Making historical connections: Building Application Layer Traffic Optimization (ALTO) network and cost maps from public broadband data

Vijay K. Gurbani
Bell Laboratories, Alcatel-Lucent
Email: vijay.gurbani@alcatel-lucent.com

David Goergen, Radu State, Thomas Engel
University of Luxembourg
Email: {david.goergen,radu.state,thomas.engel}@uni.lu

***Abstract*—The Application Layer Traffic Optimization (ALTO) protocol allows network service providers to make available a pair of maps to applications such that the applications can intelligently (compared to randomly) connect to a desired resource. The network map aggregates the service provider network into provider defined identifiers (PID) and the cost map provides a pair-wise link cost between each PID. Clearly, a service provider has an authoritative view of its network and is able to provide an ALTO server that distributes such maps. However, ALTO also envisions third-parties as being able to provide such maps. In this paper, we demonstrate how a third-party ALTO server can provide maps by mining public information. Specifically, we build our maps from the United States Federal Communications Commission public broadband data set, which contains an expressive (multi-tier wireline broadband measurements) and rich (measurements for specific application uses) dataset. In all, we examined over 1 billion records spread over 90 GBytes as part of our analysis. We borrow concepts from financial engineering and social network analysis to show how network topology and cost maps can be created, and furthermore, how peer-to-peer systems can insulate themselves from going dark by choosing supernodes effectively from mining historical data.**

I. Introduction and Problem Statement

The Application Layer Traffic Optimization Protocol [1] is a IETF-standardized protocol that constitutes a service for disseminating network information (e.g., basic network location structure and preference of network paths) to applications for improving performance. The basic information in ALTO is provided in the form of two maps: a network topology map and a cost map. These maps allows the network provider to provide a succinct, yet abstract view, of the network to allow applications express preferences when connecting to the desired resource. For example, a node in a peer-to-peer network can choose other peers that belong to the same Internet service provider (ISP) to minimize latency; similarly, a Content Distribution Network (CDN) router can use the maps to rendezvous a client with the nearest surrogate. Figure 1 shows an example of the map service. In the example, the network map partitions all endpoints into three provider-defined identifiers (PID); the *Default* PID is essentially a catch-all for all hosts not belonging to the ISP. The cost map provides directed costs between each pair of PIDs. Using these maps, a peer in PID-2 will contact peers in PID-1 first before reaching out to other peers on the Internet, forearmed with the knowledge that it costs less (2 units) to access peers in PID-1 than it does to access peers in the default PID (6 units). The costs on the link represents generic costs and are computed by the ISP according to its preferences and policies. These two abstractions — the topology and the cost maps — provided by the ISP allow peers (or clients) on the network to affect the choice of peers (or servers) they will connect with.

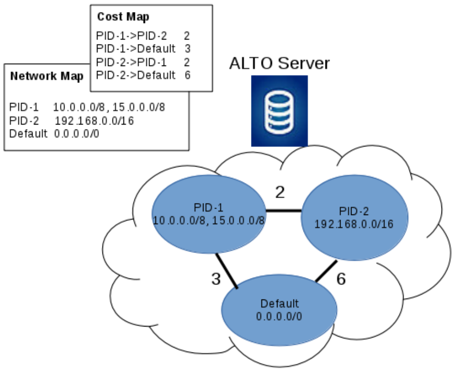


Figure 1: An ALTO server

In principle, the ALTO protocol allows for parties not associated with the ISP (third parties) to create topology and cost maps. However, in ALTO-like deployments prevalent today, it is the ISP that creates these maps and make them available on an ALTO server. After all, the ISP is the authoritative entity that knows the dynamics and reach of its network as well as the settlement price associated with peering and transit links. To the best of our knowledge, there isn't any published literature that allows third parties to create network topology and cost maps tailored to a specific application running in a network.

Our primary contribution is to demonstrate how third parties can create network and topology maps for ALTO from public sources of information. More specifically, we use the United States Federal Communications Commission (FCC) public database from the Measuring Broadband America (MBA) program. This is an ongoing, rigorous, nationwide study of residential broadband performance in the United States. We mine the public data to assign each subscriber to an ISP (this relationship is not captured in the data distributed by the FCC). The ISP clusters serve as the PIDs in our work. In fact, unlike an ISP that possesses a localized view, our work operates at a layer above, essentially considering relationships between a cluster of ISPs.

The rest of the paper is structured as follows. A brief

ISBN 978-3-901882-67-8, 10th CNSM and Workshop ©2014 IFIP

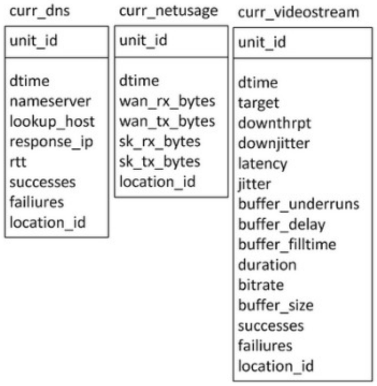
introduction to the FCC MBA program and the dataset are in Section II. The nodes participating in the FCC study are anonymized; consequently, the first task was to identify a stable set of nodes and their corresponding ISPs (Section III). This set constitutes the network map, each ISP corresponding to a PID. At this stage, each component in the network map (or graph) is disconnected. Section IV uses financial engineering techniques to derive cost maps and connect the graph depending on the specific service under consideration. Given such networks and cost maps, a peer (client) can now make better decisions on which other peers (servers) it will seek out. This decision can be optimized for a specific service, e.g., the costs involved in a low-latency service will be different than those involved in a high upload bandwidth service. In Section V, we see how historical data allows accurate identification of stable nodes; such nodes could be candidates for being promoted to supernodes in a P2P network. We describe related work in Section VI followed by a conclusion.

II. Describing the data: FCC MBA dataset

To model or reason on network dynamics, one needs to have measures derived from the actual installation over a reasonable period of time. Obtaining such data is not easy because the network operators typically do not make such data public due to privacy and business reasons. Furthermore distributing sensors evenly to measure performance and capabilities of a system need careful planning as these should not interfere with the daily traffic of the system biasing the measures. In an ongoing study that started in 2010, the U.S. Federal Communication Commission (FCC) has been measuring the nationwide performance of broadband service in the United States under a program called Measuring Broadband America (MBA). The data collected is available publicly after removing all subscriber identifying information (name, address, customer tier, IP address, etc.). The subscriber is represented only by an invariant opaque unit identification number, a *unit_id*.

The MBA program collects data with the support of national broadband service providers to measure the actual capabilities of their national wired network. MBA is an opt-in program where volunteers agree to host a whitebox in their homes. The whitebox plugs into the home network and runs a series of network performance tests hourly to gather the data. The data is then collected, analysed and published as an annual report by the FCC. More information about the MBA program is available in the FCC MBA Technical Appendix [2].

There are 14 tests run once every hour. Data from each of the tests is sent to measurement servers where it is tabulated, anonymized and made available publicly. Our work uses the data corpus from 2012 [1]. The data set consists of 12 tables, of which the following three tables were used in our work (see Figure 2): (1) Table **curr_dns** uses 9 predefined and well-known websites to measure the DNS resolution time for each of them. (2) Table **curr_netusage** captures the total amount of data received and transmitted from the subscriber and the amount of bytes transmitted and received as a result of active performance measurements. (3) Table **curr_videostream** simulates the subscriber viewing an online streaming video. A 3-second playout buffer was configured in the client, and data download at the maximum rate to ensure this buffer never drained. Each table contains *unit_id* as an invariant, allowing us to cross-index the same subscriber across tables. Each table has a *location_id* field as well, but the contents are random digits that cannot be used to derive the physical location of the subscribers.


curr_dns
unit_id
dtime
nameserver
lookup_host
response_ip
rtt
successes
failiures
location_id

curr_netusage
unit_id
dtime
wan_rx_bytes
wan_tx_bytes
sk_rx_bytes
sk_tx_bytes
location_id

curr_videostream
unit_id
dtime
target
downthrpt
downjitter
latency
jitter
buffer_underruns
buffer_delay
buffer_filltime
duration
bitrate
buffer_size
successes
failiures
location_id

Figure 2: MBA tables for analysis

III. Geo-locating unit_ids for a network map

Table I shows the total volume of data analyzed from the 2012 dataset. The first challenge was to arrive at computing resources comparable in scale to the dataset consisting of over 1 billion records spread across gigabyte-sized files. We used Map-Reduce on a 4-node cluster running Apache Hadoop (version 2.0.0-cdh 4.30). Each node is configured as a hexacore Xeon CPU with 2.4 GHz and 120 GB RAM.

	Data (GBytes)	Records (Millions)
curr_dns	75.9	984.3
curr_netusage	2.9	48
curr_videostream	11.8	96
Total	90.6	1128.3

Table I: Volume of data analyzed

A second, more pressing challenge, was to identify the geographic location of the *unit_id*s. In order to derive a topology map and impose costs on the links, it is important to know the physical locations of the *unit_id*s that contributed the measurements. However, in the MBA dataset, the population is anonymized and the individual subscriber reporting the measurement data is simply referred to by a random number. Therefore, an important task was to use the information from the public dataset to reveal a coarse location of the subscriber and assign the subscriber to a specific ISP. To geo-locate the units, we simply note that broadband subscriber devices are likely to be configured using DHCP, which imparts an IP address and DNS servers to the subscriber device. These DNS name servers are located in close physical proximity of the subscriber device. The FCC technical appendix [2] states that the DNS resolution tests were targeted directly at the ISP's recursive resolvers; therefore, a reasonable approximation of a subscribers location is the geographic location of the DNS name server serving the subscriber. We use this very heuristic to geo-locate a subscriber. The time needed for extracting DNS resolver's IP addresses associated with individual *unit_id*s

[1] The anonymized raw data for each month of 2012 is available at http://www.fcc.gov/measuring-broadband-america/2012/raw-data-2012/

was around 30 minutes for the data set. To identify the geographic location of the *unit_ids*, we used an online geo-location service, although we were limited by the number of queries. It took us two days to analyze the entire dataset. This methodology does not identify the specific location of a subscriber, who still remains anonymous. It simply locates the subscriber in a region and associates an ISP with the specific subscriber. This level of granularity suffices for our work.

Our first and very simple filter consisted of obtaining a mapping from a *unit_id* (representing a subscriber) to one or more DNS name servers that the *unit_id* is sending DNS requests to. It turned out that while this was a necessary condition for advancing, it was not a sufficient one. The raw data would need to be further processed to reduce inconsistencies and remove outliers. A number of interesting artifacts were uncovered during further processing of the data. These artifacts, enumerated below, informed the selection of the *unit_ids* for further analysis.

(1) A handful of *unit_ids* were geo-located outside the contiguous United States, such as in Ukraine, Poland or the U.K. We theorize that the subscribers corresponding to the *unit_ids* had simply configured their devices to use alternate DNS servers. These records did not contribute to our analysis.

(2) We also observed a number of non-ISP DNS resolvers, especially Google's 8.8.8.8, 8.8.4.4; and OpenDNS's 208.67.222.222, 208.67.220.220. These 4 public servers are geo-located in California. We removed these records to ensure that California was not oversampled.

(3) We noticed that a large number of *unit_ids* were geo-located in Potwin, Kansas ($37^{\circ}N97^{\circ}W$). Intrigued as to why there appeared to be a large population of Internet users being located in a small rural community (pop. 441), we investigated further. It appears that Potwin, Kansas is the geographical center of the United States. If the IP geo-location service is unable to pinpoint the location of an IP address, it returns the coordinates corresponding to the geographic center of the country. Other researchers have dubbed this the "Potwin Effect" [3]. We excised all records that showed the impact of such aggregation points so they don't skew our results.

Subsequent filters extracted the stable *unit_ids* from our dataset. In order to determine which *unit_ids* are stable, i.e., remain constant with respect to their geographic location over the observation period from January to December 2012, we extracted for each *unit_id* the IP address of each DNS name server it consulted. This was repeated for each month of the observation period. The resulting sets were cleaned up of private IP addresses and other artifacts discussed above.

To determine the stability of each *unit_id* we proceeded to sum up the occurrences of IP addresses over the whole observation period separated in monthly files. If the IP address of a DNS server occurred 12 times this meant that the *unit_id* always accessed the same DNS server and therefore remained stable over the observation period. The 2,142 stable *unit_ids* thus obtained were used for further analysis. We note that about 2,000 *unit_ids* constitute a good sample from a population of 7,968 subscribers in the July 2012 MBA report [2]. Assuming a 99% confidence level and ±3 point margin of error, we will require a sample of 1494 *unit_ids*. With our more than adequate stable set of 2,142 *unit_ids* , we continued with further analysis on the dataset to create the full topology and cost maps. The stable *unit_ids* superimposed on the map of the United States are illustrated in Figure 3. The size of the radius of a circle represents the concentration of *unit_ids*.

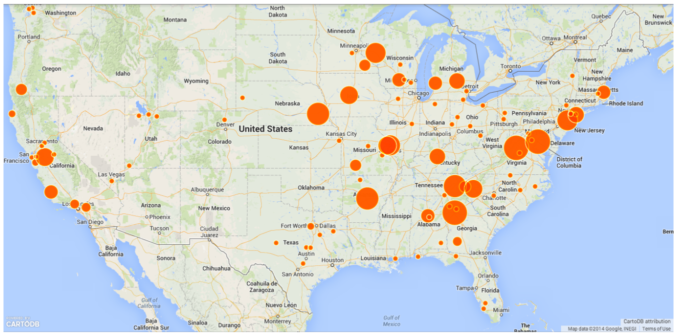


Figure 3: Distribution of stable *unit_ids*

ISP	*unit_ids*
Comcast	726
Charter Comm.	447
Cox Comm.	386
Verizon DSL (MCI)	243
Windstream Comm.	151
Mediacom Comm.	49
Cablevision	37
Time Warner	31
Century Link (Embarq)	19
Total	**2,089**

Table II: Top 9 ISPs

A side effect of geo-locating susbcribers using the DNS resolver is that we were also able to associate each of the stable *unit_ids* with an ISP. Because the DNS resolution test was targeted directly against the ISP's recursive resolvers, it was trivial, therefore, to figure out which ISP served a particular *unit_id*. Table II shows those ISPs having at least 19 stable *unit_ids*. The remaining 53 *unit_ids* are distributed over 32 ISPs, but because each ISP has a small number of *unit_ids*, they are not listed in the table. For the analysis in the rest of this paper, we only consider the top 9 national ISPs listed in Table II.

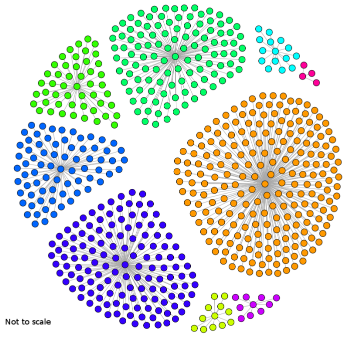


Figure 4: Top 9 ISPs as disconnected components in a graph

The collection of ISPs serves as our network map. In ALTO, a network map is provides a full set of network location groupings and endpoints contained within each grouping. The ISPs serve as our location grouping, with each ISP corresponding to a PID. The stable hosts in an ISP serve as endpoints contained within each grouping. Figure 4 shows the top 9 ISPs as isolated clusters (or disconnected components) in a network. (Only 30% of the hosts in each ISP are shown in the figure to aid in visualization.) Note that at this point, no links exists between the components; i.e., we have only created a network map, a cost map that connects the components still needs to be created.

IV. Building cost maps from FCC data

We build cost maps for each ISP using two important features that are relevant for end user experience: the upload throughput rate (from *curr_netusage* table) and and latency (from *curr_videostream* table). We compute a monthly average of these rates for each ISP and define the cost matrix in terms of Sharpe ratio [4], a common financial engineering metric. This ratio measures to which extent a given asset is an adequate tradeoff between the risk assumed by an investor compared to the expected return. The ratio is a natural extension to calculate a cost matrix: we model the investor as a network node and a portfolio as the specific service (latency or bandwidth) of interest.

$$S_p = \frac{\bar{r_p} - R_f}{\sigma_p} \quad (1)$$

The Sharpe ratio for a portfolio p, is computed by subtracting the risk-free rate of return (R_f) from the rate of the portfolio return itself ($\bar{r_p}$), and dividing by the standard deviation of the portfolio returns (σ_p), as shown in Eq. (1). Historical averages alone might not be appropriate if the associated standard deviations are high, because in such cases the effective metric of interest is much lower than the historical average. However, the Sharpe ratio takes the associated standard deviation in account, so it is a better candidate for approximating a cost function. Higher Sharpe ratios are equivalent to high averages and small standard deviations. Smaller Sharpe ratios are due to either high standard deviations, or to small averages. A negative Sharpe ratio implies that the risk-free rate of return would perform better than the portfolio being analyzed.

We create two cost maps, each specific to a class of applications. Cost map C_u is for P2P-class of applications that seeks out PIDs that host peers with high upload bandwidth. Cost map C_l is used by latency-sensitive applications (e.g., CDNs). C_u and C_l are computed by analyzing the Sharpe ratio (Eq. (1)), where $\bar{r_p}$ represents the actual average upload throughput (or average latency) for the given month and σ_p the standard deviation of the download throughput (or latency). R_f, the risk-free rate of return is modeled as one standard deviation below the yearly average upload bandwidth (or latency). Fig. 5 shows the Sharpe ratios for the upload bandwidth (Fig. 5a) and latency (Fig. 5b). The figure shows the monthly evolution (x-axis) of the Sharpe ratio (y-axis) for our 9 ISPs. The horizontal line in the graphs (at $y = 0$) represents the threshold at which the risk-free investment no longer is the best option. These ratios can be compared not only in terms of absolute values, but also on the temporal evolution for individual ISPs.

There are some interesting trends to explore. First, the data does not show any correlation between the upload throughput and the latency among the ISPs. However, Fig. 5b does show that most ISPs deliver acceptable latency according to the risk-free investment benchmark we establish. This is less the case with upload throughput (Fig. 5a), where there is a wider variation among ISPs delivering what we consider an acceptable upload speed. Second, a number of ISPs show continuous improvement over the year in upload throughput (Comcast, Cox, MCI, and Mediacom). This may be a result of cumulative engineering and provisioning. Alternatively, some ISPs shows a downward trend in the Sharpe ratio for the upload throughput as epitomized by Time Warner and Windstream. Time Warner's downward trend that starts around February and does not show a sustained uptick until July can be explained by its acquisition of Insight Communications, which was completed on February, 2012. After the merger, it took 5 months for the operating capacities to show growth.

To calculate the cost maps using the Sharpe ratio (Eq. 1), we consider PID to be a set of all the ISPs in Table II. The upload throughput cost map, C_u, is calculated using Eq. 2 below. Here, C_u^i is the upload throughput cost for each PID i, which is calculated over summation of the value of each month's Sharpe ratio for the PID i (S_m^i) and multiplying by a weight associated with the PID. To reward positive Sharpe ratios, we weigh the sum by the fraction of months (from the last 12 months) that the Sharpe ratio for PID i is positive (i.e., above the risk-free investment line). In ALTO, a lower cost value indicates a higher preference for traffic from a source to a destination. Therefore, the actual cost for each PID is further calculated from Eq. 3:

$$\forall i \in PID, C_u^i = \sum_{m=1}^{12} S_m^i * W_i \quad (2)$$

$$\begin{aligned} \forall i \in PID, Default\_cost &= \lceil max(C_u^1, C_u^2, ..., C_u^i) \rceil \\ \forall i \in PID, C_u^i &= Default\_cost - C_u^i \end{aligned} \quad (3)$$

That is, we first find the maximum cost across all PIDs; this becomes our default cost that is applied to the Default PID. Costs to the remaining PIDs are calculated as a distance from the default cost. The latency cost for each PID, C_l^i, is calculated in a similar manner. The computations of Eqs. 2 and 3 result in the cost matrices shown in Tables III and IV for C_u and C_l, respectively. The first column represents the source PID and the remaining columns the destination PID. Thus, a host in the PID Comcast will always prefer other hosts in Comcast first (cost: 0), followed by hosts in PID Mediacom (cost: 0.316) to optimize upload bandwidth, or hosts in the PID Time Warner (cost: 0.32) to minimize latency. The costs serve to connect the previously disconnected components of Fig. 4; Fig. 6 depicts the links formed by peers in the largest PID (Comcast) to peers in other PIDs based on the upload bandwidth cost of Table III. The width of the edges between the PIDs is a measure of preference; i.e., the PIDs connected by wider edges are preferred.

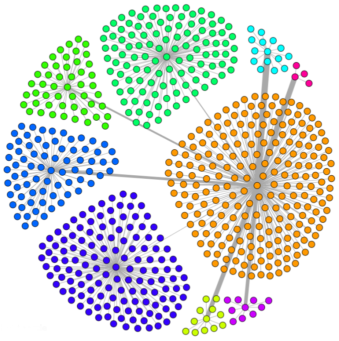
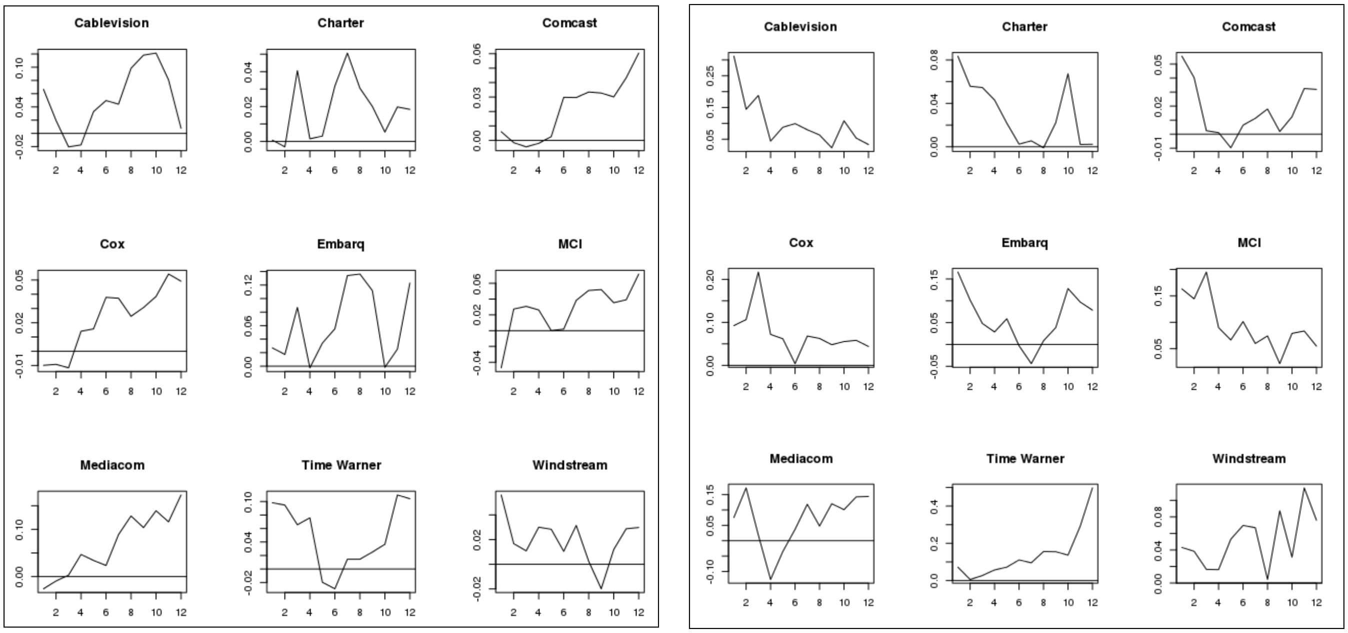


Figure 6: Connecting the Comcast PID based on C_u

An example helps explain the use of costs. Assume that a peer located in the Comcast PID wants to download content using a P2P network; it therefore



(a) Upload Throughput

(b) Latency

Figure 5: Sharpe ratios for upload throughput and latency

	Cablevision	Charter	Comcast	Cox	Embarq	MCI	Mediacom	TimeWarner	Windstream	Default
Cablevision	0.000	0.799	0.804	0.796	0.377	0.700	0.316	0.508	0.782	1.000
Charter	0.498	0.000	0.804	0.796	0.377	0.700	0.316	0.508	0.782	1.000
Comcast	0.498	0.799	0.000	0.796	0.377	0.700	0.316	0.508	0.782	1.000
Cox	0.498	0.799	0.804	0.000	0.377	0.700	0.316	0.508	0.782	1.000
Embarq	0.498	0.799	0.804	0.796	0.000	0.700	0.316	0.508	0.782	1.000
MCI	0.498	0.799	0.804	0.796	0.377	0.000	0.316	0.508	0.782	1.000
Mediacom	0.498	0.799	0.804	0.796	0.377	0.700	0.000	0.508	0.782	1.000
TimeWarner	0.498	0.799	0.804	0.796	0.377	0.700	0.316	0.000	0.782	1.000
Windstream	0.498	0.799	0.804	0.796	0.377	0.700	0.316	0.508	0.000	1.000

Table III: Cost map for upload bandwidth (C_u)

	Cablevision	Charter	Comcast	Cox	Embarq	MCI	Mediacom	TimeWarner	Windstream	Default
Cablevision	0.00	1.67	1.81	1.10	1.41	0.86	1.32	0.32	1.38	2.00
Charter	0.76	0.00	1.81	1.10	1.41	0.86	1.32	0.32	1.38	2.00
Comcast	0.76	1.67	0.00	1.10	1.41	0.86	1.32	0.32	1.38	2.00
Cox	0.76	1.67	1.81	0.00	1.41	0.86	1.32	0.32	1.38	2.00
Embarq	0.76	1.67	1.81	1.10	0.00	0.86	1.32	0.32	1.38	2.00
MCI	0.76	1.67	1.81	1.10	1.41	0.00	1.32	0.32	1.38	2.00
Mediacom	0.76	1.67	1.81	1.10	1.41	0.86	0.00	0.32	1.38	2.00
TimeWarner	0.76	1.67	1.81	1.10	1.41	0.86	1.32	0.00	1.38	2.00
Windstream	0.76	1.67	1.81	1.10	1.41	0.86	1.32	0.32	0.00	2.00

Table IV: Cost map for latency (C_l)

seeks PIDs with high upload capacities. The cost map for C_u (Table III) informs the querying peer to seek out peers in the local PID (i.e., Comcast) first (cost: 0.0), followed by looking for peers in the Mediacom PID (0.316), and then connect to peers in Embarq PID (0.377), and so on. If there aren't enough peers in the set of PIDs whose costs are known, the querying peer will seek out peers in the Default PID, where the cost is the highest. Similarly, when looking for peers with minimal latency (Table IV) , the querying peer would try peers in the PID Comcast first (cost: 0.0) followed by peers in the PID Time Warner (cost: 0.32), MCI (0.86) and so on before seeking peers in the Default PID.

The calculated costs accurately reflect how the particular PID behaves according to historical data; e.g., consider the PIDs Mediacom and Embarq in Table III and their associated curves in Fig. 5a. In the figure, the Embarq curve is above the risk-free threshold for all of the year when compared to the Mediacom curve. However, Mediacom cost in Table III is less than Embarq cost, making the former more attractive when forming connections. The cost for Mediacom reflects the upward slope of its curve when compared to the vagaries reflected by the Embarq curve. Such objective historical guidance reflected in the cost makes our approach a better solution.

V. Promoting supernodes via social network analysis (SNA)

In December 2010 Skype experienced a large-scale outage that reduced the global number of active supernodes by 25-30% [5]. A supernode acts like a directory, allowing normal Skype peers to rendezvous with each other and acting as relay nodes when necessary. Each supernode assumes responsibility for several hundred peers. Consequently, they become a single point of failure, and when they crash, as occurred in December 2010, a large part of the P2P network goes dark. The algorithm for promoting a peer to a supernode takes many attributes into consideration: uptime of a peer and how connected the peer is in the network are two indications that may be used. We analyzed the MBA dataset to discern hosts in a PID that may be candidates for promotion to a supernode. We present our analysis on the largest PID, Comcast. Comcast contains 726 hosts that remain stable throughout the year. To determine the host connectivity, we compute a Pearson correlation matrix with p-value = 0.003 and correlation value > 0.9. Using this matrix we construct for every month a number of metrics. These include the numbers of vertices and edges per month, the density, the number of cluster of connected components and the average number of components per cluster. The density of a graph is a ratio of the edges to the maximum number of possible edges. The cluster of connected components is the number of subgraphs that have at least two vertices that are connected to each other, and the average number of components per cluster gives us the average number of nodes over all subgraphs. Table V summarizes these metrics and Fig. 7 depicts the correlated components for the month of June (that month has the highest number of vertices). The data in Table V can be used by an algorithm that picks supernodes based on longevity and graph centrality.

Month	Vertices/ Edges	Density	Clusters/Avg. Components per Cluster
Jan	56/138	0.045	12/4.67
Feb	51/138	0.054	9/5.67
Mar	40/100	0.064	7/5.72
Apr	42/112	0.065	6/7.00
May	55/264	0.089	7/7.86
Jun	75/372	0.067	9/8.33
Jul	68/334	0.073	6/11.33
Aug	36/66	0.052	13/2.77
Sep	62/338	0.089	7/8.86
Oct	45/108	0.055	10/4.50
Nov	47/152	0.070	9/5.22
Dec	50/114	0.047	12/4.17

Table V: Comcast SNA

Social network analysis has other uses as well. For example, a tracker in Comcast PID that wants to choose local peers with good upload bandwidth can use historical data to precompute the clusters shown in Fig. 7. Furthermore, differences between connected components in the same PID but for different time frames and geographic locations are good markers for temporal variations — e.g., if two hosts (one in Chicago, other in Boston) are linked for a long time (say, Jan-Apr) but are not linked any more in May then this could be an indication of a significant change in the network capacities or in the traffic management policies.

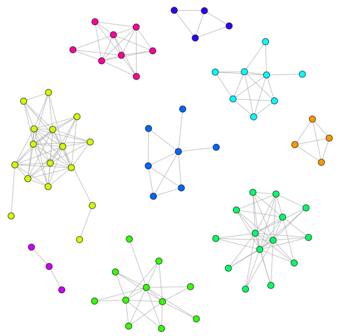


Figure 7: Correlated components in Comcast (June 2012)

VI. Related work and conclusion

D'Elia et al. [6] describe how ALTO's ranking service ranks peers in a P2P network in wireless mesh networks. Unlike our work that considers application classes (bandwidth sensitive, delay sensitive), D'Elia et al. depend on single cost that serves different application classes. Shibuya et al. [7] also provide an ALTO ranking service; however, their service uses the ISP's policies, which would not be available to a third-party.

Significant efforts [8], [7], [9], [10], [11] have been made to explore ALTO in conjunction with modeling the Internet topology at the Autonomous System (AS) level. Some of the papers simply use simulation to approximate ASes while others construct AS topologies through various means. The mapping of IP addresses to an AS provides a crude topology map and the BGP route announcements from ASes constitute a cost map. But techniques to create network and cost maps from ASes and BGP are not without disadvantages [12]. The information gathered by BGP and AS scraping does not provide the rich dynamics inherent inside an ISP. For instance, attributes used in our study, such as download speed, upload capability, and latency are not visible at the BGP routing layer.

Our work demonstrates the importance of historical data as an important source to create ALTO maps. We show how to provide multiple cost maps, each suited to a particular class of applications (i.e., delay- or bandwidth- sensitive). This is harder to achieve when BGP cost is the only available option. Future research activity will investigate how to extend this approach to wireless networks, where applications requirements can drive advanced vertical handover schemes.

References

[1] R. Alimi, R. Penno, R. Yang *et al.*, "Application Layer Traffic Optimization (ALTO) Protocol," RFC 7285 (Proposed Standard), 2014.

[2] "2012 measuring broadband america: July report," Technical Appendix, United States Federal Communications Commission, http://www.fcc.gov/measuring-broadband-america/2012/july, 2012.

[3] J. Jacobs and B. Rudis, *Data driven security: Analysis, visualization and dashboards*. John Wiley and Sons, 2014.

[4] W. F. Sharpe, "Mutual Fund Performance," *The Journal of Business*, vol. 39, no. 1, pp. 119–138, 1966.

[5] B. Trammell and D. Schatzmann, "A tale of two outages: A study of the skype network in distress," in *7th Intl. Wireless Comm. and Mobile Computing Conf. (IWCMC)*, July 2011, pp. 1282–1286.

[6] F. D'Elia, G. Di Stasi *et al.*, "Bittorrent traffic optimization in wireless mesh networks with ALTO service," in *IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2011, pp. 1–6.

[7] M. Shibuya, Y. Hei, and T. Ogishi, "ISP-friendly peer selection mechanism with ALTO-like server," in *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, Sept 2011, pp. 1–8.

[8] N. Pinthong and W. Lilakiatsakun, "Performance of bittorrent-like p2p file sharing systems inspired by ALTO," in *TENCON 2013 - 2013 IEEE Region 10 Conference (31194)*, Oct 2013, pp. 1–5.

[9] L. Guanxiu, Y. Suqi, and H. Xinli, "A novel ALTO scheme for bittorrent-like p2p file sharing systems," in *3rd Intl. Conf. on Intelligent System Design and Engineering Applications (ISDEA)*, Jan 2013, pp. 135–139.

[10] V. Burger, F. Lehrieder, T. Hossfeld, and J. Seedorf, "Who profits from peer-to-peer file-sharing? traffic optimization potential in bittorrent swarms," in *24th Intl. Teletraffic Congress*, Sept 2012, pp. 1–8.

[11] J. Seibert, R. Torres *et al.*, "The Internet-wide impact of P2P traffic tocalization on ISP profitability," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1910–1923, Dec. 2012.

[12] A. Khan, T. Kwon *et al.*, "As-level topology collection through looking glass servers," in *Proc. of the 2013 Conference on Internet Measurement Conference*, 2013, pp. 235–242.