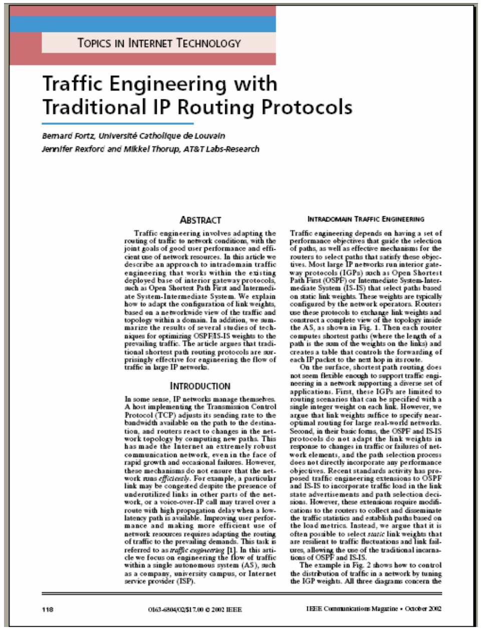
Traffic Engineering with Traditional IP Routing Protocol

By: B. Fortz, J. Rexford and M. Thorup



Traffic Engineering with Traditional IP Routing Protocols

Bernard Fortz, Université Catholique de Louvain
Jennifer Rexford and Mikkel Thorup, AT&T Labs-Research

ABSTRACT

Traffic engineering involves adapting the routing of traffic to network conditions, with the joint goals of good user performance and efficient use of network resources. In this article we describe an approach to intradomain traffic engineering that works within the existing deployed base of interior gateway protocols, such as Open Shortest Path First and Intermediate System-Intermediate System. We explain how to adapt the configuration of link weights, based on a networkwide view of the traffic and topology within a domain. In addition, we summarize the results of several studies of techniques for optimizing OSPF/IS-IS weights to the prevailing traffic. The article argues that traditional shortest path routing protocols are surprisingly effective for engineering the flow of traffic in large IP networks.

INTRODUCTION

In some sense, IP networks manage themselves. A host implementing the Transmission Control Protocol (TCP) adjusts its sending rate to the bandwidth available on the path to the destination, and routers react to changes in the network topology by computing new paths. This has made the Internet an extremely robust communication network, even in the face of rapid growth and occasional failures. However, these mechanisms do not ensure that the network runs *efficiently*. For example, a particular link may be congested despite the presence of underutilized links in other parts of the network, or a voice-over-IP call may travel over a route with high propagation delay when a low-latency path is available. Improving user performance and making more efficient use of network resources requires adapting the routing of traffic to the prevailing demands. This task is referred to as *traffic engineering* [1]. In this article we focus on engineering the flow of traffic within a single autonomous system (AS), such as a company, university campus, or Internet service provider (ISP).

INTRADOMAIN TRAFFIC ENGINEERING

Traffic engineering depends on having a set of performance objectives that guide the selection of paths, as well as effective mechanisms for the routers to select paths that satisfy these objectives. Most large IP networks run interior gateway protocols (IGPs) such as Open Shortest Path First (OSPF) or Intermediate System-Intermediate System (IS-IS) that select paths based on static link weights. These weights are typically configured by the network operators. Routers use these protocols to exchange link weights and construct a complete view of the topology inside the AS, as shown in Fig. 1. Then each router computes shortest paths (where the length of a path is the sum of the weights on the links) and creates a table that controls the forwarding of each IP packet to the next hop in its route.

On the surface, shortest path routing does not seem flexible enough to support traffic engineering in a network supporting a diverse set of applications. First, these IGPs are limited to routing scenarios that can be specified with a single integer weight on each link. However, we argue that link weights suffice to specify near-optimal routing for large real-world networks. Second, in their basic forms, the OSPF and IS-IS protocols do not adapt the link weights in response to changes in traffic or failures of network elements, and the path selection process does not directly incorporate any performance objectives. Recent standards activity has proposed traffic engineering extensions to OSPF and IS-IS to incorporate traffic load in the link state advertisements and path selection decisions. However, these extensions require modifications to the routers to collect and disseminate the traffic statistics and establish paths based on the load metrics. Instead, we argue that it is often possible to select *static* link weights that are resilient to traffic fluctuations and link failures, allowing the use of the traditional incarnations of OSPF and IS-IS.

The example in Fig. 2 shows how to control the distribution of traffic in a network by tuning the IGP weights. All three diagrams concern the

0163-6804/02/$17.00 © 2002 IEEE

Presentation by: Douglas Chan

Why traffic engineering?

- Self-managing mechanisms that are already in place do not **ensure** networks to run efficiently
  - eg. TCP adjusts sending rate
  - eg. Routers compute new paths to adapt to changing topology
  - But links can still get congested despite availability of underutilized links
  - or still be using routes with high propagation delay
- Need to ensure user performance and efficient use of network resources
  - Adapt the routing of traffic to the prevailing demands
  - At least within your AS or ISP domain

How to traffic engineering?

- Involves these three things:
  - A set of performance objectives
  - Determining the selection of paths
  - An effective mechanism for routers to select path
- Large IP networks run interior gateway protocol (IGP)
  - Eg. Open Shortest Path First (OSPF), Intermediate System-Intermediate System (IS-IS)
  - Select paths based on static link weights
  - Weights also let routers construct complete view of network and forwarding table
    - How about RIP and Cisco’s EIGRP?

Paper’s contributions

- Paper argues: “*often* possible to select *static link weights* that are resilient to traffic fluctuations and link failures, allowing the use of the traditional incarnations of OSPF and IS-IS.”
- Brings together work of various papers that achieve each individual component to traffic engineering

Is it possible?

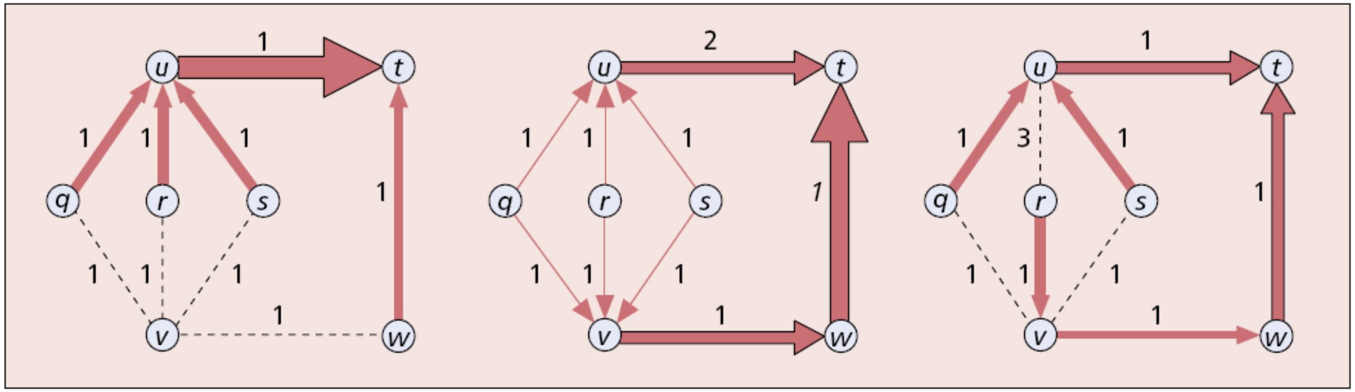
- Shortest path routing not flexible enough for a network supporting diverse applications:
  - Limited to routing scenarios with a single integer weight on each link
  - Does not represent all possible solutions to the routing problem (unlike OPT)
- Paper argues:
  - This is enough to “specify near-optimal routing for large real-world networks”
  - Weights can also be determined by wide variety of costs, performance, and reliability constraints

Is it possible?

- Not adaptable:
  - OSPF and IS-IS by themselves do not adapt the link weights in response to traffic and doesn’t care about performance constraints
  - Standards proposed to incorporate this, but require routers to collect and disseminate statistics to establish these paths
- Paper argues:
  - Can be done even with IGPs through smartly assigning static link weights

Example of controlling traffic via weights

- Goal: Minimize maximum link load



- Unit weight
- Minmax = 3

- “Naïve approach”
- Minimax = 2.5

- Global optimal
- Minimax = 2

- Just by changing link weights can alleviate congestion – attractive alternative to buying BW
- How to solve global optimization problem?

Good and bad of using traditional IGPs

- Set routing parameters by network-wide view of topology and not local views
- Good: Protocol stability
  - Routers do not adapt automatically to locally constructed (potentially out-of-date) views of traffic
  - Predictable and helps diagnose problems
- But.. Link weights configured by external entity
  - Need network management system or human operator to oversee whole network
  - How and can this be done automatically?

Good and bad of using traditional IGPs

- Good: Low protocol overhead
  - Routers do not need to track changes in load and disseminate link state info
  - Lowers BW consumed and computational load
- But…
  - Who tracks these changes then to obtain the network-wide info?
  - How to disseminate new link weights? Still consumes BW (maybe saves very little for smaller networks)

Good and bad of using traditional IGPs

- Good: Diverse performance constraints
  - Routing parameters depend on variety of performance and reliability constraints
  - Can even incorporate constraints that are difficult to formalize in a routing protocol
  - New constraints readily applied
- But…
  - How true is second point?

Good and bad of using traditional IGPs

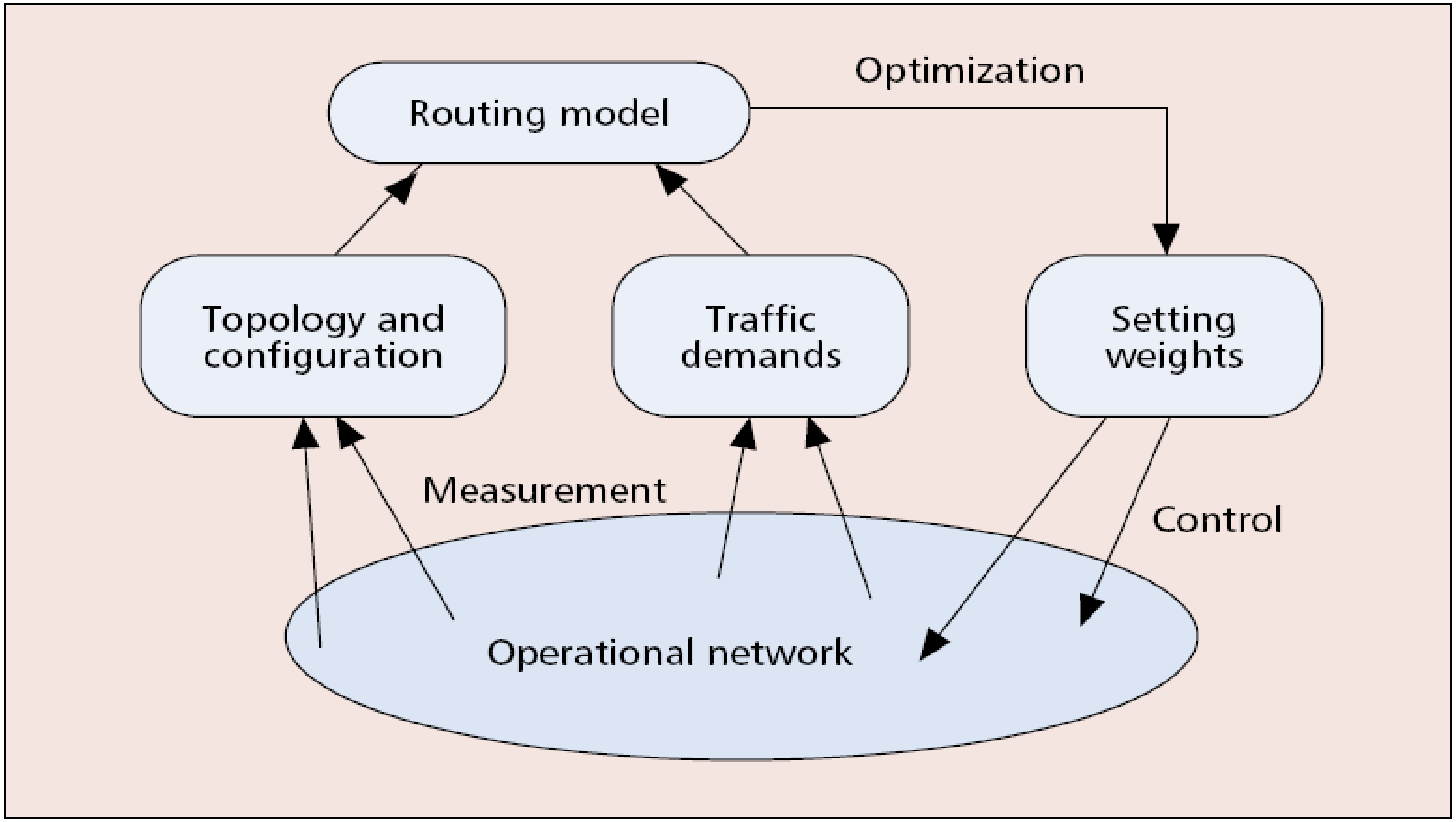
- Good: Compatibility with traditional shortest path IGPs
  - No need to upgrade existing equipment
- Good$^{*BEST!*}$: Link weights are a concise form of configuration state
  - No need for any path-level info or states concerning incident edges to other routers
  - Multiple paths are changed by modifying a single link weights
- But...
  - Need to change weights very carefully

Good and bad of using traditional IGPs

- Good: Default weights based on link capacity are often good enough

- Modification represents significant changes, should be done on relatively coarse timescale
  - But… **Worst!** Does not respond well to transient congestion then?

Traffic engineering framework



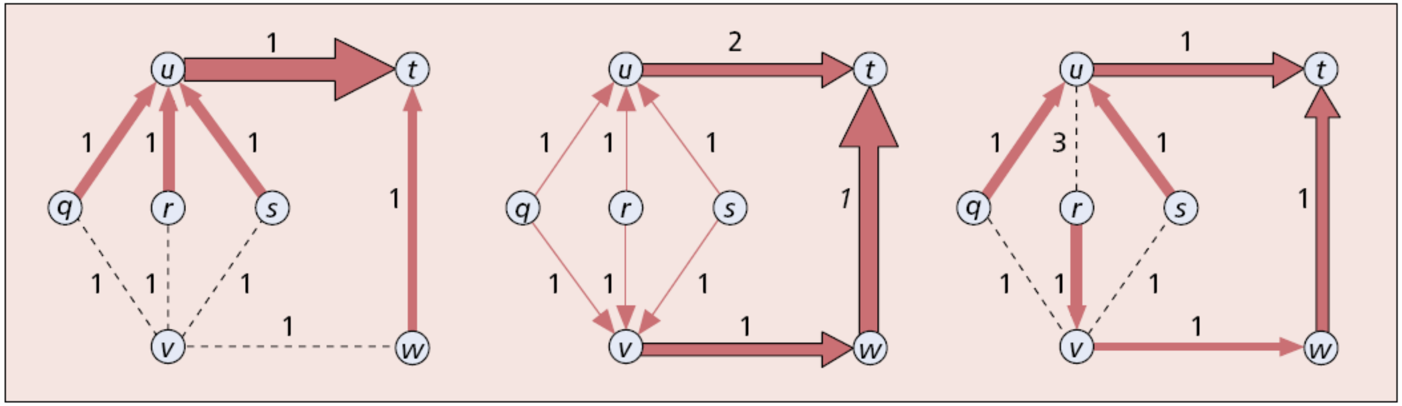
Quantifying performance

- When links have different capacities, better to consider link utilization
  - Ratio of load to capacity
  - A link’s capacity as maximal desirable load
  - Target keep max utilization under 100%
    - To protect bursts, <60%
    - Too low?

Quantifying performance

- Compare against optimal routing (OPT)
  - Direct traffic along any paths in any proportions
  - Models idealized routing scheme that can establish one or more explict paths b/w every pair of nodes
  - Need MPLS protocol
- Compare also simple default configs
  - InvCapOSPF
  - UnitOSPF

Performance with max-utilization



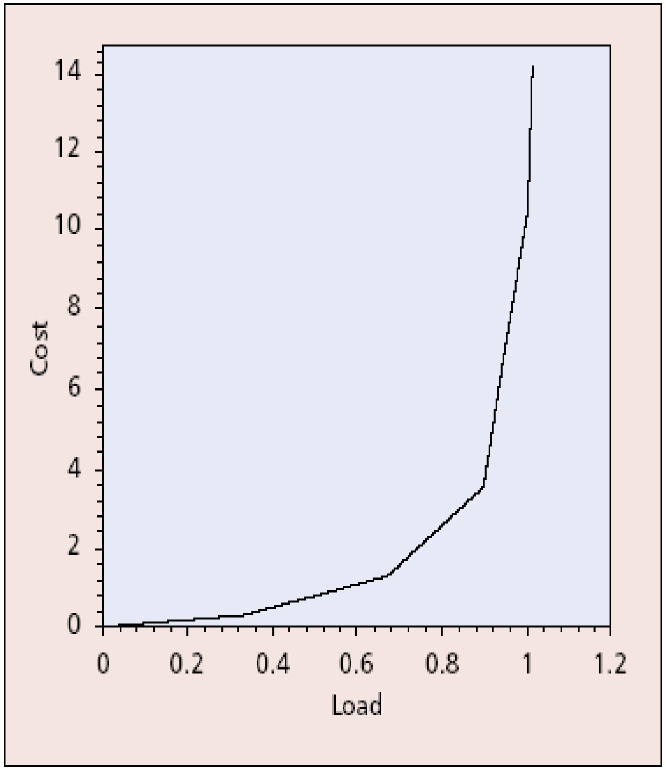
- Setting capacity of links incident to q,r and t to 1 and remaining to 2
- UnitOSPF: max-util = 150%
- Last diagram: 100%
- OPT: 100%

AdvancedOSPF

- In general good weight settings achieve OPT performance within a few percent
- Eg. AdvancedOSPF (3% from OPT on the AT&T network), but UnitOSPF and InvCapOSPF is 50% away
- Attractive alternative to buying extra links
- AdvancedOSPF cannot be improved much more
- Section V of Additional Reading
  - B.Fortz, M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," IEEE Infocom 2000
  - An iterative local search heuristics in a neighborhood that determine a weight vector that minimizes that cost function
  - Used hash tables to avoid repeating neighborhoods during exploration

Performance with a network objective

- Minimizing maximum link utilization maybe overly sensitive to individual bottleneck
  - Eg. An ingress link may always carry large amount of traffic under any solution
- Also does not penalize long paths
- Need to consider a networkwide objective: cost of using a link increases with load
- Networkwide cost of routing is then sum of all link costs



Performance with a network objective

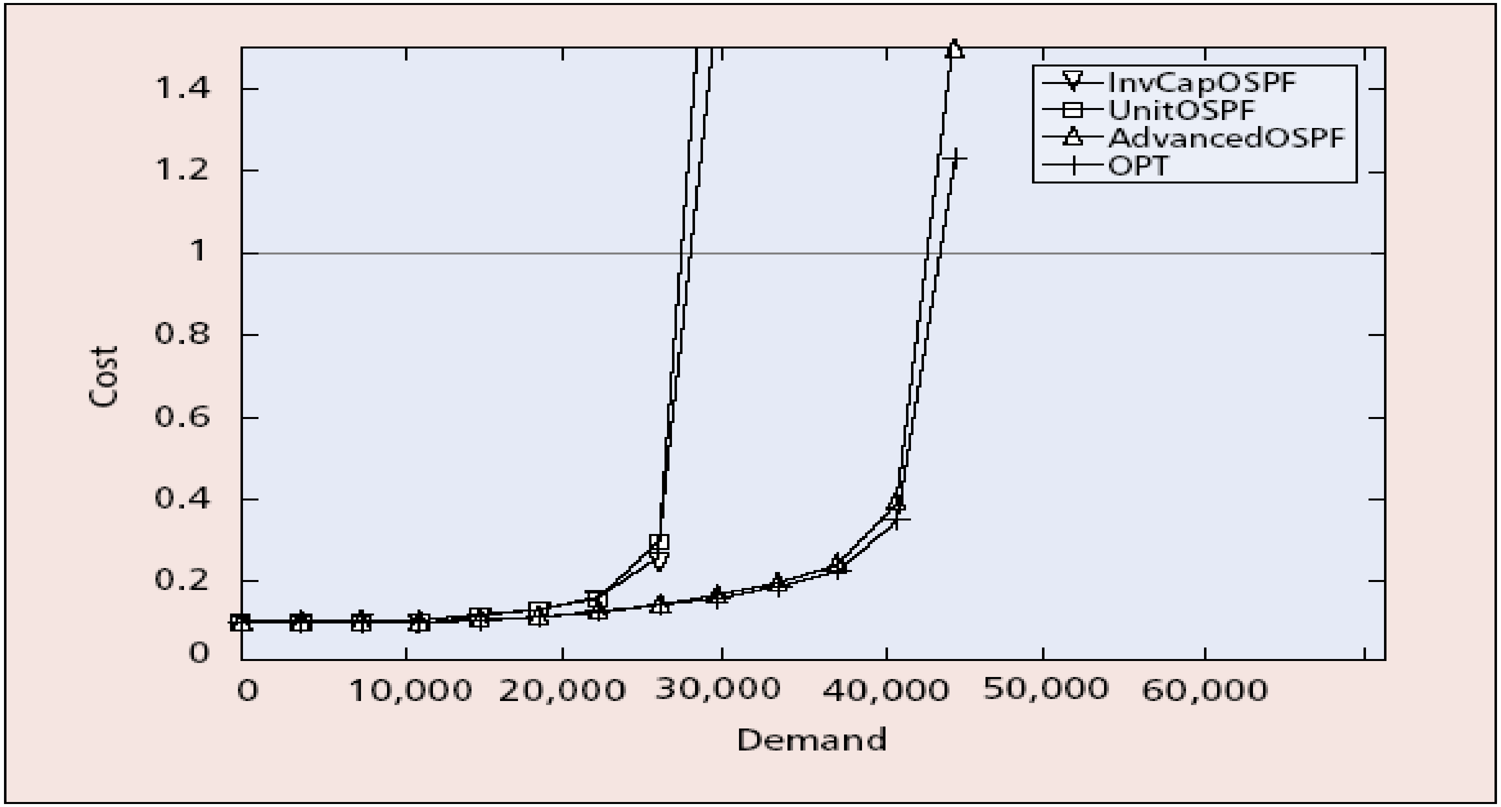


Figure 5. *Networkwide cost vs. demand for a proposed AT&T backbone.*

Performance with a network objective

- Plots networkwide cost normalized to make 1 the threshold for an overloaded network (??)
- AdvancedOSPF handles 70% more than UnitOSPF and InvCapOSPF; and only 2% less than OPT

Network objective vs max link utilization

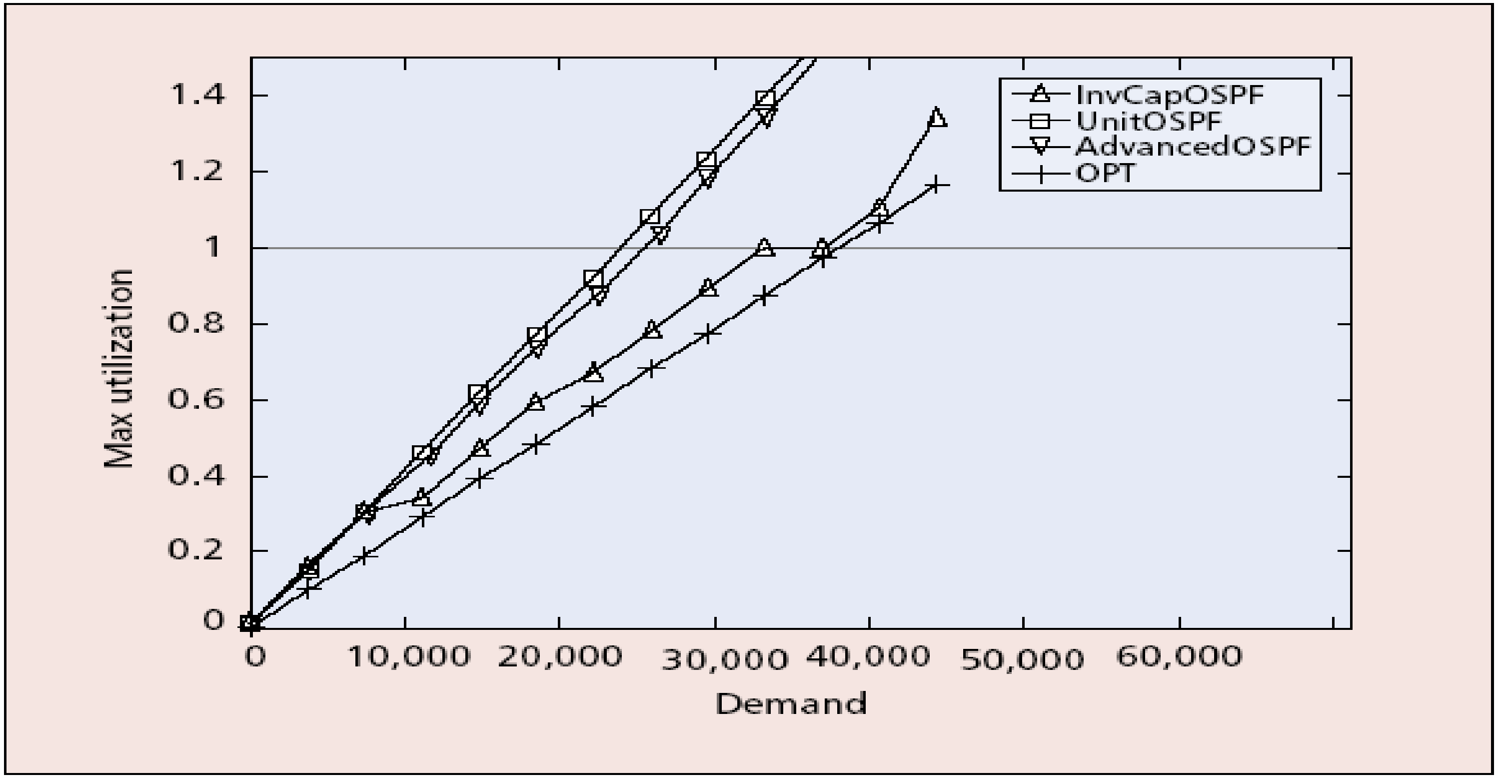


Figure 6. *Maximum link utilization vs. demand with same weights as in Fig. 5.*

Network objective vs max link utilization

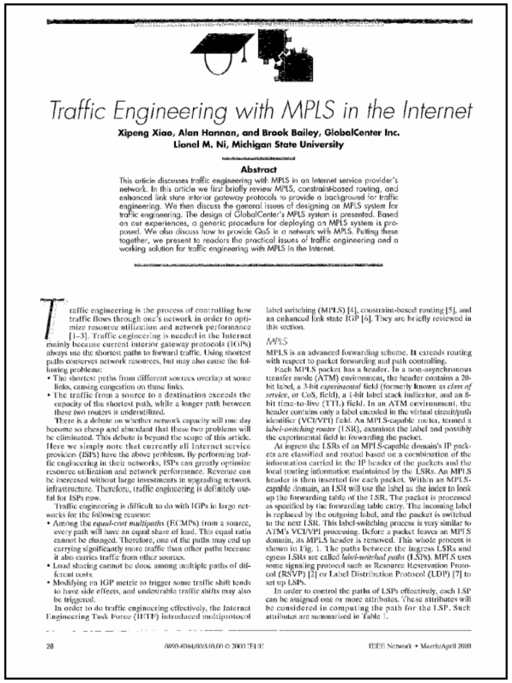
- Illustrates how the weight optimization for link cost function does in terms of max link utilization
- OPT optimal w.r.t. max link utilization
- AdvancedOSPF nears OPT when >100%
  - It avoids the high penalty for >100% utilization (not for really high utilization)
  - It is simultaneously good for both link costs and max utilization
- Good weight settings not very sensitive to exact details of objective function
  - As long as objective function assigns an increasing penalty to links with load approaching capacity

Changing traffic demands

- Test robustness by adding noise
  - Multiply by random number b/w 0 to 2
  - Expected value unchanged, but each changed by 50% on avg (??)
- Same link weights performed well
- Can find optimal weight settings for both day and night
  - Operators don't need to disrupt network
  - Works well for convex combinations of demand matrices (gradual transitions)
- Failure of a few critical links require link weight change; a single weight change enough to reduce congestion

Traffic Engineering with MPLS in the Internet

By: X. Xiao, A. Hannan, B. Bailey, L.M. Ni



Traffic Engineering with MPLS in the Internet

Xipeng Xiao, Alan Hannan, and Brook Bailey, GlobalCenter Inc.
Lionel M. Ni, Michigan State University

Abstract

This article discusses traffic engineering with MPLS in an Internet service provider's network. In this article we first briefly review MPLS, constraint-based routing, and enhanced link state interior gateway protocols to provide a background for traffic engineering. We then discuss the general issues of designing an MPLS system for traffic engineering. The design of GlobalCenter's MPLS system is presented. Based on our experiences, a generic procedure for deploying an MPLS system is proposed. We also discuss how to provide QoS in a network with MPLS. Putting these together, we present to readers the practical issues of traffic engineering and a working solution for traffic engineering with MPLS in the Internet.

Traffic engineering is the process of controlling how traffic flows through one's network in order to optimize resource utilization and network performance [1–3]. Traffic engineering is needed in the Internet mainly because current interior gateway protocols (IGPs) always use the shortest paths to forward traffic. Using shortest paths conserves network resources, but may also cause the following problems:

- The shortest paths from different sources overlap at some links, causing congestion on those links.
- The traffic from a source to a destination exceeds the capacity of the shortest path, while a longer path between these two routers is underutilized.

There is a debate on whether network capacity will one day become so cheap and abundant that these two problems will be eliminated. This debate is beyond the scope of this article. Here we simply note that currently all Internet service providers (ISPs) have the above problems. By performing traffic engineering in their networks, ISPs can greatly optimize resource utilization and network performance. Revenue can be increased without large investments in upgrading network infrastructure. Therefore, traffic engineering is definitely useful for ISPs now.

Traffic engineering is difficult to do with IGPs in large networks for the following reasons:

- Among the *equal-cost multipaths* (ECMPs) from a source, every path will have an equal share of load. This equal ratio cannot be changed. Therefore, one of the paths may end up carrying significantly more traffic than other paths because it also carries traffic from other sources.
- Load sharing cannot be done among multiple paths of different costs.
- Modifying an IGP metric to trigger some traffic shift tends to have side effects, and undesirable traffic shifts may also be triggered.

In order to do traffic engineering effectively, the Internet Engineering Task Force (IETF) introduced multiprotocol label switching (MPLS) [4], constraint-based routing [5], and an enhanced link state IGP [6]. They are briefly reviewed in this section.

MPLS

MPLS is an advanced forwarding scheme. It extends routing with respect to packet forwarding and path controlling.

Each MPLS packet has a header. In a non-asynchronous transfer mode (ATM) environment, the header contains a 20-bit label, a 3-bit *experimental* field (formerly known as *class of service*, or CoS, field), a 1-bit label stack indicator, and an 8-bit time-to-live (TTL) field. In an ATM environment, the header contains only a label encoded in the virtual circuit/path identifier (VCI/VPI) field. An MPLS-capable router, termed a *label-switching router* (LSR), examines the label and possibly the experimental field in forwarding the packet.

At ingress the LSRs of an MPLS-capable domain's IP packets are classified and routed based on a combination of the information carried in the IP header of the packets and the local routing information maintained by the LSRs. An MPLS header is then inserted for each packet. Within an MPLS-capable domain, an LSR will use the label as the index to look up the forwarding table of the LSR. The packet is processed as specified by the forwarding table entry. The incoming label is replaced by the outgoing label, and the packet is switched to the next LSR. This label-switching process is very similar to ATM's VCI/VPI processing. Before a packet leaves an MPLS domain, its MPLS header is removed. This whole process is shown in Fig. 1. The paths between the ingress LSRs and egress LSRs are called *label-switched paths* (LSPs). MPLS uses some signaling protocol such as Resource Reservation Protocol (RSVP) [2] or Label Distribution Protocol (LDP) [7] to set up LSPs.

In order to control the paths of LSPs effectively, each LSP can be assigned one or more attributes. These attributes will be considered in computing the path for the LSP. Such attributes are summarized in Table 1.

0890-8044/00/$10.00 © 2000 IEEE

Presentation by: Douglas Chan

Overview of Paper

- Gave short reviews of
  - MLPS
  - Constraint-based Routing
  - An enhanced IGP
- Discussed general issues with designing and deploying an MPLS system for traffic engineering
  - Through discussing GlobalCenter
- Providing QoS with MPLS
- Their actions are based on their experience
  - Major critique?

Multi-protocol label switching (MPLS)

- An advanced forwarding scheme
- Extends routing with respect to packet forwarding and path controlling
- Terminology: Label Switching Router (LSR) and Label Switched Path (LSP)

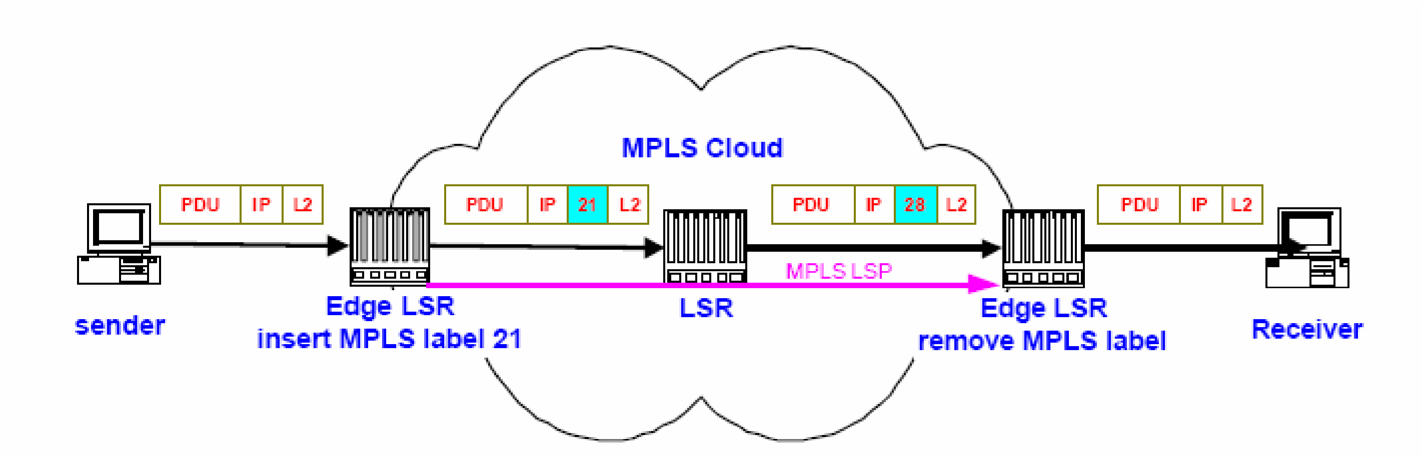


Figure 1. MPLS

Constraint-based Routing

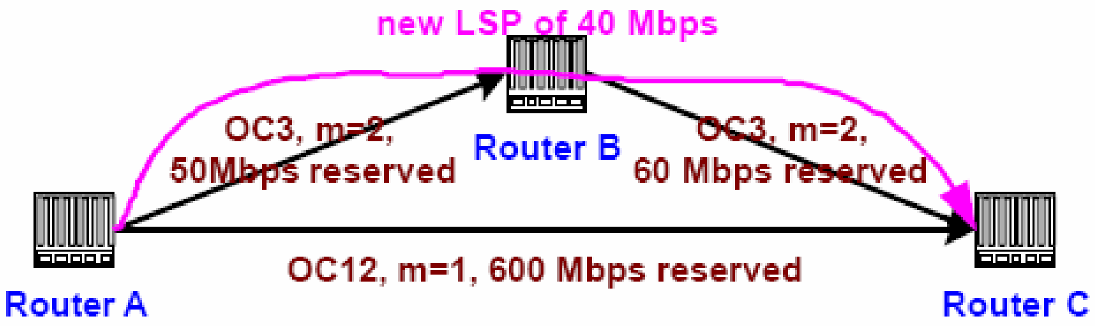


Figure 2. Constraint-based Routing

Overview of ISP

- ISP made of links interconnecting Point-of-presence (POPs)
- Up to 30 POPs arranged symmetrically:
  - Access routers (AR)
    - To customers
  - Border routers (BR)
    - To other ISPs
  - Hosting routers (HR)
    - To Web servers
  - Core routers (CR)
    - To other POPs

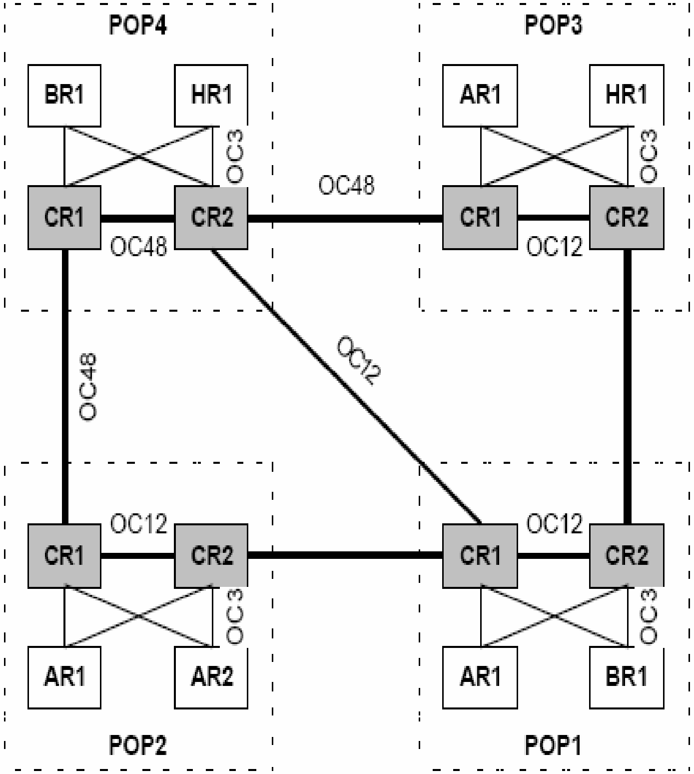
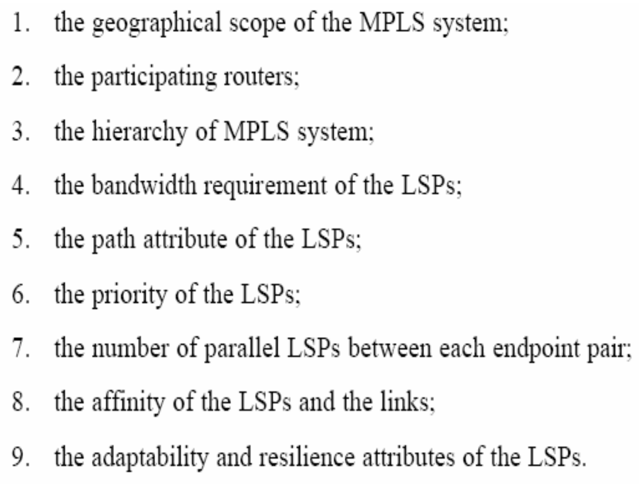


Figure 6. A sample part of an ISP network

Designing MPLS

- Determine design parameters ->
- Decide participating routers in the MPLS system
  - Forbid untrusted and “weak” routers
  - Tradeoff b/w no. of LSPs and efficiency
    - More ingress & egress LSRs
      = more LSPs
      = higher routing complexity
    - But avg. size of LSPs (BW requirement) smaller, Constraint-Based Routing has more flexibility and achieve better link efficiency
- Decide hierarchy: multiple meshed layers of LSPs
  - Reduce processing and managing overhead with smaller LSPs in a layer
- Reoptimization (switching LSPs to better paths that are now available) once per hr, too often may introduce routing instability (??)



1. the geographical scope of the MPLS system;
2. the participating routers;
3. the hierarchy of MPLS system;
4. the bandwidth requirement of the LSPs;
5. the path attribute of the LSPs;
6. the priority of the LSPs;
7. the number of parallel LSPs between each endpoint pair;
8. the affinity of the LSPs and the links;
9. the adaptability and resilience attributes of the LSPs.

GlobeCenter’s US network

- 10th largest ISP in US
  - Anyone use GlobalCrossing?
- 50 POPs of > 300 routers
- 200 routers chosen for MPLS system
  - = ~40,000 LSPs
- 2 layers of LSPs
- 9 regions

Deploying MPLS system

- All based on their experiences

1. Collect statistics using MPLS LSPs
   - Deploy LSPs w/o BW specs
   - Use LSPs to collect traffic statistics
   - So end-to-end traffic is determined

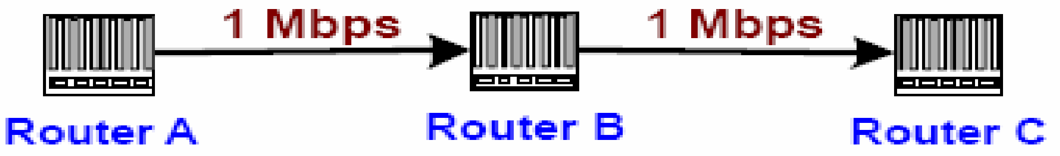


Figure 7. Statistics Collecting

Deploying MPLS system

2. Deploy LSPs with BW constraints
   - Usually use measured rate as BW requirement
     - Use the 95-percentile of all rates over a period
       - Usually close to real peak as opposed to traffic spike
   - Constraint-based routing assign LSPs so max BW of link is >= sum of specified BW of all its LSPs
   - High utilization occurs if actual sun traffic close to link BW
   - Avoid this by:
     - Undersubscribe links, eg. Design to use 60%
     - Inflate BW requirement by factor, eg. Times 1.x
     - Also allows LSP to grow
     - A tradeoff: Too much would result sub-optimal paths, reduce efficiency

- Use sim tools like WANDL first before deployment
  - Relate to Paul's comments about real world vs simulation

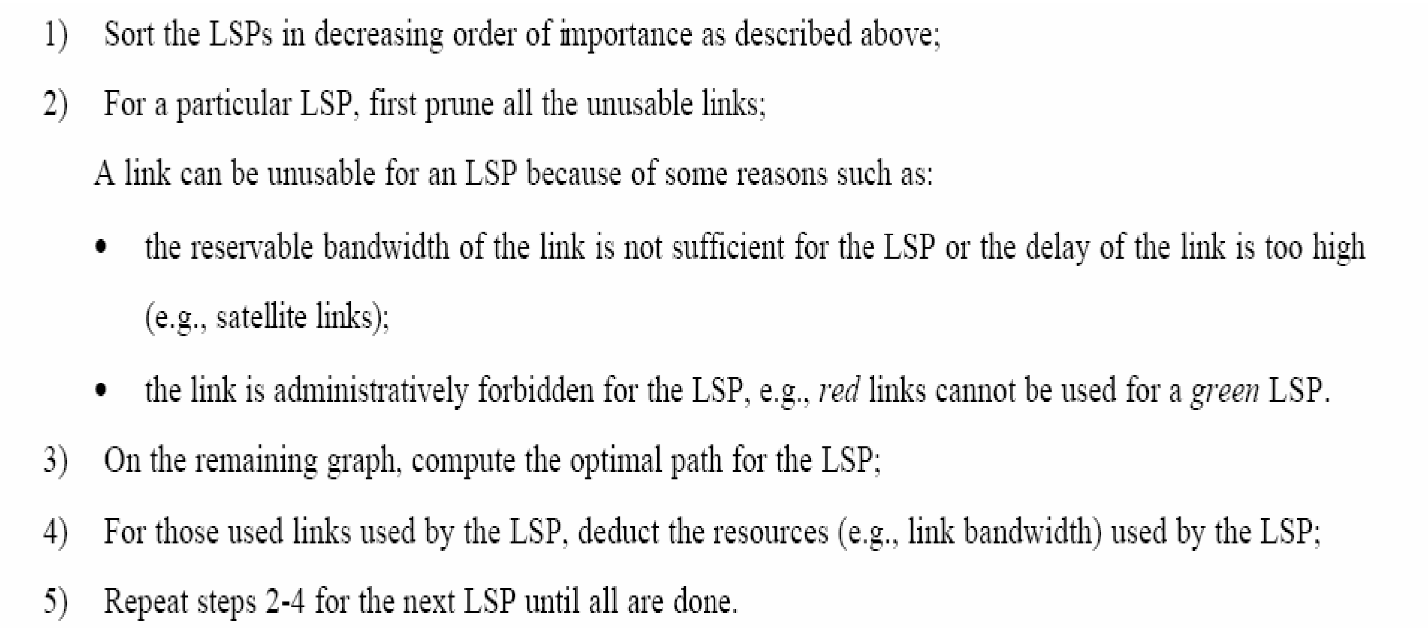
3. Periodic update of LSP BW

Deploying MPLS system

4. Offline Constraint-based Routing

- Online routing less efficient bec every router finds its LSP path
    - Inefficient bec of extra computations??
    - Computed daily – updates too far apart?
- Algorithm:
    - Compute each LSP one by one, in order of 1) priority, 2) BW requirements
    - This optimizes *bandwidth-routing metric*
        - Lest largest LSP takes best path inside each priority class

Offline Constraint-based Routing



1) Sort the LSPs in decreasing order of importance as described above;
2) For a particular LSP, first prune all the unusable links;

A link can be unusable for an LSP because of some reasons such as:

- the reservable bandwidth of the link is not sufficient for the LSP or the delay of the link is too high (e.g., satellite links);
- the link is administratively forbidden for the LSP, e.g., *red* links cannot be used for a *green* LSP.

3) On the remaining graph, compute the optimal path for the LSP;
4) For those used links used by the LSP, deduct the resources (e.g., link bandwidth) used by the LSP;
5) Repeat steps 2-4 for the next LSP until all are done.

- This may not find globally optimal layout for LSPs; but it is simple
- Problem is NP-complete, bec the BIN-PACKING problem can be reduced to it
- Optimal solution not practical except for small network
- Then how does this sol’n compares with optimal??

QoS in MPLS networks

- Use Differentiated Services fields (DS-fields)
- Can route different classes via the different virtual networks formed by the MPLS

- Current LSPs a link in building LSPs for VPN
  - Only endpoints of current LSPs are involved in signaling process of building new LSPs for VPN
  - Reduce state info in the core

Other articles on Traffic Eng. in Special issue

- **Internet traffic engineering** [Guest Editorial]
  Zheng Wang
- **NetScope: traffic engineering for IP networks**
  Feldmann, A.; Greenberg, A.; Lund, C.; Reingold, N.; Rexford, J.
- **Capacity management and routing policies for voice over IP traffic**
  Mishra, P.P.; Saran, H.
- **RATES: a server for MPLS traffic engineering**
  Aukia, P.; Kodialam, M.; Koppol, P.V.N.; Lakshman, T.V.; Sarin, H.; Suter, B.