A graph theoretical approach to traffic management in ISP networks using the overlay MPLS network

Radu Dragos
Communication Center
"Babes-Bolyai" University
400084, Cluj-Napoca, Romania
bradu@ubbcluj.ro

Sanda Dragos
Faculty of Mathematics and Computer Science
"Babes-Bolyai" University
400084, Cluj-Napoca, Romania
sanda@cs.ubbcluj.ro

Abstract

We describe a problem of traffic redundancy over Internet Service Provider Networks (ISPs) that occurs when the links to the client's network are highly utilized. The ingoing traffic is highly unpredictable when multiple ingress points exists for the flows originating in the Internet. Therefore, traffic engineering mechanisms are more difficult to apply. A solution for avoiding this problem by keeping the redundant traffic away from the ISP's network is proposed. Simulation results prove that the ingress flow cancellation method, reduces considerably the negative impact of the excess traffic on the network.

1 Introduction

According to the Internet Engineering Task Force(IETF), traffic engineering is defined as that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks [4].

The main traffic engineering objective is to enhance the performance of the network at both traffic and resource levels. MPLS plays an important role in engineering the network to provide efficient services to its customers.

RFC 2702 specifies the requirements of traffic engineering over MPLS and describes the basic concepts of MPLS traffic engineering like traffic trunks, traffic flows and LSPs [3]. The advantages of MPLS for traffic engineering include:

- label switches are not limited to conventional IP forwarding by conventional IP-based routing protocols;
- traffic trunks can be mapped onto label switched paths;
- attributes can be associated with traffic trunks;
- MPLS permits address aggregation and disaggregation (IP forwarding permits only aggregation);
- constraint-based routing is easy to implement;

The main advantage of using MPLS for traffic engineering is ability to use other than the shortest paths selected by the IGP to achieve an optimal network utilization. In the MPLS environment, this can be achieved by moving the traffic away from the over-congested shortest paths using explicit LSP tunnels in the overlay MPLS network.

2 Optimization of flows in MPLS networks

In this section we use the terms network or data network for an administrative domain (an ISP network) as a subset of the Internet. We also refer to edge nodes as gateways to avoid the confusion with the term edges used in graph theory.

2.1 Representing a network

A digraph D is a set of elements called *vertices* (the *vertex-set*) and a list of ordered pairs of this elements, called *arcs*(the *arc-list*) [7]. In the following sections we use sometimes the pair (i, j) to represent an arc e between vertices i and j.

A *basic network* is a digraph N with the following properties:

1. N has exactly one source and one sink (destination).
2. each arc *e* of N has associated a positive number $c(e)$ called capacity.

A data network is a generalization of a basic network since we have multiple sources and destinations (gateways can be sources or sinks or both of them). The routers forms the $vertex - set$ and the links between them represent the

$arc - list$. The values for the arcs capacities are given by the links bandwidth.

In graph theory, digraphs are sometimes represented using the adjacency matrix. The adjacency matrix for a digraph D with n vertices, $M(D)$ is a $n \times n$ matrix where $m(i, j)$ is the number of arcs from vertex i to vertex j.

In the Internet most of the routers are interconnected using a single link. If there exist more than one link, the links can be bounded [6] into a single virtual link having a bandwidth equal to the sum of all the bounded links. Therefore, the adjacency matrix $M(d)$ will contains only the element $\{1\}$ if exist a link or 0. Moreover, since each link has associated capacity and if there's no link the capacity is 0, $M(D)$ can be combined with capacities matrix: $m(i, j) = c(e)$, where e is the arc from i to j.

2.2 Flows in basic network

For a particular pair of gateways (source and sink), the network can be particularized to a basic network. A flow in a basic network with source S and sink T is a function f which assigns for each arc e a non-negative number $f(e)$ called the flow along the arc e, satisfying the following 2 conditions:

1. Feasibility: $f(e) \leq c(e)$;
2. Flow conservation law: For each $i \in D \setminus \{S, T\}$ we have $\sum f(j, i)$ =$\sum f(i, k)$ where $j, k \in D$ and $m(j, i) \neq 0$ and $m(i, k) \neq 0$

The value of a flow is the sum of all the flows into T; that is $\sum f(i, T)$ for $i \in D \setminus \{T\}$ and $m(i, T) \neq 0$. By the flow conservation law this is equal with the total flow out of S. A maximum flow is a flow with the maximum possible value. For a data network, maximum flow is the upper bound for the data traffic that can be sent from S to T.

The maximum flow in a basic network can be determined using the Ford-Fulkerson algorithm [9]. The algorithm does not only obtain a maximum flow from S to T but also constructs the paths through which the flow will propagate. Usually this is more than a single path.

In traditional hop-by-hop routing, the maximum flow cannot always be achieved because the shortest path algorithm will determine a single path between a source and a destination. The explicit routed MPLS LSPs allow traffic disaggregation and therefore the upper bound traffic can be achieved.

2.3 Flows in generalized networks

Data networks are more complex than basic networks since there exist more than one source and one sink. However, an upper bound traffic value can be calculated using the max flow algorithm. This is done by adding a $supersource$ (a virtual vertex connected to each source by an arc e with $c(e) = \infty$) and a $supersink$ (a virtual vertex to which every sink is connected by an arc e with $c(e) = \infty$). The maximum flow between $supersource$ and $supersink$ give us the maximum value of the traffic that can be sent trough the network.

Without loss of the generality we can assume that each gateway is both a source and a destination for a flow in the network. Therefore, we need $n(n - 1)$ virtual links to mesh-connect the gateways. For each pair of gateways (source,destination) a maximal flow can be discovered. But since we have multiple flows through the network, the flows will interfere in the sense that some physical links (arcs) will be used by multiple flows. Therefore, the bandwidth of the link will be shared between multiple flows and the capacity divided (based on a fair sharing policy) and the flows will not be maximal.

An algorithm that optimize the flows in order to obtain a maximum throughput is a very important management tool. Offline optimization could be performed using a traffic management server (bandwidth broker). A hierarchy of bandwidth brokers will improve the scalability of MPLS in the Internet and thus will facilitate end to end bandwidth guaranteed tunnels delivery. Unfortunately, there is no result like the max-flow theorem for general networks given $multi - commodity$ flows [8]. An algorithm for optimal distribution of traffic across the links in respect with a fair bandwidth sharing policy can not be implemented.

2.4 Offline versus online routing

Since there is no optimal solution for provisioning the bandwidth of the virtual links between gateways in the overlay model, the optimization should be performed online before each flow enters the network.

2.4.1 The reservation model

There are two main signaling protocols used to reserve resources for MPLS LSPs. This are Constraint-Routing Label Distribution Protocol (CR-LDP) [1], and Resource Reservation Protocol (RSVP-TE) with object extensions for LSP tunnels [2]. In the issue of having two functional similar (not to say overlapping) protocols addressing the same functional space, IETF has choose to focus the research in the favor of RSVP-TE [10].

RSVP can make use of an IP routing protocol to find a viable path from an ingress gateway to an egress gateway and reserve resources along the path. But since IP routing protocols use a shortest path routing algorithm, congestion may occurs along some links. RSVP-TE can perform explicit reservations of LSPs. A path discovery protocol should be used to find an optimal path and then RSVP will try to reserve resources along it.

If an offline model is deployed, then the path is known prior to the commencement of the flow and the RSVP will use this path to make the reservations. If an online algorithm is considered, then based on the state of the network this algorithm will compute the optimal path to be used by RSVP.

Two traffic engineering problems arise from this type of reservation:

Resource over-utilization (congested links). In the reservation scenario, if the first arrived requests reserve and use all the available bandwidth (i.e., $f(e) = c(e)$), then all the incoming requests will be blocked. In the worst scenario this could block a virtual link in the overlay network. Therefore, the mesh could not be created and this means there is no service between the two unconnected gateways.

Resource under-utilization. The apparent solution to the above situation is to prevent the requests to reserve all the available bandwidth. If more virtual links share a common link, a policy should prevent the first incoming requests to flood the link with traffic. All the request will get their own share without exceeding the limits given by a fair share policy. In this way, bandwidth is reserved on a per link based strategy. But this may result in resource under-utilization since for an undefined period of time, resources are reserved but never used.

There are some proposals which try to reduce the negative effects of the above mentioned problem. One of them is based on the philosophy that the flows should interfere as less as possible [11]. When the computation of an optimal path for a new flow is performed, the reservation is made for a path which complies with the traffic constraints but also will have a minimal interference with the existing flows and possible incoming flows.

2.5 Online reservation and offline optimization

We showed that an offline provisioning algorithm can not be deployed and an online algorithm lives open the problem of resource management. A more complex scheme should be consequently deployed.

2.5.1 The initial state

In an initial state of the network such as before any traffic is sent, some initializations should be made. One of the first stage is to compute the upper bound traffic that the network can carry. Algorithms such as the max flow, do not only find a flow of the maximum value but can also reveal links and nodes that will never (or are unlikely to) be used. Moreover, critical links and nodes can be discovered.

For an overlay model, maybe the simplest scenario is to provide bandwidth guaranteed tunnels between gateways. This is the situation for services such as VPNs or IP telephony where there is little or no fluctuation between source and destination pairs.

A more complex situation occurs when there is no stability in the overall traffic between pairs. This is the case for most of the Internet traffic since one can not predict what Web or FTP server will be used by a client. For a client request entering a gateway, the outgoing gateway can not be predicted. In the overlay model, the traffic along the virtual paths can no longer be predicted.

An ISP offering differentiated classes of service has an even more complex model. Not only bandwidth for the virtual links has to be shared but classes of priorities must be introduced for the classes of service used.

Whatever the complexity of the scenario, in the initial state, some network parameters must be set in order to support the reservation model. One of this parameter is the percentage of a critical link that can be reserver for a particular flow or pair ingress-egress. If the ISP allows for over-reservation than the flows that exceed the limit should be either short lived or of a low class of traffic. This will alow a higher priority class flow from a concurrent virtual link, to get the desired bandwidth share.

2.5.2 Online reservation

Once the initialization phase has ended, the network is ready to receive reservations for flows. The connections arrive one-by-one and make reservation requests. If in the initialization phase, a path was computed for this particular flow, than RSVP can make the reservation along an explicit path. Otherwise, the path compliant with the traffic constrains will be determined and reservation made.

One advantage with MPLS is that a trunk of flows can be disaggregated and forced to use multiple paths toward the egress gateway. Therefore, load sharing algorithms can be deployed along multiple paths. This will also reduce the blocking probability for flows requiring a large bandwidth if the residual capacity of a single path ($c(e) - f(e)$) is lower than the bandwidth requested. If multiple paths are detected and the sum of their residual capacity is greater than the requested flow, the request will be accepted (i.e. there exist a maximum flow in the residual network with a value greater than the bandwidth requested).

2.5.3 Offline reoptimization

The general Internet traffic is difficult if not impossible to predict. But some patterns may be observed by a traffic monitoring tool. Another phase of engineering the MPLS

tunnels is LSP re-optimization following previous statistics of the network traffic patterns.

In this phase the initializations made in the first stage may be modified based on the activity within the network. Modifications can be made at both physical and logical layers. At the physical layer, re-optimizations can be made for the underlaying network by removing the unused resources and improving the capacity of the highly loaded links. At the logical level the mappings of the virtual links in the overlay network can be changed to better suit the demand for resources.

At this stage, some other issues may also be considered, since there are classes of services that requires more than just bandwidth constraints. Factors such as delay, jitter and costs may also be considered. As a common property all the above mentioned values must be minimized. Therefore, in graph theory these values are introduced as costs of the flow traversing a link (arc).

There is a different algorithm for this class of problems. The algorithm is known as a "min cost flow" algorithm. The philosophy is to find a flow from source S to sink T with a minimal cost. If delay or cost constraints are to be considered, than variations of this algorithm may be used do determine an optimal flow.

In the offline optimization phase, some other computations may be performed. Maybe the most important issue is to prepare the network for a failure. An offline optimization mechanism (e.g. a dedicated server) could also compute recovery paths for the situation when one or more resources become unavailable due to a physical or logical failure. For every possible scenario, a recovery strategy can be deployed. For example in the case of a link failure, a backup LSP can be provided for the MPLS tunnels using that link. This may reduce the recovery period and increase the network availability.

This stage may be repeated as often as necessary and as long as there is traffic through the network and traffic engineering tasks to be performed.

3 Redundant traffic in ISP networks

One problem with overlay networks is that the traffic pattern may not be predictable for some classes of traffic. Let's consider the following scenario.

A client (e.g. university) is connected to Internet via an ISP. The ISP has multiple connections to Internet via multiple ISP neighbors. When a student using a Web browser initiates a file download, one can not predict which virtual path will be used by the TCP flow. For every connection of this type the ISP will encounter a similar problem. Therefore it is difficult to provide guaranteed bandwidth along the ISP network since the traffic pattern is not predictable and distributed across multiple virtual paths.

In this section we describe the problem of traffic redundancy over ISP networks in the situation when the links to the client networks are highly utilized. The proposed solutions are: the RSVP based solution; and he ingress flow cancellation. In most of the situations a client is connected to one ISP by a single link. The available bandwidth is limited by either the capacity of the link or by a SLA (service level agreement). If the client has multiple up-links to internet the problem can be reduced by considering each link in particular.

The link is used by two types of traffic:

outgoing traffic - traffic originated in the client's network and ended somewhere else in the Internet;

ingoing traffic - traffic originated in the Internet and having as destination a computer inside the client's network.

The outgoing traffic is easy to "engineer" since it is limited by either the available bandwidth or by a traffic shaper. The ingoing traffic is more difficult to manage and some problems may occur if no traffic engineering task is performed.

3.1 Traffic exceeding the maximum link capacity

In this section we describe some problems that occur in an ISP network regarding the clients ingoing traffic.

Multiple traffic flows originating in various nodes in Internet passes the ISP network and are multiplexed along a single network link toward their destinations inside the client network. The entire client network can be virtually merged to forme a single virtual destination node (sink) complying the following rule:

$\sum f(i,0) \leq c(1,0); i \in Internet$; where 0 is the client network virtual node and 1 is the border node between ISP and the client.

The ingoing flow is less than the capacity of the link which connects the client to the ISP. Therefore, apparently the ingoing traffic can be shaped and policed at node 1 where the traffic exits the network. On the other hand, if the link $(1,0)$ is congested, this will result in packets being dropped at node 1. This will enforce the SLA (if one exists) but may dramatically decrease the performance of the network since a large amount of traffic traverse the network (reducing the value of the residual bandwidth) and is dropped at the egress of the network.

3.2 Simulation results

We used the *ns network simulator* [12] to simulate the above mentioned situation. Since most of the Internet traffic is TCP based we simulated multiple TCP flows having

as destination the virtual node 0. The link $(1,0)$ has a capacity of $1Mbps$. All the other links are $2Mbps$. Multiple TCP flows are initiated and will compete for bandwidth over the link $(1,0)$. Regardless of the time when the flows begin, the TCP flows will use equal shares of bandwidth over $(1,0)$ equal with the link capacity divided by the number of flows. The situation is slightly different over the ISP links. The flows will generate in average an equal amount of traffic along the ISP links but with a higher value as shown in 1. The simulations results shows also that packets are being dropped by the sending queue at node 1. Hence, the traffic over the ISP network is greater than the traffic received by the client. Therefore, the client generates a greater amount of traffic than it pays for. We run the simulations for respectively 5, 10 and 30 simultaneous TCP flows. The difference between the traffic values over the ISP network ($average1$) and over the client's up-link ($average2$) are shown in table 1. The values presented are for a single client and relatively small number of TCP flows. In real life one ISP may have hundreds or thousands of clients each with hundreds of simultaneous incoming TCP flows.

flows	average1 (/flow)	average2 (/flow)	Excess traffic (/flow)	Total traffic excess
5	75KB/s	25KB/s	50KB/s	250KB/s
10	39KB/s	12KB/s	27KB/s	270KB/s
30	14KB/s	4KB/s	10KB/s	300KB/s

Table 1. Traffic differences between ISP and client side

We considered here the most favorable case when all the connections were TCP based. The windowing mechanism for TCP allows for a traffic stream to get feedback from the network and therefore to adjust the traffic to the available bandwidth. In a real scenario there will exist one or more UDP or RTP flows which will continue to send data at a constant bit rate regardless of a congested link. This will make UDP or RTP traffic the most bandwidth consuming flows and dramatically reducing the bandwidth available for TCP traffic as shown in Fig. 2. CBR traffic exceeding the capacity of the link will generate even a greater difference between the traffic over the ISP network and the data actually received by the client.

3.3 Framework for redundancy avoidance

In a QoS capable Internet the above problem will never occur since the bandwidth could be reserver along the whole path from source to destination.

In the current Internet architecture, it is not very probable that a Web service provider will initiate reservation requests for each file requested by a client in order to guarantee the service quality. In this scenario, the ISP must deploy it's own reservation mechanism in order to prevent wasting bandwidth.

3.3.1 The RSVP reservation solution

RSVP could be used to allocate bandwidth along the ISP network for LSP tunnels. This will enforce the flows to use a fixed, limited amount of bandwidth along the ISP network. Therefore the flow conservation law will be preserved at node 1.

In this approach RSVP must be deployed in every network node. The problem is then similar to the scalability issue for intserv [4]. For each micro-flow reservation states must be created and maintained along each node.

One possible solution is to aggregate the traffic entering a certain ingress node i and having as destination node 0.

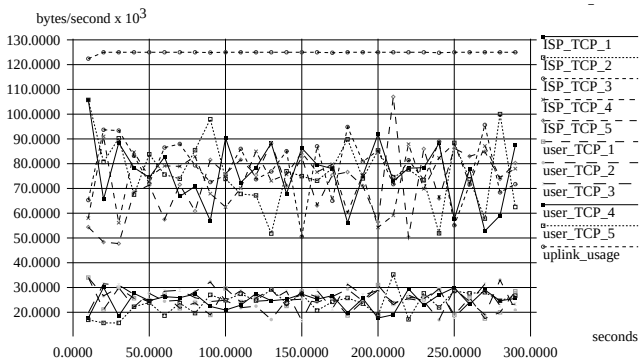

Figure 1. The traffic values over the ISP network and over the client's up-link

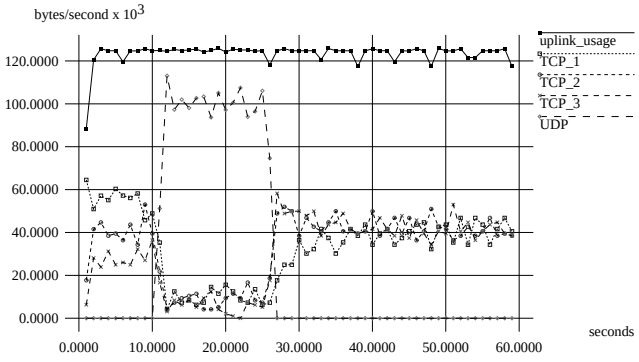

Figure 2. The influence of UDP traffic on a highly utilized link

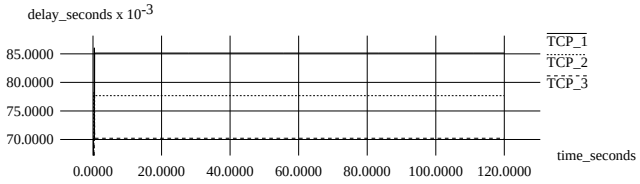

Figure 3. The delay values of TCP flows in the ingress flow cancellation approach

LSP tunnel can be used to maintain the traffic aggregates along the overlay MPLS network. This operation has to be performed for every possible ingress node. Queuing policies must be applied for the flows inside the traffic trunk to prevent UDP flows to take a higher priority than TCP traffic.

An heuristic algorithm must also be implemented to distribute bandwidth available over the link $(1, 0)$ to the merging LSP tunnels. This is not a trivial task since the Internet traffic is not predictable.

3.3.2 Ingress flow cancellation

A more simpler solution is to apply the shaping and policing mechanisms only at the ingress nodes. The idea is that if the traffic is shaped at the ingress node based on the current network state (i.e. available bandwidth and buffers), no congestion will occur along the network. The principle is to keep the traffic away from the network by dropping the packets at the ingress as opposite to egress. A similar solution but for providing bandwidth guaranteed flows was proposed in [5].

With this approach, both best-effort and guaranteed bandwidth traffic can be provided. For best effort traffic packets are dropped before entering the network if the traffic exceeds the maximum capacity of the client up-link. For bandwidth guaranteed traffic LSP tunnels can be provided to connect the incident traffic with the client's network.

This approach requires a protocol to deliver the state information from egress node 1 to the ingress nodes. SNMP clients at the ingress nodes can get the information from the egress node and based on a best-effort or more advanced scheduling scheme, can shape the traffic before entering the network.

With this approach, the exceeding traffic is kept away from the network and the jitter encountered along the network is virtually nonexistent. Fig. 3 shows a constant delay for the simulation of three flows shaped at the ingress routers. In empirical network low delay variations may occur.

4 Conclusions

One important consideration is the difficulty to deploy a large scale traffic management framework in the Internet. While some approaches deals with the process of transforming the core of the network to support QoS aware services, we propose a solution for ISP networks. Since the change of the Internet protocols will not be deployed at once, the idea is to change it piecemeal in a way that the protocols being deployed are backward compatible with legacy protocols.

MPLS and RSVP are the protocols likely to be deployed in the next generation QoS capable Internet. Used together with optimization algorithms from graph theory, these traffic engineering mechanisms can dramatically improve the performance of the live network.

A solution for increasing the available bandwidth by dropping excess traffic before entering the network was proposed. An implementation of a centralized or distributed system for a better traffic management is considered for future work. This system may help ISPs to overcome the difficulties of deploying new traffic engineering mechanism by simplifying the network management process.

References

[1] J. Ash, M. Girish, E. Gray, B. Jamoussi, and G. Wright. Applicability statement for CR-LDP. Technical Report RFC3213, IETF, January 2002.

[2] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. Rsvp-te: Extensions to rsvp for lsp tunnels. Technical Report RFC3209, IETF, December 2001.

[3] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over MPLS. Technical Report RFC2702, IETF, September 1999.

[4] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. P. Xiao. Overview and principles of internet traffic engineering. Technical Report RFC3272, IETF, May 2002.

[5] S. Bhatnagar and B. Vickers. Providing quality of service guarantees using only edge routers. September 2002.

[6] Cisco Systems. EtherChannel. http://www.cisco.com/en/US/tech/tk389/tk213/tech_protocol_family_home.html.

[7] A. Dolan and J. Aldous. *Network and Algorithms. An Introductory Approach*. John Wiley & Sons, September 1995.

[8] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, 1985.

[9] M. Gondran and M. Minoux. *Graphs and Algorithms*. John Wiley & Sons, 1986.

[10] IETF MPLS WG. MPLS IETF WG mailing list archive. URL:http://cell.onecall.net/cell-relay/archives/mpls/mpls.index.html.

[11] M. S. Kodialam and T. V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. In *INFOCOM (2)*, pages 884–893, 2000.

[12] T. N. Simulator. The network simulator - ns-2. URL:http://www.isi.edu/nsnam/ns/.