Algorithms for Constrained Bulk-transfer of Delay-Tolerant Data

Parminder Chhabra, Vijay Erramilli, Nikos Laoutaris, Ravi Sundaram, and Pablo Rodriguez

***Abstract*—**

In recent years there has been renewed interest in the problem of transferring bulk data (terabytes) utilizing commercial ISPs. The need to transfer bulk data arises in various scientific and industrial applications. Today, this data is moved using postal service in conjunction with hard drives and DVDs or special high performance dedicated networks. The key insight underlying the recent work was that many of the applications are delay-tolerant and hence the bulk data can be transferred at minimal cost, utilizing already paid-for off-peak bandwidth resulting from diurnal traffic patterns, using store and forward through intermediate storage nodes.

In this paper we expand on this theme and consider the computational complexity of transferring data over a network whose links have time-varying capacities. We show that the general problem of finding a cost-optimal transfer of the bulk data can be solved in polynomial-time using minimum cost flow algorithms on a time-expanded version of the underlying network. Our solution involves graph transformations. We present additional transformations that enable the handling of half-duplex links (e.g. fiber-optic links) as well as node processing constraints (e.g. limitations on the processing power available for filtering or archiving). An important characteristic of our solution approach is the ability to handle nodes with storage. We consider nodes with storage that varies over time in terms of both capacity and cost. We show that our solution approach provably extends to cover the case of linear costs, providing polynomial-time algorithms. However, the flat-fee storage model is NP-complete and hence unlikely to be tractable in polynomial-time. Interestingly, with constrained storage, the optimal solutions may involve loops, i.e. the data may pass through the same node more than once on its way from the destination to the source along the optimal route.

We utilize data from one of the world's leading ISPs and perform a comprehensive evaluation of our algorithm. We show that there exists a huge potential for cost savings in real-world networks with time-varying costs for both link capacities and node storage.

***Index Terms*— Bandwidth, Scheduling, Internet, Architecture, CDN (Content Delivery Network), Store-and-Forward Networks**

Manuscript submitted September 28, 2009.

Ravi Sundaram (corresponding author) is an Associate Professor in the College of Computer Science, Northeastern University, Boston, MA USA (phone: 617-373-5876; fax: 617-373-5121; e-mail: koods@ccs.neu.edu).

Parminder Chhabra (email: pchhabra@tid.es), Vijay Erramilli (vijay@tid.es), Nikolaos Laoutaris (nikos@tid.es) and Pablo Rodriguez (pablorr@tid.es) are with Telefonica Research, Barcelona, Spain.

I. Motivation

The explosion of P2P services in the last decade has fundamentally altered how we distribute content and enable new services. The growth of social networking has also crucially changed how we interact, share and consume information. The observation that a vast amount of multimedia content downloaded (or mailed via a Netflix-like service) is not consumed right away, and so, is delay-tolerant (DT) has opened the possibility of offering bulk downloads as a service that the ISPs can offer. This has meant that ISPs have had to rethink their networks beyond merely routing and forwarding packets. ISPs can enable a variety of services for a range of applications that take advantage of bulk-data transfers, both for consumers and for businesses. As an example, today, Amazon provides a service (Amazon Import/Export [1]), which allows a user to transfer large volume of data across the country through Amazon's internal network (and avoiding the high transit costs on the Internet). Clearly, there is a demand for such a service.

There are several applications for such a DT service. The popularity of services like Netflix has meant that as a next generation service, movies may be available for download from the Netflix queue to an Xbox[14] or a similar device rather than via snail mail. The enormous growth of social networking sites has meant that companies need to synchronize their data warehouses both within the country and across continents. News organizations need to move multimedia content to their web-servers across the country so that data may be accessed from the closest server. The scientific community will also benefit greatly by moving large data sets across continents reliably, quickly and in a cost effective manner (like data from the Hadron Collider near Geneva that needs to shared with Research labs in Asia, North and South America). Clearly, there are a variety of applications that can take advantage of DT nature of traffic.

In previous work [9][10], we examined the potential of using the 95-percentile scheme to transfer bulk data by using storage nodes in the ISPs without adding additional transit cost to a customer. Since traffic follows a diurnal pattern, using non-coinciding peaks and valleys between sending and receiving ISPs, we showed the potential for transferring TeraBytes of data daily by taking advantage of *storage* nodes. In the current work, we will understand how to apply graph transformation and flow-based techniques to transfer bulk data by utilizing storage effectively.

II. Our Contributions

In this paper we present theoretical algorithms as well as experimental evaluations for the problem of transferring bulk data over a network with time varying capacities and costs. We focus on graph transformations, that are the necessary ingredient for formulating the bulk transfers problems to well known flow problems. In particular we focus on the effect of storage, where the nodes possess storage capabilities that also vary in capacity and cost with time. A summary of our major contributions is:

- We present a general method for transforming any network with dynamic capacities and costs to a static time-expanded and layered network. We show how the problem of finding an optimal schedule for transfer of bulk-data can be reduced to minimum cost flow on the layered network. This allows the problem to be solved in polynomial-time using well-known methods from Linear Programming as well as the algorithmic theory of flows.
- We demonstrate specific transformations for capturing half-duplex links (an essential feature of long-haul fiber-optic links) as well as for node capacities (models node constraints arising from functionalities such as filtering or archiving).
- We show how storage at nodes can be modeled through a graph transformation that captures dynamic capacities and costs. We present an example to demonstrate how the consideration of storage could lead to cycles in the routes for the data transfer (whereas, there will be no cycles in situations with zero storage or infinite storage). We also show that the flat-fee model for storage leads to an NP-complete problem.
- We experimentally evaluate our graph transformations and flow algorithms on a real-world topology with real capacity and cost constraints, based on data from one of the biggest ISPs in the world. We show that store-and-forward networks can be an order of magnitude cheaper than cut-through routing for the bulk-transfer of delay-tolerant traffic.

Our techniques are relatively straightforward and we do not claim novelty for the transformations themselves so much as the idea of applying them to capture the rich variety of situations involving bulk-data transfer in store and forward networks.

III. Related Work

The case for bulk-transfer of delay-tolerant data was made in a sequence of two papers [9][10]. The general approach of modeling networks as graphs and solving the routing problem using flows has a vast literature [1]. Linear programming is also a deeply studied area with well-understood polynomial-time algorithms such as the Ellipsoid Algorithm or Interior-Point Algorithms [6][14]. [13] is a good source on optical networks including half duplex links, and so is the excellent book [8]. Networks with time-varying links have been studied earlier, in the context of delay [12] rather than throughput. Another paper [11] studies networks with stochastically varying links whereas the focus of this paper is on networks with time-varying links that are completely deterministic and known in advance. To the best of our knowledge we are not aware of literature that studies the impact of storage in time-varying networks for the problem of bulk data transfer.

IV. Model and Terminology

We model a network as a directed graph $G = (V,E)$ with $n = |V|$ vertices (nodes) and $m = |E|$ directed arcs (links). We use lowercase letters to denote the individual elements, i.e. $V = \{v_1, v_2, \ldots v_n\}$ and $E = \{e_1, e_2, \ldots e_m\}$, where each $e_k = (i, j)$ is the directed arc from v_i to v_j. We assume the capacities and costs of the links to be time varying. However, it is well known that there is a strong dependence of the traffic levels, especially on backbone links, on the time-of-day and day-of-week and we assume that we know the exact traffic pattern. We assume time to be measured in discrete quanta of an hour each. For the purposes of this paper, we assume a maximum time T (which may be thought of as 24, since there are 24 hours in a day and typical bulk-data transfer are expected to complete within a day) and $t = 1,2,\ldots,T$ to stand for a specific hour. We use r (for rate) to denote the capacity of a link, specifically r_{ij}^t denotes the capacity of link (i, j) at time t. Observe that since r is a rate (bits per hour) and time is measured in units of an hour, hence r also represents the maximum amount of data (in bits) that can be transferred in that hour t across the link (i, j). Similarly, we use c (for cost) to represent the cost for transferring data. We use c_{ij}^t (dollars per bit) to denote the cost of link (i, j) at time t. We use s for storage) to denote the storage capacity of a node, specifically s_i^t (in bits) denotes the storage of node i at time t. We abuse notation slightly and also use c to denote the cost of storage, specifically c_i^t denotes the cost of storage (in dollars per bit-hour) at node i at time t.

V. Algorithms

A. Time-expanded Graphs

We first explain how we can transform a network with dynamically varying capacities and costs into a static network. The transformation is best explained with a diagram, see Figures 1 and 2.

Given a network with time-varying link capacities and link costs we create a time-expanded graph as follows: we create T copies of the vertex set V^1, V^2, … V^T. Each V^t is an independent set, i.e. there are no arcs between any two vertices of V^t. The arcs run between V^t and V^{t+1} for all $1 \le t \le T-1$. For each arc (v_i, v_j) we create T copies of the form (v_i^t, v_j^{t+1}) for all $1 \le t \le T-1$, each with capacity (rate) and cost corresponding to its time slot t. We call this new static graph the time-expanded graph. Observe that the time-expanded graph can be viewed as the original graph stretched out across the time dimension. Observe also that the capacities on the arcs in the time-expanded graph exactly correspond to the rate times one hour, or r_{ij}^t bits. It is easy to see that the following theorem follows in straightforward fashion.

Theorem For any pair of source-sink destinations, including multi-commodity versions, the minimum cost flow on the time expanded graph is the optimal routing scheme on the

underlying network topology with dynamically varying capacities and costs.

Proof (Sketch): It is easy to see that any routing scheme on the underlying network topology can be realized in the time-expanded graph by routing the flows at time t in the network topology on the arcs with superscript t in the time expanded graph. The correspondence in the reverse direction is similar. ■

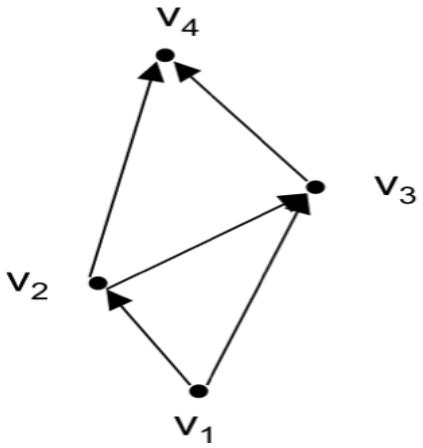

Figure 1: Underlying network topology

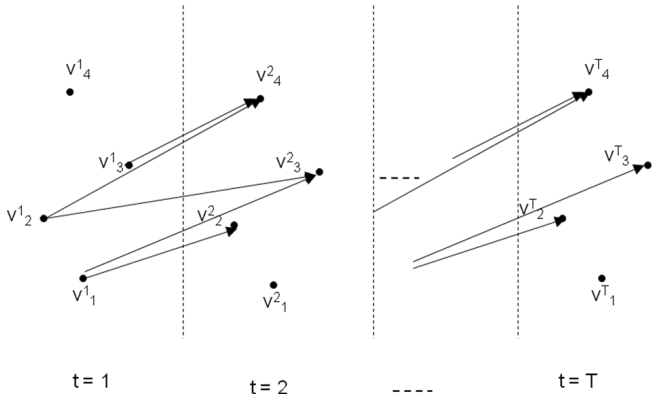

Figure 2: Time-expanded Graph

B. *Half-duplex links*

We now show our transformation can be extended to half-duplex links. Half-duplex links occur in fiber-optic networks. Such networks use wavelength division multiplexing (WDM) where the sum total of the frequencies in the two directions (uplink and downlink) is a fixed constant. In other words the link can be thought of as two arcs between the nodes i and j, (i, j) and (j, i) but with capacities summing up to a constant, i.e. $r_{ij}^t + r_{ji}^t = r^t$ It is important to be able to handle this case since fiber is the most commonly used transmission medium in long-haul networks. We now show how any such link can be represented in the time-expanded graph. Again we use a diagram for our illustration.

Theorem The transformation presented in Figure 3 exactly captures a half duplex link.

Proof (Sketch): Observe that the total flow from i to j and from j to i in the transformed gadget can never exceed r^t because it is limited by the vertical straight arc in the middle. ■

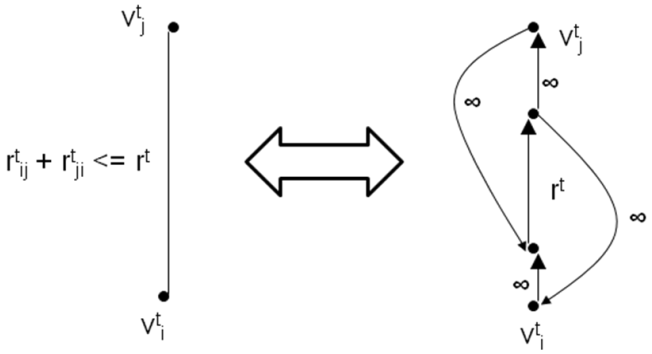

Figure 3: Half-duplex link and its transformation. Arc capacities are shown next to arcs.

C. *Node constraints*

Consider situations in which nodes have constraints on the amount of data they can handle. Such situations could arise for example when the node has to filter the data passing through it for security reasons or if it has to archive the data etc. We now show how such situations can be handled in the time-expanded graph. The basic idea to take any node it and create two nodes v'^t_i and v''^t_i and connect all original incoming arcs to v'_i and all original outgoing arcs to v''_i; we also add the arc (v'^t_i, v''^t_i) with capacity constraint r_i^t and associated cost constraint cit (if any). Again we provide a diagram explaining our transformation.

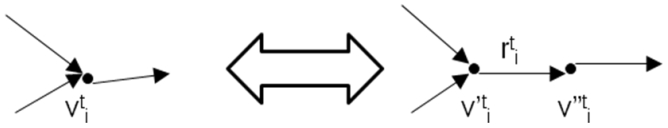

Figure 4: Node constraints in the time expanded graph.

Theorem The transformation presented in Figure 4 exactly captures node constraints.

Proof (Sketch): Observe that any flow from an incoming arc to an outgoing arc is forced to go through the arc (v'^t_i, v''^t_i). ■

VI. Storage

Up to now we have considered nodes with no storage. We have assumed that all routing is cut-through routing. In this section we consider the important case of constrained storage and present algorithms that take advantage of this in significant ways. The main thesis of this work is that store-and-forward networks allow for delivery of substantially larger quantities of data at lower costs.

A. *Infinite storage at zero cost*

Consider a network with infinite storage. The issue is how storage can be effectively represented in time-expanded graphs. The key insight here is that just as with space-time curves [3], the time-expanded graph is a space-time representation of a spatial object (the graph). Hence the storage at a node can be represented as a link between the same node at one time slot and the next. The following diagram should clarify the issue. The horizontal links with infinite storage and zero costs represent the storage at the nodes.

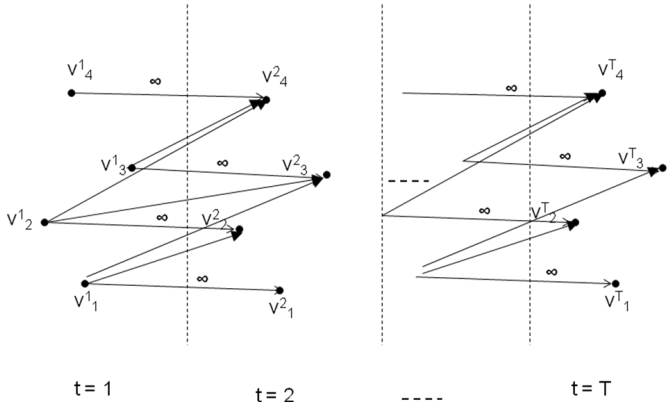

Figure 5: Infinite storage with zero costs.

B. *Flat-fee storage*

In some scenarios storage is charged on a flat-fee model where the user is charged for using any storage at all (up to some reasonable limit) independent of the actual amount used. This is a natural case to consider and it would be useful if we could extend our transformations so that existing flow algorithms and linear programs could be applied to this situation as well. Unfortunately as the following theorem shows the problem is NP-complete and hence is likely to be feasible in polynomial time whether by graph transformations or any other means [5].

Theorem The problem scenario with a flat fee for storage is NP-complete.

Proof (Sketch). It is easy to see that the problem is in NP. To show that the problem is NP-hard we will reduce another NP-hard problem, namely set cover to it. The reduction will maintain a correspondence between the number of sets in the set cover instance and the number of nodes at which storage is used. The entire reduction can be done with T = 3. Due to space constraints, we leave the details to the full version. ■

C. *Storage with time-varying capacities and costs*

The general problem of storage with time-varying capacities and costs is a generalization of the case with infinite storage and zero costs. As before, we add horizontal links of the form (v_i^t, v_i^t+1). We attach capacity r_i^t and cost c_i^t to the horizontal link at time t. The following diagram should explain the transformation clearly.

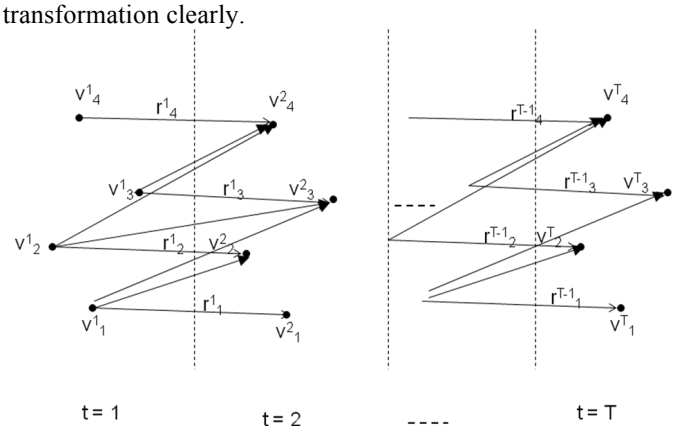

Figure 6: Storage with time-varying capacities and costs.

Theorem The transformation presented in Figure 6 exactly captures storage constraints at nodes.

Proof (Sketch): Observe that any flow across the arc (v_i^t, v_i^t+1) represents the amount of data stored at node v_i at time t. ■

Note that the units are dimensionally correct in that cost for storage is in dollars/bit-hour and hence if f bits flow across the link in one hour, or equivalently f bits are stored at that node in that hour then by multiplying by the cost we get the amount in dollars.

D. *An example with loops*

An interesting phenomenon occurs when we consider constrained storage with linear costs. Loops emerge in the optimal data routes. It is easy to see that when storage is infinite at zero cost then there is no need for the data to ever cycle - one can use the storage at a node to shortcut any cycles involving that node.

We briefly describe a simple example. Consider data at a node v_1 that wants to go to node v_3. v_1 has high costs of storage and high cost of transit to v_3 except in time slot 4 when the transit cost is low. Then, if there is a node v_2 with low transit costs to and from v_1 and with low storage costs as well, then the optimal route will involve a cycle where the data moves from v_1 to v_2, then stays there for 1 time slot, comes back to v_1 in time slot 3 and goes to v_3 in time slot 4.

VII. EXPERIMENTS

In this section, we describe the experimental setup for the algorithms that we used as discussed in Section V.

While we experimented with several nodes, we present results from a setup of Figure 7 for ease of explanation. In this figure, node V is the source in Western Europe and node U is the destination in South America. Intermediate nodes T1, T2 and T3 are three large well-provisioned transit ISPs, also in Europe. We associate storage nodes with all the transit ISPs. We obtained all the traffic load information from one of the largest transit providers in Europe that peers with ISPs from all continents. We use data from real traffic links.

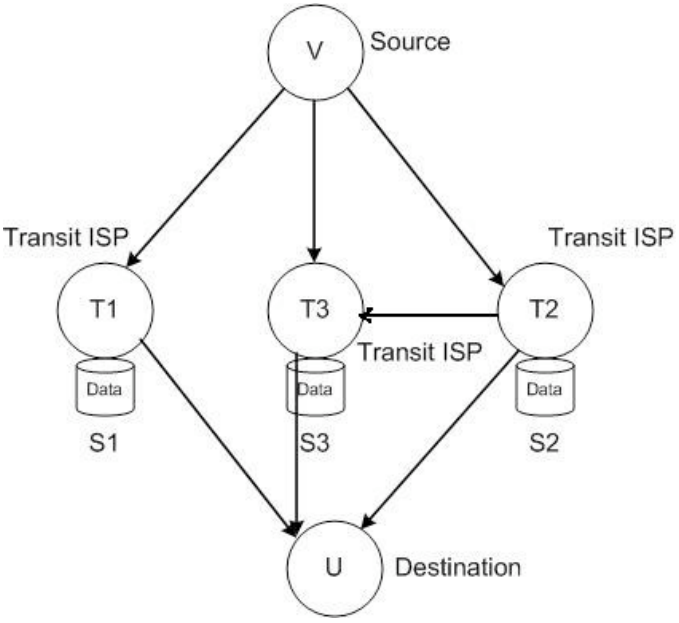

Figure 7: Topology used in the sending terabytes from a sender V in Europe to a destination U in South America.

We consider the cost of storage at the rate of $0.40/GB/Month [3] (about thrice the cost of Amazon's S3). Using this as a basis, we associate storage cost into our network model. In addition, we consider recent transit costs of $30K /Gbps/month [3][10] as a basis for calculating the data transfer cost. To make our study richer, we assumed that data transfer rate between the transit links T2 and T3 (within Europe) was about half the transit cost above. We work with one day's worth of data. The original data in 5-minute intervals was converted to hourly data. The data in all links was found to have non-coinciding peaks and valleys.

We constructed a Time-expansion graph that consisted of 122 nodes and 300 edges and executed this against Andrew Goldberg's Network Optimization Library [6] to compute the minimum cost flow.

To benefit from network storage in our experiments and justify the usefulness of the proposed algorithms, we will attempt to transfer 6TB, 18TB and 30TB in a 24-hour time window in the given network setup. These transfers correspond to different applications (data transfers across web/content servers, synchronization of multimedia content across data-centers and data transferred from the Hadron Collider respectively) that could benefit from such a service. We consider the following three scenarios (i) First, we consider how much it would cost to send data without storage nodes at current data transit rates. (ii) As a second step, we add storage nodes at the intermediate ISPs and (iii) Finally, we consider a variable transit cost model where the cost of data transfer rises during peak hours. For simplicity, we consider the transit cost during peak hours to be twice the normal transit cost.

Data	6 TB	18 TB	30 TB
No storage	0.56 Million	1.8 Million	4 Million
Storage	30K	360 K	1.2 Million
Storage + Varying cost	30K	400 K	1.4 Million

TABLE I: Table showing the result of monthly transit cost (in US Dollars) for daily data transfer volume of 6TB, 18TB and 30TB from Source V in Europe to Destination, U in South America.

We summarize the results from our analysis in Table I. We observe the following:

- With no storage at the ISP, we can transfer several TB of data everyday with an almost linearly increasing transit cost.
- With storage, the cost of data transfer does not increase linearly as the amount of data transferred. This shows that the data transfer is taking advantage of non-coinciding peaks and valleys between two links to send majority of the data during non-peak hours at the receiver without a major increase in transit cost till about 18TB. For the 30TB transfer, the transit cost contributes significantly to the data transfer. The US $30K is the normal monthly transit cost for daily 6TB transfer.
- When we introduce non-uniform pricing into the picture, again, we notice that the cost does not increase linearly with the amount of data transferred. The increase from varying cost during the peak hours is marginal. Like before, transit costs play a significant role for a 30TB transfer.

Due to a variation in transit costs across continents, the transit cost for the same amount of data to the US would be about a third of the numbers presented above.

VIII. Conclusions and Future Work

We presented a general scheme for representing a network, with dynamically varying capacities and costs for both links as well as storage, as a graph and showed how algorithmic techniques from the theory of flows and linear programming could be utilized to solve the problem of bulk data transfer. Our simulations involved the use of data from a real-world network from some of the largest ISPs in the world, and demonstrated the practical feasibility of our algorithms. We showed that by utilizing storage at intermediate relays it is possible to transfer bulk data at much lower costs, since the storage buffer allows the data to be transferred during the low-cost periods (valleys) of both the incoming and outgoing links. However, the algorithms presented in this paper rely on full knowledge of the traffic patterns and utilize centralized algorithms. In practice, traffic patterns have some variability and the models must be enhanced to accommodate this richness. Further, distributed algorithms are more robust and fault-tolerant as compared to centralized algorithms which require a central entity to collect all the information and disseminate the resulting algorithmic decisions.

In recent times there has been much interest in game-theoretic formulations where there are multiple strategic entities each pursuing their own self-interest. It would be interesting to generalize our model to such situations. One possible approach is to consider nodes to be partitioned into groups (ISPs) and have a utility function for each group that captures the group's costs and benefits.

References

[1] Ahuja, R., Magnanti, T., and Orlin, J., "Network Flows: Theory, Algorithms and Applications", Prentice-Hall, 1993.

[2] Amazon Import/Export. At http://aws.amazon.com/importexport/

[3] Church. K, Grenberg, A, Hamilton, J. "Delivering Embarissingly Distributed Cloud Services. Proceedings of ACM HotNets – VIII.

[4] Feynman, R. "The Feynman Lectures in Physics", Addison-Wesley, 1970.

[5] Garey, M., and Johnson, D., "Computers and Intractability: A Guide to the Theory of Incompleteness", Freeman, 1970.

[6] Goldberg, A. Network optimization library. Available at http://www.avglab.com/andrew/soft.html

[7] Grotschel, M., Lovasz, L., and Schrijver, A., "Geometric Algorithms and Combinatorial Optimization" Springer-Verlag, 1988.

[8] Kurose, J., and Ross, K., Computer Networking: A Top-Down Approach", Addison-Wesley, 2009.

[9] Laoutaris, N., and Rodriguez, P., "Good things come to those who (can) wait or How to handle delay tolerant traffic and make peace on the Internet", Proceedings of ACM HotNets – VIII.

[10] Laoutaris, N., Smaragdakis, G., Rodriguez, P., and Sundaram , R., "Delay tolerant bulk data transfers on the Internet", Proceedings of ACM SIGMETRICS'09, pp. 229-238.

[11] Orda, A., Rom, R., and Sidi, M., "Minimum-delay routing in stochastic networks", IEEE Transactions of Networking, 1, pp. 187-198, 1993.

[12] Orda, A., and Rom, R., "Shortest-path and Minimum-delay Algorithms in Networks with Time-dependent Edge-Lengths," Journal of the ACM, 37, pp. 607-625, 1990

[13] Ramaswami, R., and Sivarajan, K., "Optical Networks: A Practical Perspective", Morgan-Kaufmann, 2001.

[14] Schrijver, A., "Theory of Linear and Integer Programming", Wiley, 1998.

[15] Xbox live and Netflix. At http://www.xbox.com/en-US/live/netflix/default.htm