6].
- Real time monitoring of buses and free Wi-Fi internet access in the transit system [11].
- Digital patient records and telemedicine including connection to other health centers in and outside the country [8].
- Integration among all hospital and health centers in the municipality (in progress) [8].

- Plans for a smart street lighting system [9].

After being awarded with the title of Digital City [6], the municipality now keeps working towards further integration in a quest to become a Smart City.

7.6 Predictions

The future seems to be bright for Municipal Wireless Networks for several reasons. First, the social preference towards wireless computing and mobile personal devices and all the advantages that they bring to people's daily life, whether it is for working, playing, or accessing government services [25][20].

Also, in recent years there has been a strong tendency towards integration of services and connectivity, and the concept of smart cities keeps gaining momentum as it promises to deliver a better quality of life to its citizens and even tackle into environmental and health issues which have a global impact.

The technology used in these networks also keeps improving. A 2008 release from NASA announced that they are working in collaboration with M2MI to create a fifth generation or 5G, telecommunications and networking system that incorporates Voice Over Internet Protocol, video, data, wireless, and an integrated machine-to-machine intelligence layer [18]. Considering that in the past a new generation of the technology appeared around every 10 years, this 5G technology is likely to play an important role in the development of Municipal Wireless Networks in the upcoming years.

7.7 Conclusion

This work has explored different aspects around Municipal Wireless Networks. First, an overview of the benefits that the deployment of these networks can bring to the municipality was presented. Here economic growth, municipal cost reduction, enhancement of public safety, and reduction of the digital divide were uncovered as the main motivations for deployment of Municipal Wireless Networks.

Then technical aspects of these networks were explored, including the architecture of mesh networks and the different technologies that may conform them (such as Wi-Fi, XiMAX, and LTE). The report included an analysis of these technologies, as well as the possibility to combine several of them to create hybrid networks.

Also, economic aspects and business models for these networks were discussed, identifying different business models according to two main characteristics of the Municipal Wireless Network: Who operates it, and who owns it.

Within the economic aspects it was shown that failure to reach critical mass, failure to understand the technology and bad business models, can put Municipal Wireless Network projects in financial troubles, and proposed ways to avoid falling in these common pitfalls.

In addition, three different case studies where presented were Municipal Wireless Networks, from three different cities, with different levels of development and success where used to illustrate the real life challenges and rewards of these kind of projects.

Finally, a prediction about the future of Municipal Wireless Networks was made. This prediction offered an optimistic forecast based not only on social preferences and global trends, but also on current work being made by NASA to develop the next generation of cellular networks or 5G, which may play a key role on future of Municipal Wireless Networks.

Bibliography

[1] AT&T Incorporated: AT&T´s Network Now Has the Nation's Strongest LTE Signal; http://www.att.com/network/en/index.html Last updated: 2014 Last accessed: December 5, 2014.

[2] F. Bar, and H. Galperin: Building the Wireless Internet Infrastructure: From Cordless Ethernet Archipelagos to Wireless Grids; COMMUNICATIONS AND STRATEGIES., 45-70, 2004.

[3] F. Bar, and N. Park: Municipal Wi-Fi Networks: The Goals, Practices, and Policy Implications of the US Case; Communications and Strategies, 61(1), 107-125. 2006.

[4] Y. Benkler: Some Economics of Wireless Communications; Harv. JL and Tech., 16, 25, 2002.

[5] M. Brain, E. Grabianowski: How WiMAX Works; http://computer.howstuffworks.com/wimax1.htm. Last updated: 2004. Last accessed: December 05, 2014.

[6] L. Chacon: Uso de TIC para el Desarrollo; Magazine Alcaldes de Mexico. Sep. 2012. Nr.31, 144-119, 2012.

[7] S. Chia, T. Gill, L. Ibbetson, D. Lister, A. Pollard, R. Irmer,and S. Pike: 3G Evolution; Microwave Magazine, IEEE, 9(4), 52-63, 2008.

[8] D. Chihuahua: Chihuahua Cuenta con una Nueva Estrategia Digital en Salud: Secretaría de Salud; http://eldiariodechihuahua.mx/Ciudad/2014-10-13/Chihuahua-cuenta-con-una-nueva-estrategia-digital-en-salud-Secretar\%C3\%ADa-de-Salud/8e551b119878d2043983b8dcaf086992, Official local newspaper. Last updated: Oct 13th, 2014 Accessed: Nov 17th, 2014

[9] E. Chihuahua: Monitoreo Inteligente de Alumbrado Publico; http://www.oem.com.mx/elheraldodechihuahua/notas/n2863263.htm, Official local newspaper, Last updated: Jan 1st, 2013 Accessed: Nov 17th, 2014.

[10] Cisco Systems: Wireless LAN Benefits Study; Conducted by NOP World Technology in behalf of Cisco Systems, 2001.

[11] Comunicacion SEDUE: Technologia; http://www.vivebus.com/index.php/plan-de-movilidad-urbana/featured/tecnologia-del-vivebus, Chihuahua Public Transit Official Website, Last updated: Mar. 12th, 2013 Accessed: Nov 17th, 2014.

[12] D. M. Ewalt: Orlando Kills Municipal Wi-Fi Project; Forbes. June 23rd, 2005 http://www.forbes.com/2005/06/23/municipal-wifi-failure-cx_de_0623wifi.html Last accessed: Oct 5th, 2014.

[13] M. Garza: Ciudades Digitales en México Variaciones Sobre un Mismo Tema; Magazine Politica Digital. Nr.49, 14-17, ISSN 1665-1669, April, 2009.

[14] G. Goth: Municipal Wireless Networks Open New Access and Old Debates; IEEE Internet Computing, v.9 n.3, p.8-11, [doi>10.1109/MIC.2005.62], May, 2005.

[15] H. E. Hudson: Municipal Wireless Broadband: Lessons from San Francisco and Silicon Valley; Telematics and Informatics, 27(1), 1-9., 2010.

[16] Infocom: A Media Friendly Cognitive Resource Management Paradigm for Dynamic Mobile Internet Access with Reliability Guarantees; http://infocom.uniroma1.it/~enzobac/tesi5.html. Last accessed: December 05, 2014.

[17] ITU-R: Requirements Related to Technical Performance for IMT-Advanced Radio Interface; Report m.2134, 2008

[18] C. Mewhinney, M. Curie, and S. Cooper: NASA Ames Partners With M2MI For Small Satellite Development. http://www.nasa.gov/home/hqnews/2008/apr/HQ_08107_Ames_nanosat.html NASA RELEASE: 08-107. April 24th, 2008 Last accessed: November 18th, 2014.

[19] New York Department of Information Technology and Telecommunications, City of: Request for Proposals for a Public Communications Structures Franchise; PIN 8582014 FRANCH3, Release date: April 30, 2014.

[20] S. Nowotny: Cablecom Prüft Flächendeckendes WLAN; 2013, NZZ am Sonntag. http://www.nzz.ch/aktuell/wirtschaft/wirtschaftsnachrichten/cablecom-prueft-flaechendeckendes-wlan-1.18087438, Last accessed: Oct 5th, 2014.

[21] NTIA. United States Department of Commerce: Broadband Statistics Report - Broadband Availability in Urban vs Rural Areas; National Broadband Map based on Census 2010 Block Geography, Published July, 2014.

[22] G. W. Okamoto: Broadband Access for All: The Economic and Political Implications of Municipal Wireless Networks; In Wireless Telecommunications Symposium, WTS 2008 (pp. 380-385). IEEE. April, 2008.

[23] P. H. Pathak, and R. Dutta: Mesh Enabling Technology; In Designing for Network and Service Continuity in Wireless Mesh Networks, 11-35, Springer New York, 2013.

[24] D. P. Reed: How Wireless Networks Scale: The Illusion of Spectrum Scarcity; In International Symposium on Advanced Radio Technology, Boulder, Colorado, March, 2002.

[25] C. C. Reinwand: Municipal Broadband-The Evolution of Next Generation Wireless Networks; In Radio and Wireless Symposium, 2007 IEEE (pp. 273-276). IEEE. January, 2007.

[26] San Francisco, City and County of: San Francisco WiFi; http://www6.sfgov.org/index.aspx?page=246. Last accessed: December 5, 2014.

[27] S. Sharma, and N. Payal: 4G ERA; In Proceedings of the 2014 Fourth International Conference on Advanced Computing and Communication Technologies, 377-380, IEEE Computer Society, 2014.

[28] Sprint Corporation: Works when and where you need it; `http://network.sprint.com/index.html`. Last updated: 2014 Last accessed: December 5, 2014.

[29] S. D. Turner, and F. Press: Broadband Reality Check, the FCC Ignores Americas Digital Devide; Free Press, 2005.

[30] Verizon Wireless: America´s Largest and Most Reliable 4G LTE Network; `http://www.verizonwireless.com/wcms/consumer/4g-lte.html`. Last updated: 2014 Last accessed: December 5, 2014.

[31] Ville de Genève: Accès Wi-Fi; `http://www.ville-geneve.ch/themes/environnement-urbain-espaces-verts/acces/` Last updated: May 21, 2014 Last accessed: Oct 5th, 2014.

[32] E. Vos: NYC Issues RFP for Free Citywide Wi-Fi Service; MuniWireless, April, 2014. `http://www.muniwireless.com/2014/05/02/nyc-rfp-free-citywide-wifi/` Last accessed: Oct 5th, 2014.

[33] Z. Yang, S. Khamit, A. Mohammed, and P. Larson: A Comparative Study on Business Models of Municipal Wireless Cities in US and Sweden; In Business-driven IT Management. BDIM 2008. 3rd IEEE/IFIP International Workshop on (pp. 116-117). IEEE. April, 2008.

Chapter 8a

Fairness Notions of Multi-Resource Allocation in Shared Computer Infrastructures

Matthias Kaenzig

Fair multi-resource allocation in computer infrastructures is a relatively new topic and currently researched in many directions. Cloud computing services are gathering more and more attention throughout society and are already heavily in use. It is therefore of fundamental importance to come up with good mechanisms which ensure fair sharing of such clusters among the users. This paper lists particular problems which arise when dealing with multi-resource allocation and gives an overview of the current mechanisms which are so far developed. We show that there is no objective consensus on fairness in a multi-resource allocation domain in general but indeed some overall agreement on what properties every fair allocation should satisfie. We further show that there exists an actual fairness-efficiency tradeoff - fairness can therefore only get achieved at the expense of lowering the efficiency to some extent.

Contents

8a.1 Introduction

Fair resource allocation is of fundamental importance in today's computing systems. Data is widely stored in clouds and more and more computational work is getting done remotely in shared computer infrastructures. In such environments, there is a strong need of allocation mechanisms which ensure that every agent gets his fair share of the whole cluster to run his jobs. Such clusters provide different resources such as CPU, RAM, bandwidth and disk I/O, whereas jobs need different quantities of these resources in different combinations. In general, requirements are very heterogeneous - there might be jobs which require much bandwidth but little CPU while others need more CPU but less bandwidth. To deal with such demands, we need multi-resource allocation mechanisms. Smaller clusters, which are e.g. common in research facilities, usually not implement usage policies based on pricing mechanisms but rather in a way where everyone currently using the cluster theoretically has access to the whole resource-pool. We therefore need mechanisms which guarantee fair allocations such that no one can complain about his allocated part of the cluster. Fair single resource allocation mechanisms have been widely studied for the past years and are quiet well developed. Prominent policies such as *max-min fairness* allocate every agent, assuming high enough demand, identical entitlements of the whole resource. This can be intuitively called fair - everyone gets the same. Furthermore, in a single resource scenario, the most efficient allocation allocates all of the available resource, which is a highly desirable property. This is not true in a multi-resource allocation domain: due to heterogeneous demands, it is often not possible to allocate all of the resources entirely. In addition, we will see that giving everybody the same of any resource type can lead to inefficient allocations and therefore should be avoided.

Fair multi-resource allocation in computer science is a relatively new topic and therefore currently researched in many different directions. This paper lists the fundamental problems which arise when dealing with such allocation mechanisms and tries to give an overview of the mechanisms so far developed. Section 8a.2 clarifies the problem of defining fairness while section 8a.3 discusses particular problems for multi-resource allocations. Section 8a.4 illustrates some particular multi-resource allocation mechanisms. Section 8a.5 focuses on the fairness-efficiency tradeoff in multi-resource allocation and section 8a.6 discusses the findings. Section 8a.7 finally opens the topic for some discussion.

8a.2 Fairness Notions in Multi-Resource Allocation

Cambridge Dictionaries Online define fairness as ”the quality of treating people equally or in a way that is right or reasonable”. [1] However, what is right or reasonable? It is too easy to say that ’right or reasonable’ claims that, facing an allocation problem, everybody should get the same. But if not, what notions of fairness should we take into account when developing a new allocation mechanism? As we will see later in the paper, there is indeed no such thing like ’perfect fairness’ when dealing with multi-resource allocations. Different notions lead to different mechanisms - all of them could get called fair with respect to their reflections on fairness.

Although there seems actually not to be a consensus on fairness in general, we can nevertheless come up with a highly desirable property for any fair allocation. We will use it to analyse different allocation mechanisms and make them in some way comparable against each other. Thus, any fair allocation should fulfil at least the following property:

Envy-freeness A user should not prefer any allocation of another user to his own allocation

[1] `http://dictionary.cambridge.org/dictionary/british/fairness`

If we achieve *envy-freeness*, then a allocation can get called fair. If anyone gets the particular allocation which suites him the most, then no one has an incentive to envy another user in the system. To make a point, an envy-free allocation does not imply that every user is best off with his allocated quota in general but rather in this particular allocation. Let us assume two users; each of them want to run a job which requires all of the available CPU delivered by some cluster. To be best off in general (or to maximize his payoff) for a particular user would mean to get all the available CPU. This would be not envy-free at all - an envy-free allocation would allocate both users half of the CPU. This maximizes payoffs for both users in the particular envy-free allocation but does not ensure that a user may favour a totally different allocation in general. This already shows that *envy-freeness* is not bounded by a unique allocation for a particular input, but rather can get achieved by different allocation mechanisms.

8a.3 Multi-Resource Allocation Problems in Computing Systems

Allocating multiple resource types at once leads to some problems which are not always trivial to tackle. First of all, agents request bundles of goods. This is different then demanding only a single good in that sense that we must deal with vectors of requirements rather then just with scalars [5]. To map these vectors to a scalar and therefore be able to compare the different allocations, we must know the users utility functions. Although such utility functions can not be considered known in general [9], let us assume users can report their needs for jobs via some preference functions. In economics, such functions are in general differentiable and therefore imply some interchangeability among the resources - this does not apply for computational resources in a cloud: it is rather infeasible to substitute RAM with CPU, the same goes with the other resources available in computing systems. We therefore have to relax this requirement and assume *Leontief preferences*: user demand resources in fixed ratios. Most popular work on multi-resource allocation such as [2] and its extensions [3, 6, 7, 10] as well as [1] restrict user-preferences in such a way - we'll have a look at mechanisms which does not require such preferences in section 8a.4.1 and 8a.4.4.

A second problem arises when facing efficiency: a efficient allocation in a single resource domain always allocates all of the available resource. That is not the case in a multi-resource domain, especially if we enforce *Leontief preferences*. It is even unclear how to measure efficiency, as mentioned by Wong et al. [5]. Should we maximize the total amount of allocated resources or rather the total number of jobs? Additional fairness constraints make this even more complicated - any envy-free mechanism suffers from significant drawbacks in efficiency compared to mechanisms which are not bounded by such constraints. We will have a look at this in section 8a.5.

8a.4 Multi-Resource Allocation Mechanisms

An allocation mechanism is described as follows: it takes preference profiles as input and generates a feasible allocation as ouput. To measure a mechanism's capabilities, there are different properties. We will focus on the following ones for our analysis:

Sharing-Incentiveness Every user should be better off sharing the whole cluster than insisting on his partition. If there are n users, than the mechanism should allocate every user such a part that he is at least as happy as getting $\frac{1}{n}$ of every resource available.

Pareto-Efficieny It is not possible to make any user better off without making another user worse off.

Strategy-Proofness Every user should be best off if and only if he reports his true preferences.

Sharing-Incentiveness ensures at the same time *No-Starvation* of tasks as it guarantees at least $\frac{1}{n}$ of the resources to every participant. *Strategy-proofness* is of high importance because of the manipulative nature of some users. If there are possibilities to get better off by miss-reporting demands, then sooner or later such a feature will be missused: there was a big search company which provided dedicated machines only if one could ensure high utilization profiles - to simulate such utilization, users began to include infinite loops into their code [2]. A mechanism is only capable of generating outputs based on reports, not true values. Hence it is of particular interest to come up with *strategy-proof* mechanisms and therefore avoid non-truthful behaviour.
To compare the performance of the following allocation mechanisms on some actual data, let us come up with a simple example configuration, which we will use as input: there are two resources, CPU and RAM, each of them available 20 units in total. There are further two agents, user 1 and user 2. Every user needs to run tasks with particular requirements as follows:

User 1	<**5** CPU, **2** RAM>	User 2	<**3** CPU, **4** RAM>

Every user wants to launch as many tasks as possible. Any mechanism will therefore allocate as much as possible considering its particular constraints.

8a.4.1 Hadoop Fair Scheduler

Let us come up first with an mechanism which is actually widely in use: the Hadoop Fair Scheduler. It is based on a very simple policy: every user gets his equal share of the whole cluster. A cluster is hereby divided into n pools, each of them holding roughly $\frac{1}{n}$ of the total amount of resources available; by default, every user is assigned one pool. If a pool does not need its full capacity, excess is split among the other pools [4]. If we run the Hadoop Fair Scheduler with our sample input, it allocates both of the users exactly <10 CPU, 10 RAM>. This results in the following total utilization:

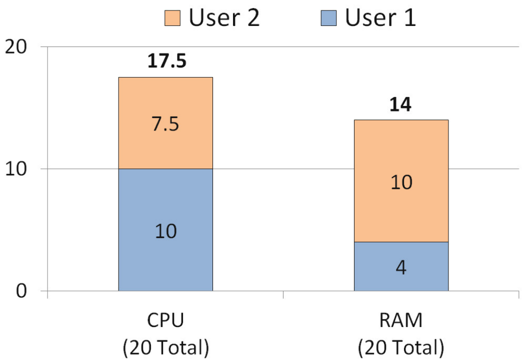


Figure 8a.1: Hadoop Fair Scheduler: Sample utilization profile

As we can see, overall utilization is not really good. Due to heterogeneous and somewhat opposing demands of user 1 and user 2, the Hadoop Fair Scheduler does not perform very well in this particular situation. It lacks the possibility to react on different user demands and solely allocates tasks in a fixed-slot based manner. User 1 is not able to use his 10 RAM without more CPUs whereas user 2 can not benefit from 10 CPUs. Allocating fixed slots

is too often more than a poor match for an actual task's demand and leads to inefficient allocations. Their is an incentive to come up with allocation mechanisms which are capable of handling such heterogeneous demands in a more efficient way. Nevertheless, the Hadoop Fair Scheduler actually satisfies 3 out of 4 properties we introduced earlier, namely *envy-freeness*, *sharing-incentiveness* and *strategy-proofness*. What it lacks is *pareto-efficiency* - there are actually other allocations which would make both users better off.
Except for the low efficiency, the Hadoop Fair Scheduler actually comes with some very nice properties which makes it particularly easy to use. It does not allocate resources based on preferences but rather just gives everybody the same, what makes the mechanism independent of effective preference functions. This is highly appreciated, because such functions are often not available in a realistic environment.

8a.4.2 Dominant Resource Fairness

Dominant Resource Fairness (DRF) was introduced by Ghodsi et al. [2] in 2011 and since then became the most popular fair multi-resource allocation policy so far. It is in fact a generalization of *max-min fairness* for multiple resources based on every users *dominant share*, whereby the dominant share for every user is defined as the highest demanded resource in percentage across all the demands for that user. Let us do an example with our sample configuration: every task of user 1 consumes $\frac{1}{4}$ of the total CPUs and $\frac{1}{10}$ of the total available RAM. CPU will therefore be user 1's dominant share. User 2 needs $\frac{3}{20}$ of all CPUs and $\frac{1}{5}$ of the total available RAM for a task, so his dominant share is RAM. The DRF allocation is then computed as follows:

$$\begin{array}{rrcll} \max & x+y & & & \\ \text{s.t.} & 5x+3y & \leq & 20 & (a) \\ & 2x+4y & \leq & 20 & (b) \\ & \frac{x}{4} & = & \frac{y}{5} & (b) \end{array} \tag{8a.1}$$

We want to maximize the total number of jobs subject to our resource constraints (a) and (b) and, what is most important, under the constraint which ensures that every user gets the same amount of its dominant share (c). Ghodsi et al. mention that their is no need to always equalize every user's dominant share: when a particular user's demand is fulfilled and there is still enough resources to allocate another user still more tasks then this should not be prohibited [2]. It is possible to model such OR-constraints with suitable tools in a linear program. Now let us have a look at the total utilization generated by the DRF allocation mechanism:

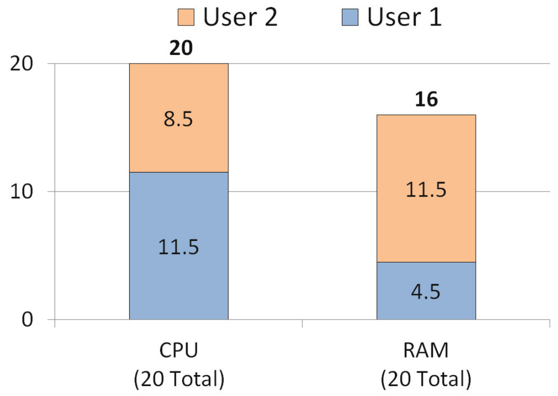


Figure 8a.2: Dominant Resource Fairness: Sample utilization profile

Overall utilization is significantly higher compared to the Hadoop Fair Scheduler allocation (see fig 8a.1). This is due to the fact that DRF actually allocates resources with respect to heterogeneous demands. DRF tries to allocate each user the maximum share of what he needs the most, namely his dominant share. If every user has the same dominant share, then DRF actually reduces to single *max-min fairness* for this particular resource. The DRF mechanism satisfies *envy-freeness*, *sharing-incentiveness*, *strategy-proofness* and *pareto-efficiency* and therefore all 4 predefined desirable properties we want our mechanisms to have. Unfortunately, these properties are often only fulfilled in a very theoretical environment; that is why we want to analyse the DRF mechanism a bit more in detail in the following section to get some very interesting insights.

8a.4.2.1 Limitations and Extensions

As DRF allocations are in fact linear programs with some additional constraints, they will in general produce fractional solutions for the decision variables, namely the number of jobs allocated to every user. This fact is not handled consistent over the whole DRF-paper: Ghodsi et al. on one hand solve allocations with linear programs and on the other hand propose a scheduling algorithm to generate DRF allocations which actually assumes tasks to be indivisible.

In reality, there is most probably some resource (e.g RAM) of which a task needs some minimum amount to run - divisible tasks are therefore not always suitable. The problem is that, assuming indivisibilities, we loose some core properties of the DRF mechanism. Parkes et al. showed that it is actually not possible to come up with a mechanism which, under indivisibilities, satisfies *envy-freeness* and *pareto-efficiency* at the same time [7]. This is bad in a sense that we either loose efficiency or fairness - both properties of fundamental importance for good and fair allocation mechanisms. This incompatibility can easily be shown on a simple example: let us have two resources, and let the total amount of each resource be 1. There are further two users, both need to run tasks with demand <1/3, 0>. If we enforce *envy-freeness*, then the only feasible allocation is allocating one task to each of both users. But this is not efficient at all - there could be executed a third job on the cluster without making anyone worse off. Parkes et al. come up with a relaxation of the *envy-freeness* property called *envy-free up to one bundle (EF1)* which is defined as follows: a mechanism is *EF1*, if no agent would actually envy another agent if this agent has one task less allocated [7]. Allocating <2/3, 0> to one user and <1/3, 0> to the other one in the example above would actually be *EF1* and pareto-efficient. Under this relaxation of *envy-freeness*, they actually develop a mechanism based on DRF which is *pareto-efficient* and *EF1* at the same time.

Beyond that, Ghodsi et al. assume one resource pool to run DRF on. This is in fact no very realistic. In general, today's clouds consist of many heterogeneous machines combined to one cluster. As shown by Wang et. al, it is not feasible to just apply DRF on each node as this results in poor efficiency and is not longer *pareto-efficient* [10]. They come up with a new mechanism called DRFH, which is a extension of simple DRF to a multi-node-environment. Basically, they seek to equalize every users *global dominant share* across all nodes subject to single-node resource constraints - e.g the mechanism must also ensure feasible allocations on node-level and not only be bounded by the total amount of available resources. Their adaptation of DRF to DRFH actually achieves better overall utilization in heterogeneous server environments then simple DRF in such settings [10].

Furthermore, DRF is only defined for static settings: it matches revealed demands (the actual input) to an output - a more realistic utilization profile of a cluster would consist of users coming and going, therefore a mechanism should be expected to be able to dynamically allocate tasks over time. The scheduling algorithm proposed by Ghodsi et al. assumes indivisible tasks and therefore, assuming identical inputs, does not lead to the

same allocations as DRF ran as a linear program in general, not to mention the properties we loose when assuming indivisibilities. There exist an extension over DRF which tackles that problem and lets the user arrive over time - but never depart - and then dynamically reallocates resources over time [6]. What is highly inefficient about that mechanism is the fact that it lets actual capacity unused - the first user which arrives over time only gets allocated at most $\frac{1}{n}$ of the total resource pool, even if there is no other agent right know using the cluster. This is at the same time somewhat inconsistent referring to what they propose: the paper wants to come up with a dynamic allocation mechanism, but on the other hand wants to know in advance how many agents there will arrive.
Last but not least, DRF assumes known utility functions. Gathering such information is on one hand technically challenging and on the other hand may not be accepted by some users due to privacy considerations [8]. It is therefore not realistic to apply DRF from scratch to allocate a cluster for the first time but rather computing reallocations based on current utilization profiles, assuming we do not have initial information on preferences. We can even argue on the Leontief-preferences requirement: in general, e.g more CPU is most often always preferred by any task - DRF does not take this into account as it expects tasks to require resources in fixed ratios.

8a.4.2.2 Evaluation

Dominant Resource Fairness comes with highly desirable properties and generates significantly higher total utilization then just allocating every user in the cluster roughly its legit quota (e.g Hadoop Fair Scheduler, section 8a.4.1). Under theoretical aspects, it actually satisfies all four of our predefined mechanism and allocation properties: *envy-freeness*, *sharing-incentiveness*, *strategy-proofness* and *pareto-efficiency*. Unfortunately, some of these properties get lost if we apply DRF in environments where we enforce things like indivisible tasks or task scheduling over time. DRF further supposes known utility functions - this assumption is often not realistic.

8a.4.3 Bottleneck Based Fairness

Bottleneck Based Fairness (BBF) was first introduced by Dolev et al. [1] in 2012. Based on existing work on fair multi-resource allocation, they come up with the following definition of a fair allocation:

No Justified Complaints Condition "A user cannot justify complaining about his allocation if either he gets all he asked for, or else he gets his entitlement [of at least some bottleneck resource]." [1]

This definition shows the somewhat different notion of fairness BBF tries to achieve then DRF. The first part is the same: both BBF and DRF do not insist of allocating every user the same when there is potentially someone who does not need its full entitlement. What makes the difference is the second part of the sentence: BBF focuses on bottlenecks, whereas DRF only tries to equalize dominant shares. This approach of DRF in fact can in some cases be seen as not totally fair. Why should a user get restricted based on a resource of which there is plenty of in the system? This is in fact true for such demands where the dominant share is actually not the bottleneck resource. Let us come up with our sample configuration again: it is easy to see that, if we start allocating tasks to user 1 respectively user 2, CPU will become our bottleneck (this is true for the DRF allocation as well, see fig 8a.2). In such a case, BBF allocates both of the users their entitlement of this bottleneck resource. For simplicity, let us assume each user is entitled $\frac{1}{2}$ of the resource. This would lead to the following utilization profile:
We can see that the total utilization is even higher as it is with DRF (fig 8a.2). This is

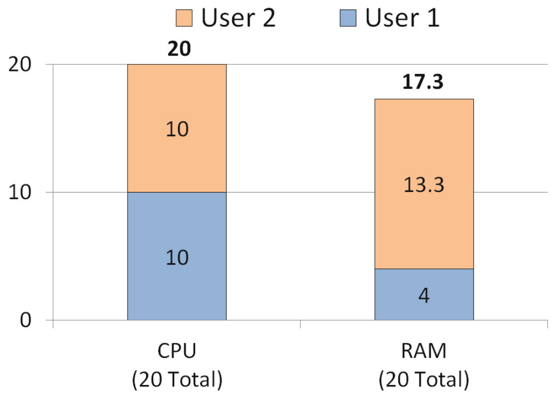


Figure 8a.3: Bottleneck Based Fairness: Sample utilization profile

due to the fact that user 2 is not longer restricted by its dominant share RAM - he can actually use more of it because it has not to be equal to user 1's dominant share CPU any more. What makes in fact sense - why should he actually get restricted on a resource for which there is no contention at all? While BBF indeed leads to high utilization and satisfies *envy-freeness*, *sharing-incentiveness* and *pareto-efficiency*, it turns out to actually be not *strategyproof* [8]. The allocations produced by BBF are further not unique - this is considered as a potential advantage by Parkes et al. as it may leave some space to have the possibility to pick the particular allocation with regard to some secondary goals [1]. We value this circumstance rather negative, because it makes evaluation difficult and ends up in potential dissatisfaction among the users for a particular allocation. As BBF supposes similar initial configurations and in general makes rather the same assumptions as DRF does, the limitations which apply for DRF illustrated in section 8a.4.2.1 also hold for BBF. What BBF totally lacks is a scheduling policy: BBF allocations can only be computed in a static way. This is a major drawback in our opinion since scheduling tasks is of fundamental importance in a computing environment.
Bottleneck Based Fairness should be considered as worthy alternative to Dominant Resource Fairness - it leads to very high utilization and satisfies core properties of fair multi-resource allocation mechanisms such as *envy-freeness* and *sharing-incentiveness*. On the contrary side we have got non-unique allocations and an overall rather complex allocation algorithm which is not *strategy-proof*. Furthermore, BBF allocations are not schedulable. This might be the reason why research on BBF is not continued by now.

8a.4.4 Greediness Metric

The Greediness Metric (GM) is a method to reallocate resources in computing systems based on monitoring current utilization. It is currently developed by P. Poullie [8]. In contrast to DRF and BBF, it does not compute initial allocations based on revealed utility functions but rather focuses on existing utilization profiles of clusters and the reallocation of its resources. Poullie et al. mention that today's clouds would often apply statistical multiplexing and it therefore is only of importance to apply a fairness policy when there is actual scarity of some resources [9]. This is not a contradiction to what DRF and BBF assume, but what the GM handles different is the actual (re)allocation. DRF and BBF both suppose tasks to require resources in fixed ratios (see section 8a.3 for *Leontief preferences*) and therefore cap the maximum amount of each resource receivable based on these ratios. If a task needs 3 CPUs and 2 RAM and the user gets allocated 4 RAM, DRF and BBF both assume that the user would not benefit from potentially more then 6 CPUs - this is often too simplistic. The GM handles this problem in a different way: tasks are not considered to require fixed ratios of resources. If a resource is actually congested, then GM reallocates this resource based on how much every user is demanding from the other resources.

GM allocations furthermore satisfie *envy-freeness*, *sharing-incentiveness*, *pareto-efficiency* (on the bottleneck resource) and *strategy-proofness*: it is trivial to see that as long as there is enough resources left (no bottleneck), it is feasible for every user to demand actually more of some resources then its legit partition (*sharing-incentiveness*). Every user is furthermore best off to demand its true requirements (*strategy-proofness*) - and if everyone gets what he wants, this must be *envy-free*. As it comes to scarcity, the bottleneck resource actually gets splitted according to every users greediness on the other resources. The only way a user could actually get more of the bottleneck resource is claiming he needs less of the other resources, but monitoring the requirements makes this an impossible try.

What makes the Greediness Metric actually stand out from other mechanisms like DRF and BBF is the fact that, in our opinion, it focuses on rather realistic scenarios then just theoretically constructs. Statistical multiplexing is common in today's clouds and reallocating resources therefore is often more important then computing allocations from scratch. The GM allocation mechanism can furthermore be considered as capable of scheduling tasks dynamically - this is highly appreciated for a resource allocation mechanism in computer environments.

8a.5 Fairness-Efficiency Tradeoff

As we have already seen in section 8a.4, there is always some tradeoff between fairness and efficiency when trying to allocate scare resources in some fair manner. Every fairness policy restricts an allocation from being as efficient as if there were no such limitations: to see that on actual data, let us run our sample configuration as a simple linear program without any fairness constraints (equation 8a.1 in section 8a.4.2 wihtout constraint (c)) and look on the utilization profile:

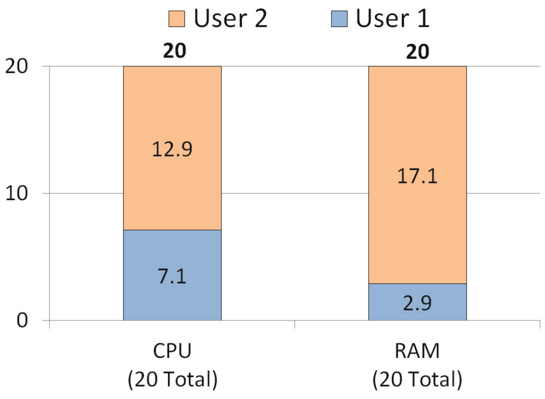


Figure 8a.4: No fairness constraints: Sample utilization profile

As we can see, every resource is allocated all of its 20 units. This is efficient - but not fair. The allocation is not *envy-free* nor does it provides *sharing-incentiveness* - user 1 would be better of just insisting on his fair half of the cluster then participating in such an allocation mechanism. On the other hand, such a mechanism is not truthful: users could lie about their actual demands and therefore get allocated more of the cluster.

We can state that, given an allocation mechanism, adding additional fairness constraints in general always lower total efficiency generated by the mechanism. There might be some particular input configurations which may lead to actual similar solutions, but typically not. What we have further seen in section 8a.4 is the fact, that different mechanisms perform on different levels for particular inputs. In our particular example, BBF performed better then DRF in case of efficiency - this can be the other way round for other configurations as for instance shown in [1].

8a.6 Conclusion

Based on initial fairness notions for multi-resource allocations, we have illustrated four different allocation mechanisms. Each of them generate fair allocations in that sense that all of them satisfie *envy-freeness*, a property of fundamental importance when analysing fair allocation mechanisms (see section 8a.2). Although there actually seems to be some consensus on what fair is, every mechanism however generates its own, distinct allocations. This shows that there is no way to objectively describe fairness in a multi-resource allocation environment. How to fulfil *envy-freeness* is not determined and therefore can get achieved in many different ways. Good mechanisms furthermore satisfie *sharing-incentiveness*, *pareto-efficiency* and do not let users benefit from lying about their actual demands, what we call *strategy-proofness* (see section 8a.4). These properties are satisfied by Dominant Resource Fairness and the Greediness Metric, but not by Bottleneck Based Fairness which is indeed not truthful.

There is furthermore some general disagreement on what the initial configurations are: DRF and BBF both describe mechanisms which generate allocations from scratch - they assume known utility functions and based on them computed some particular outcomes. The GM on the other hand is being developed to rather handle reallocations based on current utilization profiles. While it is possibly feasible to use DRF and BBF on monitored inputs as well, both mechanisms depend on rather strong theoretical aspects which can often not be considered as a given in a realistic scenario.

Beyond that, we have seen that every fair mechanism which satisfies *envy-freeness* does actually suffer from more or less serious efficiency drawbacks. It is actually not possible to maintain efficiency and at the same time enforcing *envy-freeness* - these are two rather incompatible properties. With regard to a particular input, different mechanisms perform with varying degrees considering efficiency - there is actually no best mechanism in general as performance depends on the given input.

8a.7 Discussion

Fair multi-resource allocation mechanisms for shared computer infrastructures are important in today's world - resources need to get distributed among the users in such am manner that everyone is willing to actually use such products like cloud computing services. Is it however somewhat different when looking from a provider perspective, where maximizing efficiency is definitely higher rated then guaranteeing every customer its fair share. One can argue that as long as service level agreements (SLA) are fulfilled, their is in fact no need to go further and actually ensure things like *envy-freeness* throughout allocations. On the other hand, how can a particular customer in fact verifying its current allocation based on such fairness notions? It is therefore justified to come up with the question how big the willingness for large providers to actually implement such fairness concepts in fact is. Why should they prefer fairness over efficiency? Fields to actually use such fair allocation mechanisms may therefore rather lie in private clouds. In such environments, congestion is more likely and users can easier verify if they receive a fair share.

What we further missed throughout the current work is some actual focus on dynamic allocation policies. Most of the mechanisms lack scheduling implementations and only consider the static setting. We think dynamic allocation is of fundamental importance when it comes to allocating (possibly multiple) resources in computer infrastructures.

Bibliography

[1] D. Dolev, D. Feitelson, J. Halpern, R. Kupferman, and N. Linial, *No Justified Complaints: On Fair Sharing of Multiple Resources*, 3rd Innovations in Theoretical Computer Scinence Conference, Cambridge, USA, 2012, pp. 68-75

[2] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica: *Dominant Resource Fairness: Fair Allocation of Multiple Resource Types*, University of California, Berkeley, USA, 2011

[3] A. Gutman, and N. Nisan: *Fair Allocation Without Trade*, In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, Valencia, Spain, 2012, pp. 719-726

[4] Hadoop Fair Scheduler, *Hadoop 1.2.1 Documentation*, `http://hadoop.apache.org/docs/r1.2.1/fair_scheduler.html`

[5] C. Joe-Wong[1], S. Sen[1], T. Lan[2], and M. Chiang[1]: *Multi-Resource Allocation: Fairness-Efficieny Tradeoffs in a Unifiying Framework*, [1]Departement of Electrical Engineering, Princetin University, Princeton, USA, [2]Departement of Electrical and Computer Engineering, George Washington University, Washington, USA, 2012, pp. 1206-1214

[6] I. Kash, A. Procaccia, and N. Shah: *No Agent Left Behind: Dynamic Fair Division of Multiple Resources*, 2013 International Conference on Autonomous Agents and Multi-agent Systems, St. Paul, Minnesota, USA, 2013, pp 351-358

[7] C. Parkes, A. Procaccia, and N. Shah: *Beyond Dominant Resource Fairness: Extensions, Limitations, and Indivisibilities*, In Proceedings of the 13th ACM Conference on Electronic Commerce, Valencia, Spain, 2012, pp. 808-825

[8] P. Poullie; *Decentralized Multi-Resource Allocation in Clouds*, Departement of Informatics, University of Zuerich, Zuerich, 2013

[9] P. Poullie, B. Kuster, and B. Stiller: *Fair Multi-resource Allocation in Clouds*, Departement of Informatics, University of Zuerich, Zuerich, Switzerland, 2014

[10] W. Wnag, B. Li, and B. Liang: *Dominant Resource Fairness in Cloud Computing Systems with Heterogeneous Servers*, Departement of Electrical and Computer Engineering, University of Toronto, Toronto, Canada, 2013

Chapter 8b

Fairness Notions in Single Resource Allocation

Moritz Baggenstos

This paper discusses fairness in networks. But what exactly is fairness? We all judge fairness in a relative way. The perception of what is fair is based on comparison with others. It is difficult to define exactly what is fair and what is not. It is even more difficult to say a resource allocation is more fair than another resource allocation. Fairness has not the same meaning for everyone which makes it very difficult to define a good fairness measure. The tradeoff between fairness and utilization of a network is discussed. This paper presents ideal properties of a good fairness measurement as well as some of the most common fairness measures which can be applied to today's networks. There are some algorithms and methods presented which will help to allocate the resources of a network equally to all users. This paper discusses the max-min fairness algorithm, Jain's Index, Alpha Fairness, the ratio between the smallest and the largest entries as well as the Proportional Fairness. Five axioms of a good fairness measure are included as well in this paper.

Contents

8b.1 A Definition of Fairness

"Fairness is the quality of treating people equally or in a way that is right or reasonable." [1] If a system is not fair for it's participants than they don't participate voluntarily. So in order to get people to participate in a network some level of fairness has to be guaranteed. Fairness is usually a qualitative measurement. This means the level of quality is important. It is difficult to compare qualitative arguments. On the other hand there are quantitative measurements which is in a numeric form and allows to run statistical analysis. Some of the quantitative measurements are either too specific or specialized for a specific application or there is an important characteristics are missing. Fairness is not an absolute number thus it is difficult to choose between the different methods. This paper includes the max-min fairness algorithm, the Jain's Index, the Alpha Fairness, the ratio between the smallest and the largest values and the Proportional Fairness.

Because fairness is relative it may be true that for some people it is fair if a big player gets a big share of a network and others would prefer if everybody is treated exactly equally all get the same share. With different weights and priorities this problem can be solved in most cases. This paper focuses on objective fairness measurements which take the pathway and desired throughput of each specific player of the network into consideration. Envy-freeness report is an important concept regarding fairness. A player in an envy-free fair network should not prefer the resource allocation of another player. This considers the different needs of each player. Not everybody wants or needs the same. A big player needs a lot of resources while a small player requires much less resources and wants to avoid starvation and a small amount of throughput.

8b.2 The tradeoff between utilization and fairness

Utilization describes the total used capacity of each individual link of a network. Utilization is an important technical trait of any network with limited resources. Efficiency is defined of the actual throughput of a network divided by the maximum possible throughput. By maximizing efficiency usually the profit and the utility of the network increase as well. The resulting trade off is an important concept to consider. The following figures 8b.1 of a network and its efficiency curve related to the fairness illustrate the problem of the tradeoff between efficiency and throughput. α -fairness is a measurement which quantifies the fairness in a network. A higher value means more fairness. It range lies between 1 and 0. Further information will be in the section about α -fairness.

In this graphic 8b.1 it is demonstrated how the α -fairness improves with decreasing network utilization. The utilization drops steadily when flow[a,c] increases its throughput and thus limits flow[b,c]. The Max-Min Fairness Algorithm, which will be introduced later in this paper, would assign each flow from b to c the bandwidth of 1/2. This results in a network utilization of 3/4 because 1/2 of the link capacity form a to b would be unused. If flow[a,c] uses the whole capacity 1, the utilization of the network is at its maximum but the flow[b,c] would starve which results in a bad fairness indicator.

It is important to avoid starvation in networks. Starvation means that a link doesn't get any of the capacity and thus cannot transmit any data. If a link in a network starves the network cannot be fair.

The equal per flow approach is a good solution for simple networks. Bottlenecks are links which limit the throughput from the whole network. The throughput of these bottleneck are key points for fairness as well as efficiency in a network. Each flow gets a proportionally equally share of the bottleneck and stays limited because of this bottleneck. Max-Min Fairness improves the equal per flow approach with allowing the other flows to increase

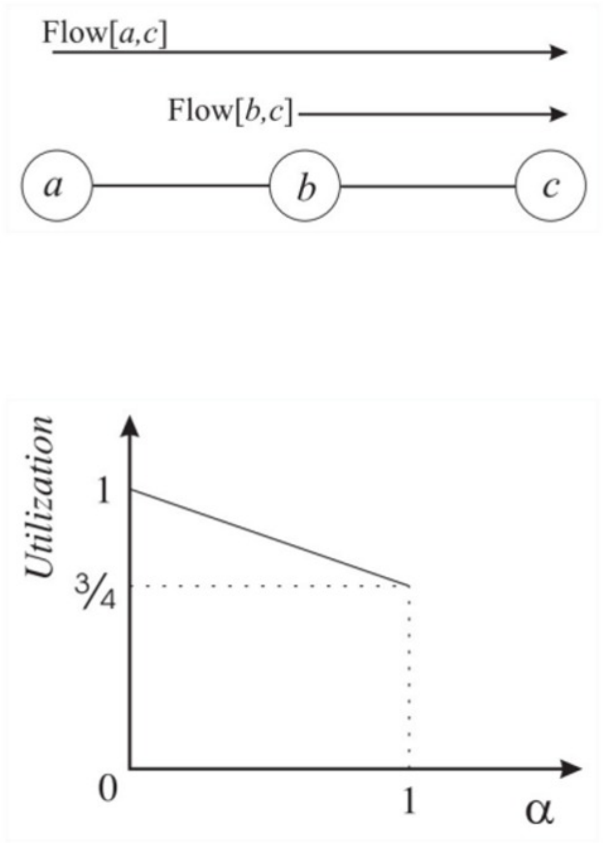


Figure 8b.1: This graphic illustrates the tradeoff between fairness and utilization of the network[7]

their rates until a bottleneck is reached. The following graphics demonstrate the trade off between fairness and efficiency.
Efficiency is 1 in this network but the resulting network is unfair because of starvation of flow B. Applying the Max-Min Fairness algorithm which will be introduced later in this paper will result in the following fair network.

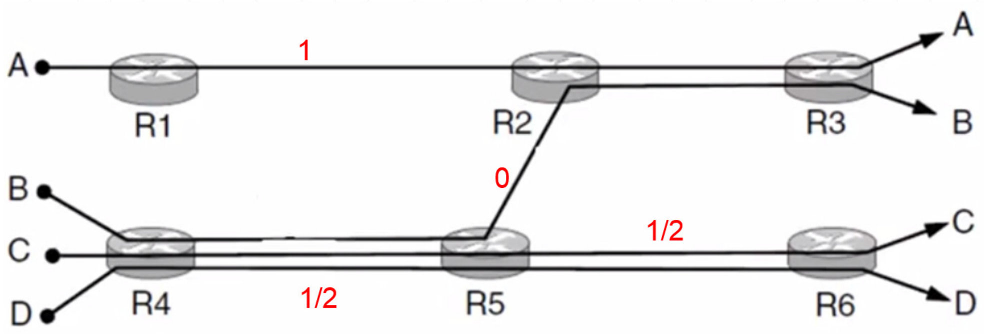


Figure 8b.2: The throughput is maximized in this network. This results in the best efficiency factor of 1.[5]

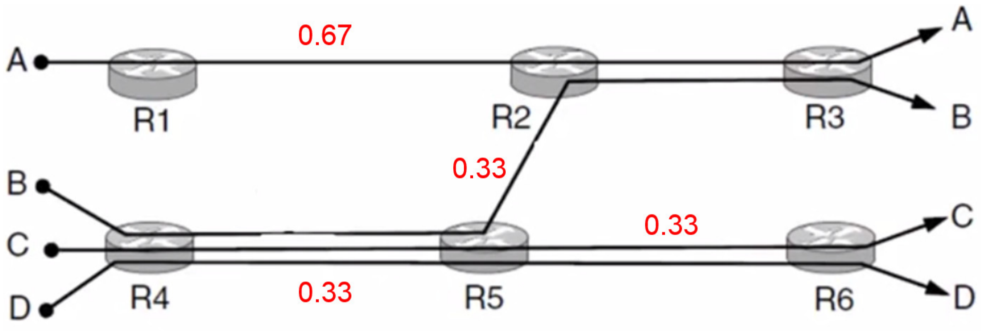


Figure 8b.3: This is an Max-Min Fair Network. The network efficiency is 1.66/2=0.83.[5]

This simple network is fair because all users are treated exactly equally. All flows get the same share and those who can get more without making any other worse get more until the limit of their throughput is reached. Thus this network is envy-free fair.

An interesting question regarding this topic is how much efficiency or utilization can be improved by compromising on fairness until to an acceptable point. Which of the both, efficiency or fairness, is more important depends always on the type of network as well as information transported through the network.

8b.3 Five Axioms of a Fairness measurement

On the search for a good suitable fairness measurement a researcher team from the Princeton University and the Rice University described the following five axioms of a fairenss measurement [4]. Assume that x is a resource allocation vector with n non-negative elements. A fairness measurement $f(x)$ is a mapping from x to a real number. $f : \mathbb{R}^n_+ \to \mathbb{R}$, for all integer $n \geq 1$.

8b.3.1 Axiom of Continuity

The fairness measure $f(x)$ should be continuous on $\mathbb{R}^n_+$ for all integer $n \geq 1$.

8b.3.2 Axiom of Homogenity

The fairness measure $f(x)$ should be a homogeneous function of degree 0:

$$f(x) = f(t \cdot x), \quad \forall\, t > 0$$

Without loss of generality for a single user we take $|f(x_1)| = 1$ for all $x_1 > 0$ which means the fairness measurement should be a constant for $n = 1$.

8b.3.3 Axiom of Asymptotic Saturation

The fairness measure $f(x)$ of equal resource allocations should become independent of the number of users:

$$\lim_{n \to \infty} \frac{f(1_{n+1})}{f(1_n)} = 1$$

8b.3.4 Axiom of Irrelevance of Partition

If the elements of the fairness measure x split into two parts $x = [x^1, x^2]$ It should be possible to compute the fairness index $f(x^1, x^2)$ recursively.

$$f(x^1, x^2) = f\left(w(x^1), w(x^2)\right) \cdot g^{-1}\left(\sum_{i=1}^{2} s_i \cdot g\left(f(x^i)\right)\right)$$

where $w(x^1)$ and $w(x^2)$ describe the sum of the resource vectors x^1 and x^2. $g(y)$ is a continuous and strictly monotonic function that could generate the following function h:

$$h = g^{-1}\left(\sum_{i=1}^{2} s_i \cdot g\left(f(x^i)\right)\right)$$

This has the positive weights satisfying $\sum_i s_i = 1$ such that h qualifies as a *mean* function of $\{f(x^i), \forall i\}$

8b.3.5 Axiom of Monotonicity

For $n = 2$ users the fairness measure $f(\theta, 1-\theta)$ should be monotonically increasing as the absolute difference between the two elements (i.e. $|1 - 2\theta|$) shrinks to zero.

8b.4 Max-Min Fairness

If the following algorithm is applied to a network this network is considered Max-Min fair.
The algorithm is as followed: Each flow starts at rate 0. Then each flow increases its rate equally until there is a new bottleneck in the network. All flows which run through the bottleneck hold their rate fixed. All remaining flows increase their rate until a new bottleneck is reached.
In the resulting resource allocation the increase of the rate of one flow will result in a decrease of the rate of another smaller flow. In the basic form of the algorithm every flow is treated equally. Regardless if it wants to transmit 100 Mbit/s or only 0.5 Mbit/s. Small players get a big share of a Max-Min optimized network and big players get the same small throughput like the small players. There exist also other similar algorithms which take the rate of throughput into account. A Max-Min optimized network may be very inefficient depending on it's architecture. A network is either Max-Min fair or it is not. It's advantage is its simple application and easy understandable algorithm. The algorithm ensures equality between all flows and is therefore fair.

8b.5 Alpha Fairness

α -fairness is one of the most common fairness metrics used. If you maximize the log utility function ($\alpha = 1$) the network is proportionally fair. The maximizer of the α - fair utility function as $\alpha \to \infty$ is max-min fair.
Generally an $\alpha \to \infty$ is more fair than $\alpha = 1$, which is fairer than $\alpha = 0$. However it is unclear what it means to say $\alpha = 3$ is more fair than $\alpha = 2$. α -fair utility functions are continuous and strictly increasing. Its maximization results in Pareto optimal resource allocation.
The α -fairness is widely accepted because it optimizes fairness under desired efficiency constraints. The tradeoff from utilization and α -fairness can be shown with a graph 8b.1.

8b.6 Jain's Index

Jain's index is one of the first developed fairness measurements. It was developed in the 1980thies in the United States by Rajendra Jain. He is the first who mentions desired properties of a good fairness measurement. Theoretically they are the same like some the five axioms of a Fairness Measurement [4] but they are described simpler and less mathematical. According to Rajendra Jain his index fulfills the following properties: population size independence, scale and metric independence, boundedness between 0 and 1 as well as continuity [3].
This rates the fairness of a set of values where there are n users and xi is the amount of transmit of the ith connection. The result ranges from best case 1 to the worst case $1/n$. The best case is reached when all users receive the same allocation. The theoretical approach for this measurement comes from the quote: each users throughput is at least as large as that of all other users which have the same bottleneck.
$\mathcal{J}(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} {x_i}^2}$
There exists a discrimination index. It may be used as the opposite of a fairness index. It is defined as 1 - Jain's index.

8b.7 Proportional Fairness

The proportional fairness is a measurement of fairness which takes different rates of throughput into account. If the Proportional Fairness indicator is at it's maximum 1 the resulting resource allocation of this network takes the higher desired throughput values of big players into consideration and allocates them a higher throughput.
An allocation of rates $\overrightarrow{x}$ is proportionally fair if and only if, for any other feasible allocation $\overrightarrow{y}$, we have:
$\sum_{i=1}^{I} \frac{y_i - x_i}{x_i} \leq 0$
So every small change in the allocation must have a negative affect on the average throughput of the whole network.

8b.8 Ratio between the smallest and biggest throughput

This fairness measurement is very simple. It's application makes more sense in networks with similar players. Starvation gets punished with an zero as solution. If there are distinct players different weighs can be introduced in order to compare them better.

$$R = \frac{Min(X_i)}{Max(X_i)}$$

With X as the throughput of a specific player.

8b.9 Conclusion

There are a lot of different fairness metrics. Each has its advantages and disadvantages. The five axioms of an optimal fairness measurement are a theoretical construction which describe the properties of an optimal fairness measurement. The measurements presented in this paper don't fulfill all the five properties. It requires a very complex mathematical construction in order to create a measurement which fulfills all of these properties. The presented alpha fairness fulfills the most. This is a reason why the Alpha fairness is one of the most common one. It always ranges between 0 and 1 and its value is a continual function. Alpha fairness takes the overall throughput of a network into account. The max min algorithm is easy to apply but it doesn't try to maximize the efficiency of the network. It treats each flow exactly the same. The choice of which fairness metrics should be applied depends on the type of network and the transmitted information. The tradeoff between efficiency and fairness cannot be avoided in complex networks but limited by good network architecture. This decision is important when constructing a network because fairness is important in networks. Normally it is important for a network to keep small players satisfied and avoid starvation because of the economical concept: ”network effect” every player contributes at least a tiny part to the total utility of the whole network. Fairness is important for small players to keep them satisfied and to avoid that they leave the network.

Bibliography

[1] Cambridge Dictionary; http://dictionary.cambridge.org/dictionary/british/fairness

[2] S. Floyd: Metrics for the Evaluation of Congestion Control Mechanisms; http://www.ietf.org/rfc/rfc5166.txt, 3.2008

[3] Rajendra K.Jain, Dah-Ming W.Chiu, Williamm R.Hawe: A Quantitative Measure of Fairness and Discrimination for Ressource Allocation in Shared Computer Systems;http://www.cs.wustl.edu/~jain/papers/ftp/fairness.pdf, 26.09.1984

[4] Tian Lan, David Kao, Mung Chiang, Ashutosh Sabharwal: An Axiomatic Theory of Fairness in Network Resource Allocation; 7.10.2009

[5] David Wetherall, Arvind Krishnamurthy, John Zahorjan from University of Washingon: Computer Networks 7-2: Fairness of Allocations; https://www.youtube.com/watch?v=w33pJ8XEGaU, 31.08.2013

[6] Zukerman et al.: Efficiency-fairness tradeoff in telecommunications networks.

[7] Moshe Zukerman, Liansheng Tan, Hanwu Wang, and Iradj Ouveysi: Efficiency-Fairness Tradeoff in Telecommunications Networks; IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 7, JULY 2005

Chapter 9

Internet Service Providers: Peering and Charging

Markus Cadonau

In this chapter, we examine how the Internet can be modeled as network of autonomous systems, particularly Internet service providers, and how its hierarchy has evolved since the introduction of the World Wide Web. Out of the two classic relationships among Internet service providers, transit and peering, we mainly focus on peering by highlighting its advantages and disadvantages, describing a typical peering process, discussing current challenges in peering. We find that overall revenue of autonomous systems, including content providers and content delivery networks, can be optimized through cooperation.

Contents

9.1 Introduction

Ever since the introduction of the World Wide Web, the Internet has been changing in an accelerated fashion in terms of its hierarchy, network traffic and commercialization. Started by deregulation, internet service providers (ISPs) have changed how they connect to each other, new significant autonomous systems (ASs) have emerged, the Web evolved from static Web sites to so called Web 2.0 services and applications with rich media content. Further, new services—such as voice over IP (VoIP), peer to peer (P2P) file sharing virtual networks etc.—have been established. All these changes are affecting how Internet service providers charge their customers and each other for different types of network access.

In Section 9.2 we describe the two basic inter-network connection types, peering and transit relationships. Section 9.3 shows the evolvement of the Internet hierarchy over time. Section 9.4 discusses why peering spreads, but also what its disadvantages are. In Section 9.5 we describe a typical peering process. Current challenges in peering are documented in Section 9.6.

9.2 Internet Service Provider Relationships

Traditionally, there have been two main inter-network relationships among autonomous systems, e.g. Internet service providers: namely, peering and transit relationships. An autonomous system (AS) is seen as multiple IP-connected devices under control of a single administrative entity with a clear routing policy outwards [9]. Examples of autonomous systems are Internet service providers, content providers (CPs), transit providers, academic research networks etc. Measured by autonomous system numbers (ASN), there are 35000 ASes as of 2012 [10]. Both relationship types, peering and transit, normally make use of the Border Gateway Protocol (BGP) for routing announcement exchanges [12]. However, the relationships differ in what routing information is exchanged on each side of the partnership, as well as in the business aspect. Technically, the connections between networks do not necessarily differ between a peering and a transit relationship. Both partners can either exchange traffic through one or multiple—usually spatially separated—links. Peering is detailed in Section 9.2.1 and the transit relationship in Section 9.2.2.

9.2.1 Peering Relationship

While technically being able to advertise more via Border Gateway Protocol, a peering partner only announces the reachability of its own end-hosts to the contract partner:

“Peering is the business relationship whereby ISPs reciprocally provide each other connectivity to each others’ transit customers.” [12]

Therefore, peering is a non-transitive [10], bilateral relationship. In a classical peering agreement, no money, only network access, is exchanged between the two partners. If peering were the only option to connect to other networks, an Internet service provider would have to peer with every other provider in order to guarantee full network access to its customers.

9.2.2 Transit Relationship

While a traditional peering relationship is ideally fairly symmetrical in terms of traffic flows and extended network reachability, a transit business relationship consists of an asymmetrical exchange, whereby one partner provides the other with full access to its routing table, usually by selling access to it [12].

9.3 Evolution of the Internet Hierarchy

Early on, the Internet was regulated. As is common in more authoritative systems, a hierarchical structure was built. Once deregulated, it became more interwoven; Especially the ISPs formally at the top of the hierarchy were circumvented by their transit customers through peering. Nowadays, the Internet has transformed into a peering mesh and become very flat. Section 9.3.1 lays out the original structure of the Internet, Section 9.3.2 discusses how Donut Peering formed and Section 9.3.3 shows how the Internet presents itself today.

9.3.1 Tiered Internet Architecture

Until the mid-1990s the architecture of the Internet was quite strictly hierarchical. Three tiers are distinguished. Members of Tier 1 are defined as having access to the global Internet routing table without having to pay for it, i.e. a Tier 1 internet service provider does not purchase transit from anyone [12]. On the other end of the spectrum, smaller in size and only providing Internet access to endusers, are Tier 3 Internet service providers, typically only purchasing transit [10]. Tier 2 Internet service providers, the middle tier, are selling transit access to Tier 3 providers while also purchasing transit connectivity to the global Internet via Tier 1 ISPs.

9.3.2 Donut Peering

The term “Donut Peering” is used to describe the fact that by the end of the second millennium former Tier 2 Internet service providers have been peering with each other, essentially forming a mesh network around the Tier 1 ISPs [15]. Different reasons were cause for this development (see Section 9.4). The original Tier 1 ISPs were trying to conserve their oligopoly, shutting out the entire middle tier from peering with them. The benefits of peering among themselves, essentially becoming a new Tier 1 ring, outweighed the costs of forming the new network connections.

9.3.3 Emergence of Content Providers and Content Delivery Networks

By today, the internet hierarchy has become increasingly flat. Content plays an important role. So much that certain content providers and content delivery networks—aggregating, distributing and delivering content from multiple providers—themselves are considerable entities in the entire Internet architecture and infrastructure (see Figure 9.1). Hence, their economical influence also increased. In 2009, already 10 percent of inter-domain traffic on peering and transit links involved content delivery networks [10]. Whereas initially peering was mostly initiated for cost reasons, it is now usually done for performance reasons. Traffic follows shorter routing paths, transit providers are left out [6].

9.3.4 Content Based Peering

In exchange for free peering, content providers sometimes have their own content servers within Internet service providers’ infrastructure, or at Internet exchange points (IXPs, see Section 9.5.3) respectively, to be closer to customers [10].

It is suggested that in the future of the Internet, peering strategies based on content become more relevant and could lead to efficiency gains. Each peering partner would install caches at strategic routing points and new distributed networking protocols would manage appropriate replication of content among the partnering networks’ caches [16].

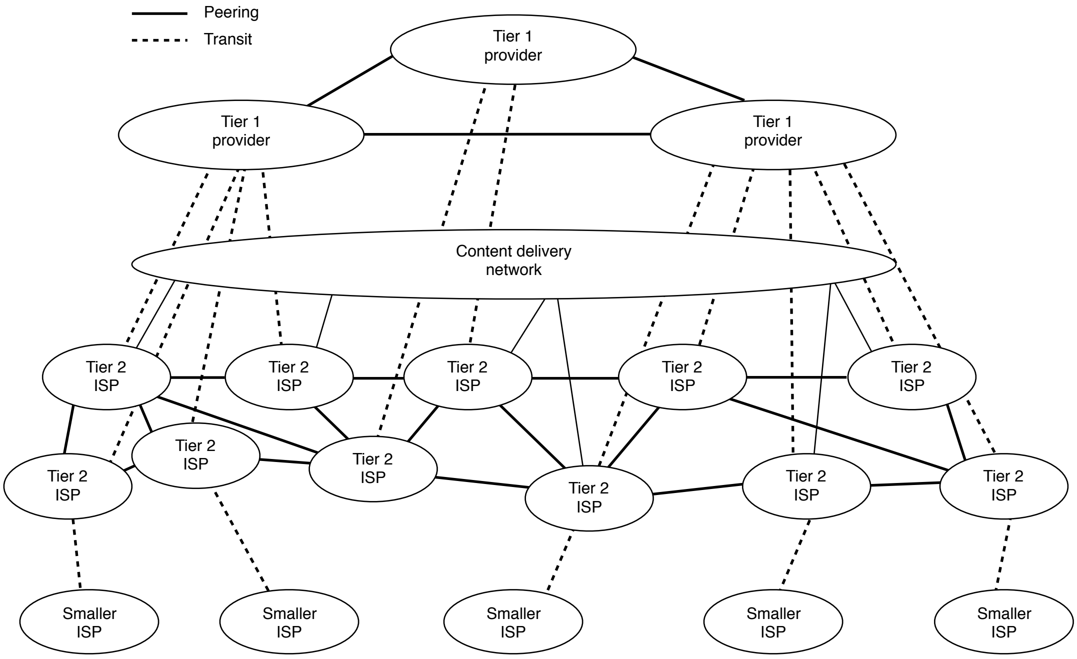


Figure 9.1: Model of Internet hierarchy with Donut Peering and content delivery network [2]

ISPs would play a more active and cooperative role in content management to ensure quality of service (QoS). A collaborative caching strategy among ISPs could replace the traffic-centric model of networking. In fact, with the trend to multilateral peering [15], i.e. at IXPs, this becomes more feasible.

9.4 Advantages and Disadvantages of Peering

The main reasons a peering agreement is established between two Internet service providers are to lower transit costs and to lower latency. However, over the course of time, those reasons have not always played an equally important role. First, saving the costs of transit traffic by routing the traffic directly to other ISPs—with which peering was established—saved providers considerable amounts of money. A somewhat faster response time was secondary. Now, saving 100ms is significant. In fact, end-customers less likely purchase and return to an online retailer with bad Web site performance [10]. Transit costs per data unit, on the other hand, are a mere fraction of what they used to be; at one point IP transit cost nearly $1000/Mb/s per month, billed near peak consumption rate. It is less than $0.42/Mb/s per month now [10]. This change can at least partially be attributed to the competition by peering [10]. Albeit, with inter-domain traffic annual growth rates of roughly 50 percent, the transit costs are still a factor [11].

Providers which charge customers by data unit have a third, less relevant yet supporting argument for peering. Better connectivity leads to more consumed data. Thus, more revenue can be generated from usage-based traffic billing of customers.

There are also some downsides to peering, i.e. reasons for Internet service providers not to peer or stop peering with one another. Especially Tier 1 ISPs initially had strong reservations about peering with Tier 2 ISPs. By peering with a current transit customer a provider loses that transit revenue, or potential future transit revenue in other cases. Also, from a business perspective, an ISP might not want to help a—potentially smaller—competitor by acknowledging it as an equal peering partner. If a benefit is not mutual, or traffic and investment asymmetries are overly disproportional, potential peering partners are discouraged from entering a peering agreement. Such a traffic imbalance, whereby one provider dumps a few times more data on its peering partner's network than vice versa, can also manifest itself at a later stage of the relationship. Not least, peering can consume many resources; New links potentially have to be set up entirely, requiring hardware and engineering. Often, the costs for it are sunken and a free-rider problem is created [3]. That is, once the investment in new infrastructure is made, others have little incentive to pay their share for its amortization and are only willing to pay the relatively little additional expenses for the added traffic. As we will see in Section 9.5, there have often been no fully negotiated and detailed service level agreements (SLAs) regarding peering. From a business perspective, this increases risks. An Internet service provider has no guarantees about a peering link, nor any legal measurements for compensation in case a peering partner decides to change the peering characteristics one-sidedly.

9.5 Peering Process

A typical peering process can be described as consisting of three phases [12]: Phase 1 concerns identifying potential peers (see Section 9.5.1), Phase 2 deals with contacting and qualifying peering candidates (see Section 9.5.2), and Phase 3 is about concrete implementation of a peering agreement (see Section 9.5.3). Overall, peering is often seen as art rather than science [7]. Therefore, the following process is not to be understood as prescriptive but as a general observation of how peering often is established.

9.5.1 Identification of Potential Peers

Identifying potential peering partners can be done in different ways. From an economical perspective, a traffic analysis of an Internet service provider's outgoing and ingoing data makes sense. The networks with which traffic exchange is most significant are predestined peering candidates, given that data is so far exchanged via transit partner. Depending on the objectives, this traffic quantity approach promises the most reductions in traffic expenses and the most benefit in network traffic performance. Not uncommon is a pre-selection of potential partners based on intuition and existing or previous relationships [12]. Large ISPs tend to have relatively fixed peering conditions. Yet, they commonly only expose them under non-disclosure agreements (NDAs). All in all, the determination of one's own potential peering benefits is quite feasible, but it is very difficult to evaluate how a partner would profit [7]. Sometimes—especially if technological hurdles are not too big—a peering trial is established to gather concrete data.

9.5.2 Contact and Qualification

During the second phase of a peering process, initial peering negotiation is undertaken. An Internet service provider first needs to establish contact with a potential peer. Depending on the role of the initiating and contacting members of the ISP, the negotiation is started on a less or more formal level. Typically, there is a person specifically tasked with peering and traffic engineering issues [12]. During the main negotiating phase, traffic statistics, peering policies, case-specific peering arguments are—among other details—shared. At the end of this contact and qualification phase is the decision whether there is enough motivation for both parties to continue the peering discussions. If both sides want to continue, they proceed to discussing peering methodology (see Section 9.5.3).

9.5.3 Implementation Discussions

Existing technical infrastructure may already play a role in general peering negotiations. After all, the question that ISPs essentially have to answer is whether to invest in peering with one another and in necessary resources for peering. But once the general decision to peer is made, resource requirements estimates have to be refined and thus specific implementation questions have to be answered and agreed upon. The primary goal is to institute interconnection points, the secondary goal is to steer optimal traffic behavior.

There are generally two interconnection methods: direct-circuit and exchange-based interconnections. One does not exclude the other. ISPs expecting a small number of regional interconnects and not very high bandwidth requirements prefer direct-circuit connections, given that there is an existing presence in the area. Otherwise, peering would not make much sense in this case. Key issues are link locations and cost distribution. Typically, costs are split with the regional peering partner [12].

ISPs anticipating high bandwidth requirements in a region or interconnections with many peers tend to choose exchange-based links over direct-circuit ones. Either way, the goals are to peer as quickly as possible, reduce the costs while maximizing benefits.

There are additional criteria for an ISP to be considered when—in principle—selecting an exchange-based approach. A key question is if there are existing attractive Internet exchange points (IXPs) in the area or if a new one has to be built. IXPs require a critical mass of participants for them to be profitable and beneficial [12]. In Europe, where the ISPs are diverse, IXPs play an important role for the Internet and are very similar to each other in terms of services offered; the number of peering links within such an IXP is so big that the IXP can be seen as a microcosm of the Internet [1].

Influence of Topology Aspect on Peering Decision

For regional ISPs available transit and peering connections can be limited or not available. Besides only locally using Internet transit services, which are 10 to 100 times more expensive compared to services in major cities [13], such ISPs face the decision to rather finance a connection to a medium or major hub where more peering and transit options are available. Depending on the amount of carried traffic as well as the fixed and variable costs for the line and the remotely purchased services, it could economically be beneficial to do so and thus contribute to a flatter Internet. More direct connections to content providers are especially important for customer satisfaction.

9.6 Peering Challenges

There are some challenges with traditional peering agreements in combination with today's Internet characteristics. Different solutions to overcome certain issues exist or are proposed. In Section 9.6.1 we address the hot-potato routing problem. Section 9.6.2 discusses asymmetry in costs and network traffic. We consider data congestion in Section 9.6.3, and in connection the influence of different Internet players on each other in Section 9.6.4.

9.6.1 Hot-Potato Routing

The term "hot-potato routing" refers to the fact that data traffic destined to another network is usually transferred to that network at the closest intersection, handed off at the first opportunity. If two spatially large Internet service providers peer with each other, this can have a negative impact on one network if it is mostly done one-sided. One provider is then basically dumping data on the other's infrastructure which has to carry it over large distances to its destination. For this reason, ISPs—especially such that cover wide areas—specify in service level agreements that data traffic is only to be accepted at interconnect locations closest to the destination of the traffic; or more generally, that traffic has to be balanced and multiple peering links at different hubs are a requirement for an agreement [13].

9.6.2 Asymmetry

Traffic as well as infrastructure cost imbalance in general can be a cause for differences and disputes in traditional peering agreements, as they are built on the assumption that both Internet service providers contribute more or less equally to the partnership. In this case, ISPs can also partially protect themselves by setting ratios in the bilateral peering agreements limiting a potential traffic asymmetry. Generally, ISPs set the ratio limit at 5:2 between ingoing and outgoing traffic [10]. Especially in nowadays content-heavy Internet asymmetrical traffic is the norm [8], whereas one ISP with mostly content providers as customers directs big data flows to another ISP with mostly content consumers as customers. Those customers normally only pay the content providers directly for the content; their ISPs generate the same revenue from them due to flat-rates but have to carry increasingly more data streams.

9.6.3 Congestion

It is always a possibility that short-term congestion occurs at interconnections of autonomous systems as part of the Internet. Nonetheless, it is argued that chronic congestion is not a result of technical limitations but business differences; a lack of addressing the issue [3]. Often, it is not clear who is responsible to take care of the issue, or by whom resources needed to fix it are paid for. Further, as was the case in the Netflix (CP) vs. Comcast (ISP) congestion dispute, a relatively simple shift of traffic to other links can largely solve the technical aspect of the problem overnight [3]. In general, congestion can be reduced by increasing capacity of a problematic link or reducing load on it. Albeit, it generally is more efficient from a global network point of view if different entities are cooperatively working together to ensure adequate quality of service and quality of experience for customers [3]. It is the way most profits could be generated overall.

9.6.4 Peering Politics

It used to be that Tier 1 Internet service providers were rather powerful because they controlled access to the global Internet. Today, as the entire structure of the network has become flatter and most ISPs are more interconnected, there are still larger, more influential and smaller, less influential providers. But additionally, large content providers and content delivery networks have emerged. While less connected and thus less powerful ISPs have traditionally always paid larger ones for transit, similarly sized ISPs have peered. Content providers, which are structured differently than ISPs, have complicated the matter in the meantime. Sometimes, it is objectively unclear who should pay whom for better network or content access. Content providers can be on both sides of the paying relationship [7]. Intermediary content delivery networks have not made it easier. And end-customers, depending on their ability to switch to different ISPs, can also have a big influence. This can result in power plays among the involved parties and has lead to considerations of regulatory involvement [14].

In order to maintain profitability—which can be done without violating “network neutrality” [5]—it is understood that ISPs opt to have some income through transit services with selective peering and through paid peering, i.e. receiving compensation from content providers [4, 10] in addition to improving customer experience due to better content access. To limit the power of ISPs, especially monopolistic or duopolistic ISPs, and protect customers, limited regulation enforcing global Internet reachability for customers of ISPs is encouraged [4].

9.7 Conclusion

We conclude that peering among autonomous systems plays an important role in the evolution of the Internet by making it faster and more affordable. Not only to improve quality of service and thus customer satisfaction but also to optimize revenue, all Internet service providers have incentives to selectively peer. Since classic peering would eventually lead to stagnation, especially when uneven partners are involved, paid peering has emerged. It is shown that cooperation between Internet entities would overall be optimal in terms of revenue and connectivity. However, it is unclear how such cooperation can be achieved, or which regulatory measurements would encourage it.

Bibliography

[1] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. Anatomy of a Large European IXP. In *Proceedings of the ACM conference on applications, technologies, architectures, and protocols for computer communication (SIGCOMM 2012)*, pages 163–174. ACM, 2012.

[2] Kimberly Claffy and David Clark. Platform Models for Sustainable Internet Regulation. *Journal of Information Policy*, 4:463–488, 2014.

[3] David D Clark, Steven Bauer, William Lehr, Kimberly Claffy, Amogh D Dhamdhere, Bradley Huffaker, and Matthew Luckie. Measurement and Analysis of Internet Interconnection and Congestion. In *Proceedings of the Research Conference on Communication, Information and Internet Policy (TPRC 2014)*, 2014.

[4] Pierre Coucheney, Patrick Maillé, and Bruno Tuffin. Network Neutrality Debate and ISP Inter-Relations: Traffic Exchange, Revenue Sharing, and Disconnection Threat. Research Report, 2012.

[5] Amogh Dhamdhere and Constantine Dovrolis. Can ISPs be Profitable Without Violating "Network Neutrality"? In *Proceedings of the 3rd International Workshop on Economics of Networked Systems*, pages 13–18. ACM, 2008.

[6] Amogh Dhamdhere and Constantine Dovrolis. The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh. In *Proceedings of the 6th International Conference on emerging Networking EXperiments and Technologies (CoNEXT 2010)*, page 21. ACM, 2010.

[7] Amogh Dhamdhere, Constantine Dovrolis, and Pierre Francois. A Value-based Framework for Internet Peering Agreements. In *Proceedings of the 22nd International Teletraffic Congress (ITC 2010)*, pages 1–8. IEEE, 2010.

[8] Wenjia Fang and Larry Peterson. Inter-AS traffic Traffic Patterns and Their Implications. In *Proceedings of the Global Telecommunications Conference (GLOBECOM 1999)*, volume 3, pages 1859–1868. IEEE, 1999.

[9] John Hawkinson and Tony Bates. Guidelines for Creation, Selection, and Registration of an Autonomous System (AS). RFC 1930, RFC Editor, 1996.

[10] Bill Krogfoss, Marcus Weldon, and Lev Sofman. Internet Architecture Evolution and the Complex Economies of Content Peering. *Bell Labs Technical Journal*, 17(1):163–184, 2012.

[11] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. Internet Inter-Domain Traffic. In *Proceedings of the ACM conference on applications, technologies, architectures, and protocols for computer communication (SIGCOMM 2010)*, pages 75–86. ACM, 2010.

[12] William B Norton. Internet Service Providers and Peering. In *Proceedings of the 21st North American Network Operators' Group Meeting*, volume 19, pages 1–17, 2001.

[13] Brough Turner. Impact of Internet Peering on Network Architectures and Economics. In *Proceedings of the Optical Fiber Communication Conference (OFC 2014)*. Optical Society of America, 2014.

[14] Lukas Wiewiorra and Sascha Schweitzer. Paid Peering and Content Delivery. In *Proceedings of the 35th International Conference on Information Systems (ICIS 2014)*. Association for Information Systems, 2014.

[15] Bill Woodcock and Vijay Adhikari. Survey of Characteristics of Internet Carrier Interconnection Agreements. Research Paper, Packet Clearing House, May 2011.

[16] Jia Zhao, Jianfeng Guan, Changqiao Xu, Wei Su, and Hongke Zhang. Peering Strategic Game Models for Interdependent ISPs in Content Centric Internet. *The Scientific World Journal*, 2013, 2013.