Some of the note pages contain hypertext links to web pages. You can obtain an html version of this tutorial with the hypertext links by sending an email to the author.
Outline

- IP–based intradomain traffic engineering
  - Traffic engineering
  - Fast restoration

- MPLS–based intradomain traffic engineering

- BGP–based interdomain traffic engineering

- Issues and challenges
Traffic engineering

- **Problem**
  - Shortest path chosen by IP routing does not always produce a good network utilization
  - fish problem
  - How to better optimize the utilization of network ?
  - How to react to changes in traffic conditions ?

For more information on traffic engineering, see

D. Awduche et al., Overview and principles of Internet traffic engineering, RFC3272, May 2002


A special issue of IEEE Network Magazine was focussed on traffic engineering in March/April 2002

The IETF TE working group is focussed on traffic engineering, see : http://www.ietf.org/html.charters/tewg−charter.html
Traffic engineering prerequisite

- To engineer the packet flow in your network...
you first need to know:
  - amount of packets entering your network
    - preferably with some information about their source
      (and destination if you provide a transit service)
  - amount of packets leaving your network
    - preferable with some information about their destination
      (and source if you provide a transit service)

- How to obtain this information in an accurate and cost effective manner?

For a discussion on the types of monitoring or measurements suitable for traffic engineering purposes, see:

Wai Sum Lai et al., A framework for internet traffic engineering measurement, Internet draft, draft-ietf-tewg-measure-02.txt, March 2002

Other references include

An extended version appeared in IEEE/ACM Transactions on Networking


Traffic Matrix Estimation: Existing Techniques and New Directions. A. Medina (Sprint Labs, Boston University), N. Taft (Sprint Labs), K. Salamatian (University of Paris VI), S. Bhattacharyya, C. Diot (Sprint Labs)

See also the papers presented at the ACM SIGCOMM Internet Measurement Workshops and at PAM
Link–level traffic monitoring

- Principle
  - rely on SNMP statistics maintained by each router for each link
  - management station polls each router frequently

- Advantages
  - Simple to use and to deploy
  - Tools can automate data collection/presentation
  - Rough information about network load

- Drawbacks
  - No addressing information
  - Not always easy to know the cause of congestion

A very popular tool for link–level monitoring is MRTG, see http://people.ee.ethz.ch/~oetiker/webtools/mrtg/
Flow–level traffic monitoring

• Principle
  • routers identify flow boundaries
    • does not cause problems on cache–based routers
    • more difficult on hardware–based routers
      • Higher–end routers rely on sampling
  • Layer–3 flows
    • IP packets with same source (resp. destination) prefix
    • IP packets with same source (resp. destination) AS
    • IP packets with same BGP next hop
  • Layer–4 flows
    • one TCP connection corresponds to one flow
    • UDP flows
  • routers forward this information inside special packets to monitoring workstation
Flow level traffic monitoring (2)

- Advantages
  - provides detailed information on the traffic carried out on some links
- Drawbacks
  - flow information exported to monitoring station
    - layer-4 flows
    - layer-3 flows
  - CPU load on high speed on routers
    - not available on some router platforms
    - sampling where one packet out of 100 is processed
  - Disk and processing requirements on workstation that collects monitoring data
    - router-based flow aggregation can sometimes help

Flow–level traffic monitoring tools started with the development of Netflow on Cisco routes (http://www.cisco.com/warp/public/732/Tech/nmp/netflow/). Netflow is available in various formats (V1, V5, V7, V8), depending on the router platform and the desired monitoring information.

Since then, several third-party software have been developed to collect Netflow data. A good list of pointers for such tools is maintained by Simon Leinen at SWITCH (http://www.switch.ch/tf–tant/floma/software.html).

Several vendors have also adopted the Netflow format (http://www.juniper.net/techpubs/software/junos53/swconfig53–policy/html/sampling–config.html).

Within IETF, the IPFIX working group is expected to develop a standard alternative to Netflow. See http://www.ietf.org/html.charters/ipfix–charter.html.
Traffic engineering
simple case study

Traffic matrix
A→F : 0.6
A→B : 0.6
B→F : 0.6
A→E : 0.6

Link load: 120%!
The overlay model

- **Principle**
  - Establish virtual links along the chosen path so that border routers become virtually adjacent
    - \( \frac{N\times(N-1)}{2} \) virtual links inside network
    - traffic between two adjacent routers with flow through their direct virtual link
    - IGP (OSPF/IS-IS) considers routers as adjacent

- virtual links
  - usually ATM or frame relay, IP tunnels, MPLS LSP in some cases

Several methods have been proposed to optimize the layout of the virtual links used in the overlay model. Some references to such methods include:


Traffic engineering
with the overlay model

A’s routing table
F: d=1: A->F
D: d=1: South
B: d=1: A->B
E: d=1: A->E1, A->E2
G: d=2: West, South
C: d=2: South, East

Traffic matrix
A->B : 0.6
A->F : 0.6
B->F : 0.6
A->E : 0.6

B’s routing table
A: d=1: North
C: d=1: South
F: d=1: B->F1, B->F2
E: d=2: South
D: d=2: South
G: d=3: South
IP–based traffic engineering

- How to solve the traffic engineering problem in a pure IP network?
- Two types of solutions

- Network–level traffic engineering
  - Force aggregate traffic flows to follow some paths inside the network
  - Possible in some cases by playing with link costs

- Router–level traffic engineering
  - Allow a router to use several paths instead of a single one for a given route
  - Possible on some router implementations
Equal Cost Multipath

- Simple network-level load balancing mechanism supported in OSPFv2 and ISIS

- Principle
  - OSPF/ISIS distributes the complete network topology to all routers inside network
  - based on this topology, each router computes the routes towards all destinations
  - if a router finds several equal cost paths reaching one destination, it may balance its traffic over these paths
  - load balancing is done at the discretion of this router without coordination with other routers
  - since routes are equal cost routes, loops will not occur provided that the routing table is stable

For a description of OSPF, see J. Moy, OSPF: anatomy of an Internet routing protocol, Addison-Wesley, 1998

ISIS is defined in

R. Callon, Use of OSI IS-IS for Routing in TCP/IP and Dual Environments, RFC1195, Dec. 1990
Equal Cost Multipath (2)

- Example

F's routing table
... E: d=2 : South

D's routing table
... E: d=2 : South–West South–East

G's routing table
... E: d=2 : South–East

A's routing table
F: d=1 : West
D: d=1 : South
B: d=1 : East
G: d=2 : West, South
C: d=2 : South, East
E: d=3 : East, South, West

Traffic flow
A→E : 0.6

B's routing table
... E: d=2 : South

C's routing table
... E: d=1 : South–West

ICNP 2002, Paris, France
© O. Bonaventure, 2002
How to dispatch IP packets?

- **Principle**
  - For each destination, remember the P equal paths instead of a single one
  - Place these paths in forwarding table
  - When a packet arrives, load balancing algorithm selects one path among the P available paths

![Diagram of IP packet dispatch](diagram.png)

- Forwarding decision based on longest match
- Load balancing implies that forwarding table may contain several entries for each destination
Load balancing mechanisms

- Per packet load balancing
  - Send 1st packet on path1, 2nd on path2, ...

- Destination based load balancing
  - Send packet on path = (IP dest) mod P

- Per transport connection load balancing
  - Principle
    - bitstring = [IP src:IP dest:IP protocol:Src port:Dest port]
    - compute path = Hash(bitstring) mod P
      - hash function should be easy to implement and should produce very different numbers for close bitstring values
      - candidate hash functions are CRC, checksum, ...

For an evaluation of the performance of such mechanisms, see:
Zhiruo Cao, Zheng Wang, Ellen Zegura, Performance of Hashing–Based Schemes for Internet Load Balancing, INFOCOM 2000
Limitations of Equal Cost Multipath

Traffic matrix
A→F : 0.6
A→B : 0.6
B→F : 0.6
A→E : 0.6

A’s routing table
F: d=1 : West
D: d=1 : South
B: d=1 : East
G: d=2 : West, South
C: d=2 : South, East
E: d=3 : East, South, West

- Drawbacks of ECM
  - load balancing only works for exactly equal costs paths
  - and few paths are exactly equal in ISP networks
  - local decision taken by each individual router
IP–based network level traffic engineering

- How to improve the traffic distribution throughout the entire network?
- Principle
  - IGP link cost influences the utilization of this link
  - Typical IGP link cost settings include
    - link delay to select shortest path measured in seconds
    - f(bandwidth) to select shortest–high bandwidth path
  - example: \[ M = \frac{1}{\text{link bandwidth}} \]
  - Careful selection of the IGP link costs to balance traffic
    - rerouting traffic outside a busy link by manually tweaking costs
    - optimizing the flow of traffic instead a network for a given traffic matrix can considering it as a classical optimization problem
    - difficult when routers do not support ECM
    - possible with some restrictions when routers support ECM

For example, the default OSPF link cost setting on Cisco routers is described in OSPF Design guide, available from http://www.cisco.com/warp/public/104/1.html

A method to optimally set the OSPF weights for a known traffic matrix was proposed in

Other references include
How to improve the traffic distribution?

- A should send traffic towards E through its South port
- B should send traffic towards F through its South port
IP-based network level traffic engineering (3)

- Possible setting of the IGP link costs

D’s routing table
A: d=2: North
B: d=2: South-East
C: d=1: South-East
E: d=2: S-E, S-W
F: d=2: S-W
G: d=1: S-W

A’s routing table
F: d=3: West
D: d=2: South
B: d=3: East
E: d=4: South
G: d=3: South
C: d=3: South

B’s routing table
A: d=3: North
C: d=1: South
D: d=2: South
E: d=E: South
F: d=4: South
G: d=3: South

C’s routing table
B: d=1: North
A: d=3: North-West
D: d=1: North-West
E: d=1: South-West
F: d=3: N-W, S-W
G: d=2:N−W, S−W

Traffic matrix
A→F: 0.6
A→B: 0.6
B→F: 0.6
A→E: 0.6

ICNP 2002, Paris, France
Outline

- IP–based intradomain traffic engineering
  - Traffic engineering
  - Fast restoration

- MPLS–based intradomain traffic engineering

- BGP–based interdomain traffic engineering

- Issues and challenges
Dealing with link/router failures

- Providing a good service means
  - providing the promised bandwidth/delay ...
  ... even in case of network failures

- How to deal with routers/link failures
  - Detect the failed component
    - Detection by the routing protocol itself (adjacency information)
    - Detection with the help of layer 2 information (carrier lost)
  - Propagate the bad news inside the network
How to detect failures at layer 2?

- Link failure detection heavily depends on the type of layer 2 link and its implementation
  - Packet over SONET/SDH
    - usually, SONET/SDH interfaces are able to detect failures in O(few tens msec)
  - PPP over leased line
    - modem may detect failure or PPP
  - Ethernet link
    - Not designed for quick failure detection, but is changing
- Router must react once informed by layer 2
  - often, routers wait some time before updating link state to avoid flip/flap
  - routers should quickly react to link down signals and could wait longer to react to link up signals
How to detect failures at layer 3?

- Failure detection based on routing protocols

  - OSPF
    - Hello packets sent every 10 seconds
    - Neighbor is dead if no hello received in 40 seconds
  
  - ISIS
    - Default hello interval is set to 10 seconds
    - Three successive hello packets must be unanswered before a failure is detected.

  - RIP
    - Router sends routing table every 30 seconds
    - Neighbor is dead if no table within the 180 seconds

Routing protocols need 10s seconds to detect failures!

Newer implementations of RIPv2 should support triggered updates. With these triggered updates, a bad news (i.e. The metric of a route being set to infinity to indicate a link failure) may be sent immediately. However, once a triggered update has been received, a router should not send another update for the same route within a period of 1 to 5 seconds. This dampening process is used to reduce the network load, but it may also increase the convergence time inside the network.
Architecture of a normal IP router

The "best" paths selected from the routing tables built by the routing protocols are installed in the forwarding table.

Forwarding decision based on longest match
Update of TTL and checksum fields in IP packets

Routing protocol

OSPF table

BGP routing table

IP packets

Forwarding Table

Control

Forwarding

IP packets
IGP convergence: an example

- What happens when R0 fails?

- R4 detects failure of R0
- R4 creates new link state packet (LSP)
  - R4–R0 is removed from R4’s link state packet
  - creation of link state packet is pretty fast
    - but note that to avoid flapping, R4 will not be allowed to create another link state packet before a few seconds
- R4 floods its link state packet to R3 and R7
  - pacing timer to limit bandwidth used by link state packets

A discussion of ISIS convergence, focussed on a single vendor implementation, may be found in:
A. Martel, IS–IS Network Design Solutions, Cisco Press, 2002
See also: Gianluca Iannaccone, Chen–nee Chuah, Richard Mortier, Supratik Bhattacharyya, Christophe Diot. Analysis of link failures in an IP backbone., IMW 2002
By default, on Cisco’s implementation of IS–IS, a delay of at least 5 seconds must elapse between the creation of two successive link state packets. By default, on Cisco’s implementation of IS–IS, the interval between the flooding of two successive link state packets is set to 33 milliseconds, but several parameters to tune the behavior of IS–IS have been introduced recently.
Several groups are working on improving the convergence of IGP protocols. These improvements include the development of algorithms to incrementally update the IGP routing table and tuning of the timers used by a given implement.
IGP convergence: an example (2)

- How does router R4 updates its RIB and FIB?

  - R4 uses ECMP and has two entries to reach R5
    - R4 can disable the R4–R0 path while still using R4–R3–R5
    - No immediate recomputation of the routing table is required and forwarding table can be quickly updated
  - R4 doesn’t use ECMP and has one path towards R5
    - A recomputation of the routing table and an update of the forwarding table on R4 is required

Based on the benefits of using ECMP for load balancing and fast restoration, some ISPs have started to replace the SONET/SDH protected links to connect distant routers by unprotected Packet over SONET/SDH links that travel disjoint paths on distinct fiber ducts.
IGP convergence: an example (3)

- Processing of link state packet at R3
  - Most IGP implementations wait some time before processing a link state packet
    - computation of routing table is CPU intensive and router should avoid doing this computation too often
  - Computation of routing table
    - based on Dijkstra’s algorithm that builds the complete shortest path tree and whose complexity is $O(L \cdot \log(N))$
    - vendors are starting to implement incremental versions of Dijkstra
  - Update of the forwarding table

An interesting study of the performance of OSPF on Cisco routers was published in


Testing on Cisco 7513 and 12012 provided the following measurements:

Processing of link state packet: less than 1 millisecond
Flooding of link state packets: 30–40 milliseconds (function of pacing time)
Dijkstra computation: 25 milliseconds for a 100 nodes network
Update of forwarding table: 100–300 milliseconds (depends on router architecture)

On Cisco’s ISIS implementation, the default waiting time before starting Dijkstra is set to 5.5 seconds, but it can be decreased if required.

Incremental versions of Dijkstra’s algorithm are described in

A good textbook on MPLS is the following


A more practical book on MPLS, centered around Cisco routers is:


The MPLS technology is standardized by IETF, see

http://www.ietf.org/html.charters/mpls−charter.html

Most of the standardization documents on MPLS, including the deprecated ones may be found at

N. Demizu. Multi layer routing. Available from

http://www.watersprings.org/links/mlr/.
The main standardization documents on MPLS are:

Labeled IP packets

- **Generic solution**
  - Insert special 32 bits header in front of IP packet

```
01234567890123456789012345678901
```

- **Technology specific solutions**
  - Reuse the already available "labels" below layer 3
    - Frame Relay
    - Asynchronous Transfer Mode
    - Fiber/lambda switching with special label semantics
    - SONET/SDH with special label semantics

The encoding of the MPLS label is defined in:
E.Rosen, Y.Rekhter, D.Tappan, D.Farinacci, G.Fedorkow, T.Li, and A.Conta.

The utilization of MPLS to support ATM and frame relay switches are discussed in:


Operations performed on labeled packet

- Three types of operations
  - PUSH
    - insert a label in front of a received packet
  - SWAP
    - change the value of the label of a received labeled packet
  - POP
    - remove the label in front of a received labeled packet
How to improve scalability?

- Scalability of LSPs
  - it should be possible to place small LSPs inside large LSPs and preferably recursively
Labeled IP packets (more)

- How to support hierarchy of LSPs?
  - it should be possible to place small LSPs inside large LSPs and preferably recursively without predefined limits on the number of levels supported

- Solution adopted by MPLS
  - each labeled packet can carry a stack of labels

  \[
  \begin{array}{c|c}
  \text{Label} & S \\
  \hline
  01234567890123456789012345678901 & S \\
  \end{array}
  \]

  - label at the top of the stack appears first in packet
  - \( S=1 \) if the label is at the bottom of the stack
  - \( S=0 \) if the label is not at the bottom of the stack
Behavior of edge LSR

- How does edge LSR at ingress determine the label to be used to forward a received packet?

- Principle
  1. Divide the set of all possible packets into several Forwarding Equivalence Classes (FEC)
     - A FEC is a group of IP packets that are forwarded in the same manner (e.g. over the same path, with the same forwarding treatment)
     - examples
       - all packets belonging to same destination prefix
       - all TCP packets sent to same BGP next hop
  2. Associate the same label to all the packets belonging to the same FEC
Behavior of edge LSR (2)

Example

Ra’s Mapping table
IP subnet/prefix        Label
138.48.0.0/16          L1
139.165.0.0/16          L2
12.0.0.0/8              L3
11.0.0.0/8              L2
192.163.13.1            L2

LSP 1
Label: Ingress=Ra, L1, L2, Egress=Rc

LSP 2
- Ingress=Ra, L1, L3, Egress=Rd

LSP 3
- Ingress=Ra, L1, Egress=Rb
Distributing labels

● How to coordinate the label forwarding tables of all LSRs in a given network?

● Use a special protocol to distribute FEC–label mappings
  ♦ LDP : Label Distribution Protocol
  ♦ RSVP–TE : extensions to RSVP

● Piggyback FEC–label mappings inside messages sent by routing protocol
  ♦ possible if routing protocol is extensible
    ♦ BGP can be easily modified to associate label with route
    ♦ RIP cannot be used because its syntax is not extensible
    ♦ link–state protocols (OSPF IS–IS) do not distribute routes

For more information on LDP, see :


The utilization of BGP to distribute MPLS labels is described in :


This label distribution mode is used notably for MPLS/BGP VPNs :


Outline

- IP–based intradomain traffic engineering

- MPLS–based intradomain traffic engineering
  - MPLS architecture
  - Traffic engineering
  - Fast restoration

- BGP–based interdomain traffic engineering

- Issues and challenges
MPLS–based traffic engineering

- Principle of the solution
  - Build a normal IP or IP+MPLS network
    - packet forwarding on shortest path towards destination
  - Collect traffic statistics at edge routers and information about link load
    - identify the most congested parts of the network
    - does MPLS help to collect statistics?
  - Ingress routers establish LSPs along a well chosen path to divert large traffic flows away from heavily loaded links


Example
MPLS in large ISP networks

- Pure IP–based ISP network
  - eBGP on border routers
    - current full BGP routing table: +−120,000 active routes
  - iBGP full mesh
    - 4 border, 3 core routers
    - 24 iBGP sessions

To correctly forward IP packets, border and backbone routers need a full routing table. For this, they need to be part of the iBGP mesh.
MPLS in large ISP networks (2)

- **BGP free ISP backbone**

  **Backbone router**
  - Maintains *internal* routing table of ISP network
  - only knows how to reach routers *inside* ISP

- **Border router**
  - Maintains *full* BGP routing table
  - routing table indicates for each destination
  - AS path towards destination
  - IP address of next-hop to reach destination
MPLS in large ISP networks (3)

- Principle of the solution
  - Use a hierarchy of labels
    - top label is used to reach egress router
    - second label is used to reach eBGP peer

In this case, the MPLS LSPs are established by two distinct protocols. LDP is used to establish the LSP "trees" going from the ingress border routers to each egress border routers. This LSP allows a border router like R1 to send label-swapped packets towards an egress border router like R5. This lower level of labels is used by the core LSRs to forward the packets.

Then, the iBGP sessions between the border routers are used to distribute the upper level of labels. For example R5, when advertising the routes learned from external router RG will indicate label=G, while it would indicate label=H for the routes learned from external router RH. This second level of labels will allow router R5 to rely only on label swapping for all the received packets.
Evaluation of the traffic matrix

- SNMP MIBs can provide useful info
  - Standard traffic engineering MIB
    - Set of tables with inter-tables references
    - Tunnel table for setting up MPLS tunnels.
    - Resource table for setting up the tunnel resources.
    - Tunnel hop table for configuring strict and loose routes
    - Performance table to monitor MPLS tunnels
      - counters: # packets, #bytes, #errors
  - Simple traffic engineering MIB
    - large table with information on all configured LSPs
    - statistics
      - # octets, # packets
    - path information
      - source, destination, route configured and recorded
    - fault tolerance information
      - alternate path, bandwidth, administrative group, ...

See:

C. Srinivasan et al., Multi Protocol Label Switching Traffic Engineering MIB, Internet draft, draft-ietf-mpls-te-mib-08.txt, work in progress, Jan 2002

K. Kompella, A traffic engineering MIB, Internet draft, draft-ietf-tewg-mib-02.txt, work in progress, Feb. 2002
Traffic engineered LSPs

- How to establish a traffic engineered LSP?

**Ingress LSR requires a LSP with**
- 10 Mbps bandwidth
- < 50 msec delay
- no packet losses

**Need information about capabilities of each link**
**Need an algorithm to select the best path according to specific constraints**
**Need signaling protocol to establish LSP**
Constrained routing

- What should be added to traditional routing algorithms?
  - a way to distribute information about current network state
    - routers must know load of remote links to choose paths meeting constraints for flows with QoS guarantees
  - a way to compute a path subject to constraints
    - current routing algorithms find shortest path
    - how can we find a path with
      - minimum hop count
      - at least 10 Mbps
      - at most 10 msec of delay
      - minimum monetary cost
Distributing load information

- Principle
  - Piggyback load/QoS info in OSPF/ISIS packets
- Potential problem
  - Link load information is not distributed immediately
  - Routers must establish flows based on partial information about the current load in the network

1. New flow [B=4] is created between R4 and R6
2. Before information about load changes, R3 wants to create a new flow [B=2] towards R6
- R3 believes that R3–R4–R6 is the best path

For a discussion of those issues, see e.g.:
Constraints

- Three types of constraints on path selection
  - Additive constraint
    - find path minimizing $\sum(d_1, d_2, ..., d_n)$
      - example
        - hop count
        - link delay or cost
      - Multiplicative constraint
        - find path minimizing $\prod(d_1, d_2, ..., d_n)$
          - example
            - loss rate
  - Concave constraint
    - find path containing links whose characteristic is always above a given constraint
      - example
        - bandwidth
        - resource class or color
Finding a constrained path

- Single additive or multiplicative constraints
  - apply Dijkstra’s algorithm

- 2 or more additive/multiplicative constraints
  - unfortunately problem becomes NP hard
  - need to evaluate all possible paths to find exact solution
  - several heuristics have been proposed in literature to find acceptable solutions

The Dijsktra algorithm builds a shortest path tree and is run on each router to determine the shortest path tree from this router to all routers inside the network. This tree is computed incrementally as follows.

First, the tree only contains the router which performs the computation. Then, all the routers that are adjacent to the router performing the computation are considered to be candidate and are put on the candidate list with costs equal to the cost of the links between this router and the candidate router. The candidate router with the smallest cost is added to the shortest path tree and removed from the candidate list. All the neighbors of this router are then examined to see if a better path can be found for a candidate list. The candidate list is updated and the algorithm continues until all routers are added to the tree.

The shortest path tree from R3 is shown below:

The complexity of QoS routing was first analyzed in Z. Wang and J. Crowcroft, Quality of service routing for supporting multimedia applications, IEEE Journal on Selected Areas in Communications, 14(7):1228–1234, 1996.
Finding a constrained path (2)

- **Concave constraints**
  - fortunately easy to handle
    - remove from the network map all links that do not satisfy the constraint
    - utilize Dijkstra's algorithm on the reduced map
  - example
    - find shortest 3 Mbps from R3 to R6

- utilize only links with some kind of protection
Constrained routing in IP networks

- Several solutions proposed by researchers
- Lessons learned
  - Constrained should be applied to flows and not on a per packet basis
  - Currently, constrained routing is only used to establish LSPs
  - Bandwidth and delay are key constraints
    - delay jitter is less important and difficult to efficiently support
  - Path selection should be performed by the source
    - the source of a flow selects an explicit route
    - intermediate nodes perform connection admission control but do not perform any constrained routing decision
    - path selection algorithm does not need to be standardised
    - if the new flow is acceptable, establishment continues otherwise the source will have to compute another path

- Existing constrained routing protocols
- OSPF-TE, ISIS-TE, PNNI (ATM)

ISIS-TE is described in:


PNNI is described in


Additional information about QoS routing protocols may be found in:


OSPF–TE

- Extension to OSPF designed to aid in the establishment of LSPs for traffic engineering
- OSPF–TE distributes new info about each link
  - link type and link Id
  - local and remote IP addresses
  - traffic engineering metric
    - additional metric to specify the cost of this link
  - maximum bandwidth
    - maximum amount of bandwidth useable on this link
  - maximum reservable bandwidth
    - maximum amount of bandwidth that can be reserved by LSPs
  - unreserved bandwidth
    - amount of bandwidth that is not yet reserved by LSPs
  - resource class/color
    - can be used to specify the type of link (e.g. Expensive link would be colored in red and cheap links in green)

OSPF–TE is described in :


Similar extensions have been proposed for ISIS, see


The inclusion of a traffic engineering metric in OSPF–TE and ISIS–TE allows network operators to define two potentially distinct metrics in their network. The traditional metric is used to route normal IP packets and could be based on the link bandwidth for example. The TE metric is used to route LSPs and could be based on the propagation delay for example. Some operators combine these two metrics to establish LSPs, see :

F. Le Faucheur et al., Use of IGP Metric as a second TE Metric , Internet draft, draft–ietf–tewg–te–metric–igp–00.txt , work in progress, March 2002
Using RSVP to distribute MPLS labels

- Principle

- RSVP supports downstream on-demand label allocation
  - RSVP extension for MPLS called RSVP-TE

- Ingress LSR sends PATH message towards egress LSR
  - PATH message includes Label Request Object

- Egress LSR sends back RESV message
  - RESV propagates the labels hop-by-hop

RSVP-TE is defined in the following documents:


Besides RSVP, a second protocol which is an extension to LDP can be used to establish LSPs for traffic engineering purposes:


Note however that the IETF has decided to stop the development of CR−LDP.
Using RSVP to distribute MPLS labels (2)

- LSP establishment with RSVP–TE

PATH contains address of LD

Routing finds next hop=L2

LD knows that LS wishes to establish an LSP

LD allocates a label on link L3-LD

LS

L1

L2

L3

LD

PATH

PATH

PATH

PATH

PATH

PATH

PATH

PATH

PATH

PATH

PATH

RESV[Label=L9]

RESV[Label=L0]

RESV[Label=L7]

RESV[Label=L1]

L1 Label table
West:L9 --> East:L0

L2 Label table
West:L0 --> East:L7

L3 Label table
West:L7 --> East:L1

Packets towards LD will use label L9

LD allocates a label on link L3-LD
RSVP–TE : Explicit Routes

- Principle
  - Ingress LSR may specify the route to be followed by an LSP being established
  
  - Explicit Route specification is a list of
    - IP addresses
    - Subnet prefixes
    - Autonomous System numbers

- Two types of route specifications
  - Strict route
    - the LSP must pass through each LSR specified by ingress LSR
  - Loose route
    - the LSP can pass through non–specified LSR between two specified LSRs
RSVP–TE : Explicit Routes (2)

R3 may select new route to reach R8 and update ERO

PATH [ERO: R3,*,R8]
R7 will forward PATH to R8
PATH [ERO: R4, R7,R8]

PATH [ERO: R1,R3,*,R8]

Record Route Object may be used to record entire route

- In practice
  - Ingress LSR computes strict path based on network topology known thanks to IGP

Note that although the ERO and RRO objects can contain AS numbers and IP prefixes, in practice, they are mainly used with IP addresses.

The establishment of interdomain LSPs is more complex than simply adding AS numbers inside the ERO object. For a discussion on interdomain LSPs, see:

C. Pelsser and O. Bonaventure, RSVP–TE extensions for interdomain LSPs, Internet draft, draft–pelsser–rsvp–te–interdomain–lsp–00.txt, work in progress, October 2002
RSVP–TE for traffic engineered LSPs

Issues to consider

- How to reserve bandwidth for a given LSP?
  - Rely on Tspec and Rspec as with Integrated Services
    - Controlled Load service if bandwidth needs to be reserved
    - Null service if bandwidth need not be reserved

- How to support varying traffic flows?
  - It should be possible to dynamically modify the LSP resources. If there are not enough resources to support an increase, the LSP should keep the old resources

- How to dynamically reroute an established LSP?
  - For example more bandwidth is available on another path or because one link used by the LSP failed
RSVP-TE: Resource increase

- How to smoothly increase the bandwidth of an existing LSP?
- Simple solutions

1. Change resources in PATH and RESV messages
   - if there are not enough resources available in the network to support the bandwidth increase, network will send RESVErr and entire LSP will be removed from network
   - not suitable for important LSP

2. Try to establish new LSP
   - create new LSP and once accepted remove old LSP
   - drawback: the new LSP might be rejected due to the resources already used by the existing LSP
RSVP-TE : Resource increase (2)

- Smooth resource increase

```
PATH[Tunnel=X, Id=Y, Bw=60]
PATH[Tunnel=X, Id=Y, Bw=60]
PATH[Tunnel=X, Id=Z, Bw=80]
PATH[Tunnel=X, Id=Z, Bw=80]
```

`L1`'s state
- Filter Shared Explicit
- Session: Egress: LD, Tunnel Id : X
- LSP Id : Ingress: LS, Id : Y
- upstream=LS
- Bandwidth : 60 Mbps

```
LD 100 Mbps L1 100 Mbps LS

100 Mbps
PATH[Tunnel=X, Id=Y, Bw=60]
PATH[Tunnel=X, Id=Y, Bw=60]
PATH[Tunnel=X, Id=Z, Bw=80]
PATH[Tunnel=X, Id=Z, Bw=80]
```
RSVP-TE: Resource increase (3)

- L1's state
- Filter Shared Explicit
- Session: Egress: LD, Tunnel Id: X
- LSP Id: Ingress: LS, Id: Y
- Session: Egress: LD, Tunnel Id: X
- LSP Id: Ingress: LS, Id: Z
- upstream=LS
- Bandwidth: 80 Mbps

PATHtear[Tunnel=X,Id=Y]

RESV[Bw=80, SE[Tunnel=X,Id=Y;Tunnel=X,Id=Z]]

RESV[Bw=80, SE[Tunnel=X,Id=Y;Tunnel=X,Id=Z]]
RSVP-TE : Changing routes

- How to change the route of an explicitly routed LSP?
- Same principle as for resource increase

Differences between Old_Path and New_Path:
- ERO object
- LSP Id

Old_Path and New_Path share resources on R0–R1 and R1–R3
Old_Path and New_Path share resources on R7–R8
RSVP–TE : Preemption

- **Problem**
  - Some LSPs may be more important than others and should have priority if resources are scarce

- **Solution**
  - Two priorities are specified for each LSP
    - **Setup Priority**
      - priority to steal resources (bandwidth) from existing LSPs
    - **Holding Priority**
      - priority to hold resources (bandwidth) confronted to new LSPs
  
- A new LSP may cause an existing LSP to be removed if $\text{Setup\_Prio(new)} > \text{Holding\_Prio(old)}$
MPLS traffic engineering example

- First step
  - Setup traffic engineered LSPs on shortest path
    - no resources reserved for LSPs
  - Collect statistics on each LSP to determine load

Traffic matrix
A→F : 0.6
A→B : 0.6
B→F : 0.6
A→E : 0.6

Link load : 140%!

A tutorial description of MPLS–based traffic engineering by one of its early proponents may be found in:

Second step
Based on the known bandwidth requirements, compute constrained path and switch LSPs to these new paths with RSVP−TE

Traffic matrix
A→F : 0.6
A→B : 0.6
B→F : 0.6
A→E : 0.6

In this example we assume that a single LSP is used to send traffic from one source to one destination. This is the typical case today.

Note however that some research work has studied the utilization of several LSPs between each source–destination pair and have developed methods to share the load between the various LSPs. See:

However, with the support of Diffserv in an MPLS network, there might be several LSPs between a source–destination pair, with one LSP per type of Diffserv. For a discussion on Diffserv over MPLS, see:


Outline

- IP–based intradomain traffic engineering

- MPLS–based intradomain traffic engineering
  - MPLS architecture
  - Traffic engineering
  - Fast restoration

- BGP–based interdomain traffic engineering

- Issues and challenges
MPLS based fault–tolerance

- Can we achieve faster restoration by relying on MPLS instead of relying only on IP?

- MPLS forwarding does not depend on routing
  - a MPLS router can forward packets even when routing has not converged provided that a secondary LSP exist

- Solution
  - Establish secondary LSPs to protect important LSPs
    - secondary LSP is established and maintained inside the network but carries traffic only in case of failure of the primary LSP
  
  - when an outgoing link or next hop router fails, stop using the primary LSP and switch all traffic to the protection LSP
    - this operation can be done by the MPLS router itself without any cooperation with other routers provided the protection LSP exists

Detecting router failures

- Another solution
  - router periodically transmit heartbeat messages to neighbors
  - neighbor is considered dead if no reply received
- Using RSVP for heartbeat messages
- Optional Hello messages

The RSVP Hello message is described in
(Format: TXT=132264 bytes) (Status: PROPOSED STANDARD)

In this document, the default value for the hello interval is 5 milliseconds. It is interesting to note the default values used by router vendors for this timer:

3 seconds in:
http://www.juniper.net/techpubs/software/junos52/swconfig52−mpls−apps/html/rsvp−cc

3 seconds in:

Relying on RSVP−TE allows to detect link failures and failures of the control plane. Work is ongoing within IETF to develop protocols that allow to detect failures of the MPLS data plane as well, see

K. Kompella et al., Detecting Data Plane Liveliness in MPLS, Internet draft, draft−ietf−mpls−lsp−ping−00.txt, work in progress, March 2002
Selection of path for secondary LSP

- How to select a suitable path for a secondary LSP to protect a primary LSP?

- Principle of the solution
  - Ingress LSR selects path for primary LSP by using its path selection algorithm on the entire network topology
  - Knowing the path of primary LSP, ingress LSR computes a path for the secondary LSP by using its path selection algorithm on the network topology where the resources used by primary LSP have been removed
  - Secondary LSP must rely on different physical resources than primary LSP
  - This implies that the ingress LSR will need to know the physical resources used by each LSP

Several algorithms have been proposed in the literature to select the path of protection LSPs. See:

L. Melon, F. Blanchy and G. Leduc, Decentralized local backup LSP calculation with efficient bandwidth sharing, ITC 2003, Tahiti, Feb 2003

K. Kar, M. Kodialam and T. Lakshman, Routing restorable bandwidth guaranteed connections using maximum–2 route flows, INFOCOM 2002

M. Kodialam and T. Lakshman, Minimum interference routing with applications to MPLS traffic engineering, INFOCOM 2000
Selection of path for secondary LSP (2)

- **Principle of the solution**
  - Identify sets of Shared Risk Link Groups
    - a SRLG identifies set of resources that can fail together
      - all SDH circuits going through the same fiber
      - all SDH circuits going through a single TDM switch
      - routers connected to same power source ..
  - Advertise the SRLGs of each resource with IGP
  - Prune links and routers that are part of same SRLG as primary LSP when computing secondary

In the case shown above, a secondary LSP to protect from a failure of the link between R3 and R5 will have to avoid all the links that are part of SRLG:b .. This implies that the secondary LSP cannot utilize the link between R0 and R5. A possible secondary LSP to protect this link is thus R3–R4–R0–R6–R5.
End-to-End secondary LSP

- **First solution**
  - Secondary LSP established between ingress LSR and egress LSR
  - In case of failure, PathErr message sent to ingress

- **Drawbacks**
  - PathErr may take some time to reach ingress LSR
  - Packets may be lost between failure detection and reception of PathErr
  - One secondary LSP for each primary LSP

Another solution with looped LSPs has been proposed in:

How to improve LSP protection?

- Should be able to protect segments of LSP
- Two types of failures to consider
  - Router failure
    - A router and all the links attached to this router fail
  - Link failure
    - A link fails between two adjacent routers

SRLG protection can be more complex in practice
Per-LSP link failure protection

- Principle
- Protection LSP established by each LSR to protect each link used by primary LSP

Extensions to RSVP–TE have been proposed to allow the ingress LSR to request the establishment of automatic link protection LSPs by each LSR on the path of a primary LSP. See

P. Pan, D. Gan, G. Swallow, J. Vasseur, D. Cooper, A. Atlas, M. Jork, Fast Reroute Extensions to RSVP–TE for LSP Tunnels, Internet draft, draft-ietf-mpls-rsvp-lsp-fastreroute-00.txt, work in progress, Jan 2002

With MPLS, the time to recover from a failure when a protection LSP has been established is a function of:
- the time to detect the failure (layer–2 detection or rsvp–hello)
- the time to update the label table

It can be expected that a restoration time of a few hundreds of milliseconds or less is feasible in operational MPLS networks.
Per-LSP router failure protection

- Principle
  - Protection LSP established by each LSR

### R2's Label forwarding table
Inlabel | Outport | Outlabel
--- | --- | ---
L2 | North−E | L4

### R3's Label forwarding table
Inlabel | Outport | Outlabel
--- | --- | ---
L1 | South−E | L3

### R3's Label forwarding table after failure
Inlabel | Outport | Outlabel
--- | --- | ---
L1 | South−E | L3

### R4's Label forwarding table
Inlabel | Outport | Outlabel
--- | --- | ---
L4 | North−E | L5

### R5's Label forwarding table
Inlabel | Outport | Outlabel
--- | --- | ---
L3 | E | L3

### R5's Label forwarding table
Inlabel | Outport | Outlabel
--- | --- | ---
L3 | South−E | L3

### R6's Label forwarding table
Inlabel | Outport | Outlabel
--- | --- | ---
L3 | NE | L4

### R7's Label forwarding table
Inlabel | Outport | Outlabel
--- | --- | ---
L4 | NW | L4

---

Primary LSP

Protection LSP for R3-R2-R4
Protecting a complete path

- Principle
  - one protection LSP is used to protect each link

- Advantage
  - traffic is immediately switched to secondary LSP

- Drawback
  - a large number of protection LSPs may be required
Protecting a complete path (2)

- **Alternative**
  - establish tree of protection LSPs rooted at egress

- **Advantage**
  - Reduces the number of required LSPs in large networks
  - merging if resources are reserved for protection LSP

- **Drawback**
  - Protection LSPs are longer than with previous solution

The RSVP-TE extensions required to support this time of one-to-one LSP protection are defined in:

P. Pan, D. Gan, G. Swallow, J. Vasseur, D. Cooper, A. Atlas, M. Jork, Fast Reroute Extensions to RSVP-TE for LSP Tunnels, Internet draft, draft-ietf-mpls-rsvp-lsp-fastreroute-00.txt, work in progress, Jan 2002
This type of protection LSP is also called a bypass tunnel

Extensions to RSVP–TE have been proposed to allow the ingress LSR to request the utilization of bypass tunnels by each LSR on the path of a primary LSP to be protected. See

Outline

- IP–based intradomain traffic engineering
- MPLS–based intradomain traffic engineering
- BGP–based interdomain traffic engineering
  - BGP principles
  - BGP–based traffic engineering
  - BGP restoration in case of failures
- Issues and challenges

Classical references on BGP include


A more readable textbook description of BGP may be found in

J. Stewart, BGP4 : interdomain routing in the Internet, Addison Wesley, 1999

A List of the internet drafts related to BGP may be found in


Tim Griffin maintains a long list of references on BGP, see http://www.research.att.com/~griffin/interdomain.html
Intradomain versus Interdomain routing

- **Intradomain routing (IGP)**
  - **Objective**
    - select the best path towards each destination based on some metrics (e.g. Delay, bandwidth) used inside AS
  - **Issues**
    - IGP should react quickly to changes in topology

- **Interdomain routing (EGP)**
  - **Objective**
    - select the best path towards each destination that is compatible with the routing policies of the transit ASes without knowing the topology of the transit ASes
  - **Issues**
    - Each AS can define its own routing policy
    - EGP should be scalable (13,000 Ases, 120,000 routes)

IGP : Interior Gateway Protocol
EGP : Exterior Gateway Protocol
Routing policies

- In theory, ISPs can define various types of routing policies, but two are very popular

- **shared−cost peering**
  - usually used on links between ASes of the same size
  - ASx (ASy) agrees to receive from ASy (ASx) packets sent towards ASx or its direct customers
  - ASx (ASy) does not provide unlimited transit to ASy (ASx)

- **customer−provider peering**
  - ASc is a smaller ISP than ASP
  - ASc buys transit service from ASP
  - ASP agrees to transmit packets from ASc towards any destination
  - ASP agrees to announce the routes received from ASc

Policies used by ISPs are often stored in Internet Routing Registries and expressed in the RPSL language:

C. Alaettinoglu et al., Routing Policy Specification Language (RPSL), RFC2622, June 1999

D. Meyer et al., Using RPSL in practice, RFC2650, August 1999

For ISP policies, see:
http://www.ripe.net/ripencc/pub−services/whois.html
http://www.arin.net/whois/index.html
http://www.apnic.net/apnic−bin/whois.pl

Articles on routing policies include:


The Border Gateway Protocol

- **Objective**
  - Distribute interdomain routes in a scalable manner while supporting routing policies

- **Principles**
  - Distance-vector routing protocol
  - BGP routers exchange routing tables
    - BGP session is established over TCP connection
    - No periodic advertisement of routes as with RIP
      - routes are first advertised when BGP session is established
      - routes are updated when they change
      - routes are removed when they stop being reachable
  - BGP routers use policies to filter and rank the routes sent or received

Classical references on BGP include

Y. Rekhter and T. Li, A Border Gateway Protocol 4 (BGP−4)}, Internet draft, draft−ietf−idr−bgp4−17.txt, work in progress, 2002

A more readable textbook description of BGP may be found in

J. Stewart, BGP4: interdomain routing in the Internet, Addison Wesley, 1999

A List of the internet drafts related to BGP may be found in

E. Chen, Y. Rekther, List of the Current BGP Documents, Internet draft, draft−chen−bgp−reference−03.txt, work in progress, 2002

Tim Griffin maintains a long list of references on BGP, see http://www.research.att.com/~griffin/interdomain.html
The Border Gateway Protocol (2)

- The two variants of BGP
  - eBGP between border routers of distinct AS
  - iBGP between BGP routers inside AS

The iBGP connection between the border routers in one AS is used to distribute to all border routers the interdomain routes received over eBGP sessions. In small networks, these iBGP sessions are established in a full-mesh. In larger networks, this full-mesh is replaced by the utilization of route reflectors or confederations.
The BGP messages

- OPEN
  - update to establish the BGP session between two peering routers
- NOTIFICATION
  - used in case of errors to inform the remote peer of the problem before closing the BGP session
- KEEPALIVE
  - transmitted regularly if there are no other messages
- UPDATE
  - message used to distribute routing information
  - advertisement of IP prefixes
  - withdrawal of IP prefixes
## The BGP UPDATE message

- **Content of each UPDATE message**

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Withdrawn routes</td>
<td>Routes removed</td>
</tr>
<tr>
<td>(variable length)</td>
<td></td>
</tr>
<tr>
<td>Path attributes</td>
<td>Attributes common to all announced routes</td>
</tr>
<tr>
<td>(variable length)</td>
<td></td>
</tr>
<tr>
<td>Network Layer Reachability Info</td>
<td>Routes announced</td>
</tr>
<tr>
<td>(variable length)</td>
<td></td>
</tr>
</tbody>
</table>

- An important point to note is that a BGP router can only announce a single route towards each prefix.

Although a router will only advertise one route towards each destination, a router may utilize several different paths inside its forwarding table to reach this destination.

The BGP attributes

- Attributes found in UPDATE messages
  - ORIGIN
    - how the route was learned (IGP, EGP, Incomplete)
  - AS−PATH
    - the list of AS through which the announcement passed
  - NEXT−HOP
    - the IP address of the router that advertised the route
  - MULTI−EXIT−DISCRIMINATOR
    - can be used for traffic control purposes
  - LOCAL−PREF
    - can be used for traffic control purposes
  - ATOMIC AGGREGATE and AGGREGATOR
    - rarely used
The BGP decision process shown above is a simplified version of the decision process used by router vendors. For more information on the BGP decision process implemented by various vendors, see your favorite router’s documentation or e.g.:


The import policy specifies, for each peer, the routes that can be accepted and those that must be rejected. Furthermore, the import policy may specify a set of changes to the attributes contained in the received routes.

The export policy specifies, for each peer, the routes that can be exported to this peer, possibly after a modification of the attributes attached to the exported routes.
Outline

- IP–based intradomain traffic engineering
- MPLS–based intradomain traffic engineering
- BGP–based interdomain traffic engineering
  - BGP principles
  - BGP–based traffic engineering
  - BGP restoration in case of failures
- Issues and challenges

For a discussion of BGP–based traffic engineering techniques, see :


N. Feamster, J. Borkenhagen, J. Rexford, Controlling the impact of BGP policy changes on IP traffic, AT&T Technical Memorandum, 2001

A practical description of the utilization of such techniques on Cisco routers may be found in :

I. van Beijnum, Building Reliable Networks with the Border Gateway Protocol, O'Reilly, 2002. The traffic engineering chapter of this book is available from http://www.bgpexpert.com
This figure is based on a study of all the interdomain traffic of three distinct ISPs at different periods of time. The trace was collected during one week for BELNET, the Belgian Research ISP, five days for YUCOM, a dialup ISP based in Belgium and one day for PSC, a gigapop in the US. This figure is analyzed in:


A detailed analysis of the characteristics of interdomain traffic based on a stub ISP may be found in:


A similar result concerning the traffic distribution was obtained by studying the traffic of a tier−1 ISP, see

N. Feamster, J. Borkenhagen, J. Rexford, Controlling the impact of BGP policy changes on IP traffic, AT&T Technical Memorandum, 2001
Objectives of interdomain traffic engineering

- **Stub AS**
  - Content–provider
    - Optimize the flow of the outgoing traffic
      - Cost of interdomain links
      - Quality of interdomain routes
      - Load balancing
  - Provider of dialup/broadband service
    - Optimize the flow of the incoming traffic taking into account
      - Cost of interdomain links
      - Load balancing

- **Transit AS**
  - Hot potato routing
  - Cold potato routing
  - Try to have stable interdomain traffic
Control of the outgoing traffic

- Objectives
  - balance traffic among external links
  - prefer some links over others for any reason

- How to achieve this control?
  - AS administrator tunes the BGP decision process of all border routers to influence the selection of the best−path on each router
    - rely on input filters to ignore/change some routes learned
    - use weight to influence the local selection of the best−path on a single router
    - add local−pref to the routes advertised via iBGP inside the AS to influence the other border routers of the AS
    - Rely on IGP to select closest next−hop

Usually, the control of the outgoing traffic is based on a manual configuration of the routers. However, recently some vendors have proposed tools to automate the control of the outgoing traffic based on measurements. See e.g.:

J. Bartlett, Optimizing multi−homed connections, Business Communications Review, January 2002


S. Borthick, Will route control change the Internet, Business Communications Review, September 2002
In this example, R12 has three valid candidate routes to reach 4.0.0.0/8. Our simplified decision process does not allow to select between the two external peers. Real decision process include additional steps to ensure that only one best path is selected. These steps can for example be the age of the route (prefer the oldest which is supposed to be more stable than the youngest) or the IP address of router that announced the router. The selected best path will be advertised by R12 via iBGP towards R11.

In this example, R11 selects the path through R32 because it prefers the path learned via eBGP over the path learned via iBGP.
Control of the outgoing traffic  
local–pref

- Principle
- set local–pref to influence all routers of the AS

In this example, the local–pref attribute inserted by R12 before distributing the route via iBGP will influence the routing decision of all BGP routers inside AS1. This implies that when local–pref is used, some coordination among the border routers of the AS is required.

Some vendors support the weight attribute in addition to local–pref. This attribute is local to the router and not distributed over iBGP or eBGP sessions.
Control of the incoming traffic

- Objectives
  - balance traffic among external links
  - prefer some links over others for any reason

- How to achieve this control?
  - AS administrator needs to send different BGP advertisements on different links to influence the BGP decision process of routers in distant AS

  - But routers in distant AS and transit AS can also tune their outgoing traffic ...
Control of the incoming traffic
Sample network

- Routing without tuning the announcements
  - packet flow towards AS1 will depend on the tuning used on the decision process of AS2, AS3 and AS4

To control the flow of packets that enter a given network, we must act on the announcements that are sent by this network.
Control of the incoming traffic
Selective announcements

- Principle
  - Advertise some prefixes only on some links

- Drawbacks
  - Splitting a prefix increases size of all BGP routing tables
  - No redundancy in case of link failure

In this example, AS1 forces AS3 to send the packets towards 10.0.0.0/8 on the R31–R11 link and the packets towards 11.0.0.0/8 on the R32–R12 link. This is a common method used to balance traffic over external links, but an important drawback is that if the R11–R31 link fails, AS3 would not be able to utilize the R12–R32 link to reach 10.0.0.0/8 and would be forced to use the path through AS2.
Control of the incoming traffic
More specific prefixes

- **Objective**
  - Announce a large prefix on all links for redundancy but prefer some links for parts of this prefix

- **Principle**
  - Since CIDR, when routing a packet, a router will select the *longest match* in its routing table

- Advertise different overlapping routes on all links
  - The entire IP prefix is advertised on all links
  - subnet1 from this IP prefix is also advertised on link1
  - subnet2 from this IP prefix is also advertised on link2
  - ...

© O. Bonaventure, 2002
Control of the incoming traffic
More specific prefixes (2)

- Principle
  - Advertise partially overlapping prefixes

R31’s routing table
- 10/7:AS1 via R11 (eBGP, best but unused)
- 10/7:AS1 via R32 (iBGP)
- 10/8:AS1 via R11 (eBGP,best)
- 11/8:AS1 via R32 (iBGP,best)

R32’s routing table
- 10/7:AS1 via R12 (eBGP, best but unused)
- 10/7:AS1 via R31 (iBGP)
- 10/8:AS1 via R31 (iBGP,best)
- 11/8:AS1 via R12 (eBGP,best)

Compared with the utilization of the selective announcements, the main advantage of using more specific prefixes is that if link R11–R31 fails, then the packets towards 10.0.0.0/8 will still be sent by AS3 through the R32–R12 link since they are part of the 10.0.0.0/7 router learned from R12.

An important drawback of this solution is that it unnecessarily increases the size of the BGP routing tables of all routers on the Internet. For this reason, several ISPs block prefixes that are too long. For example, some ISPs do not accept prefixes longer than /22, and other try to filter prefixes based on the allocation rules of the regional IP address registries.

For more information on this filtering, see:


For more information on the growth of the BGP routing tables, see:

G. Huston, Analyzing the Internet’s BGP routing table, Internet Protocol Journal, 2001, see also http://www.telstra.net/ops/bgp/
AS–Path prepending is a popular technique since in the BGP decision process, the selection of the shortest AS–Path is one of the most important criteria. In theory, the length of the AS–Path is not necessarily an indication of the quality of a path, but some studies have shown that, on average, short AS–Paths offered a better performance that longer paths.

More information on these studies may be found in:

A. Broido et al., Internet expansion : refinement and churn, European Transactions on Telecommunications, special issue on traffic engineering, January 2002

Due to the importance of the "shortest AS–Path" criteria in the BGP decision process, most interdomain routes used in the Internet are relatively short (up to 3–4 transit AS between source and destination for most routes).

See http://ipmon.sprintlabs.com/paccess/routestat/trends.php?type=addrReachability for some information on the addresses that are reachable at N AS hops from a large ISP like Sprint.
Control of the incoming traffic
AS–Path prepending (2)

- AS–Path prepending can be combined with more specific prefixes

R31’s routing table
- 10/8:AS1 via R11 (eBGP, best)
- 10/8:AS1:AS1:AS1 via R32 (iBGP)
- 11/8:AS1:AS1:AS1 via R11 (eBGP)
- 11/8:AS1 via R32 (iBGP, best)

R32’s routing table
- 10/8:AS1 via R31 (iBGP, best)
- 10/8:AS1:AS1:AS1 via R12 (eBGP)
- 11/8:AS1:AS1:AS1 via R31 (iBGP)
- 11/8:AS1 via R12 (eBGP, best)
Control of the incoming traffic
Multi Exit Discriminator

Example

R31’s routing table
- 10/8:AS1 via R11 (eBGP, MED=0, best)
- 10/8:AS1 via R32 (iBGP, MED=1)
- 11/8:AS1 via R11 (eBGP, MED=1)
- 11/8:AS1 via R32 (iBGP, MED=0, best)

R32’s routing table
- 10/8:AS1 via R12 (eBGP, MED=1)
- 10/8:AS1 via R31 (iBGP, MED=0, best)
- 11/8:AS1 via R12 (eBGP, MED=0, best)
- 11/8:AS1 via R31 (iBGP, MED=1)

Although the MED can be set to any value, a common method is to set the MED equal to the IGP cost towards the next-hop. This is the method used in this example, where we assumed unitary link costs.

An issue with the MED attribute is that if the link cost inside an AS changes frequently due to flapping, this flapping may be exported to peer AS.

Another issue with the MED is that its utilization may create problems in some cases, see:

Control of the incoming traffic
Summary

- Advantages and drawbacks
  - Selective announcements
    - always work, but if one prefix is advertised on a single link, it may become unreachable in case of failure
  - More specific prefixes
    - better than selective announcements in case of failure
    - but increases significantly the size of all BGP tables
    - some ISPs filter announcements for long prefixes
  - AS–Path prepending
    - Useful for backup link, but besides that, the only method to find the required prepending is trial and error...
  - Multi–Exit–Discriminator
    - only useful between direct peers
    - usually requires an agreement before being useable
    - May cause routing loops in some conditions
The BGP Communities attribute

- **Principle**
  - Transitive attribute containing a set of communities
  - each community acts as a marker
    - one community is represented as a 32 bits value
    - usually routes with same marker are treated same manner

- **Standardized communities**
  - NO_EXPORT (0xFFFFFFFF01)
  - NO_ADVERTISE (0xFFFFFFFF02)

- **Delegated communities**
  - 65536 communities have been delegated to each AS
    - ASX65536 ASX:0 through ASX:65535

The BGP community attribute is described in

R. Chandra et al., BGP Communities attribute, RFC 1997, August 1996

A first discussion of the utilization of this attribute may be found in :


A detailed survey of the utilization of the community attribute today may be found in :


A new extended BGP community attribute offering a more structure space with 8 bytes per extended communities is being defined within IETF :

Utilization of BGP communities

- Communities used for tagging
  - Community attached by router that receives route to indicate where route was received
  - Example (Eunet, AS286)
    - 286:1000 + countrycode for Public peer routes
    - 286:2000 + countrycode for Private peer routes
    - 286:3000 + countrycode for customer routes

- Communities used for signalling
  - Community attached by router that sends route to request upstream to perform some actions
  - Example (UUNet, AS702)
    - 702:80 : Set Local Pref 80 within AS702
    - 702:120 : Set Local Pref 120 within AS702
The BGP redistribution communities

- **Drawbacks of community-based TE**
  - Requires error-prone manual configurations
  - BGP communities are transitive and thus pollute BGP routing tables

- **Proposed solution**
  - Utilize extended communities to encode TE actions in a structured and standardized way
  - actions
    - do not announce attached route to specified peer(s)
    - attach NO_EXPORT when announcing route to specified peer(s)
    - prepend N times when announcing attached route to specified peer(s)

The BGP redistribution communities are described in:

O. Bonaventure et al., Controlling the redistribution of BGP routes
Internet draft, draft-ietf-ptomaine-redistribution-01.txt, work in progress, August 2002

An implementation of these communities in zebra is described in:

Community-based selective announcements

- R22 does not announce 10/7 to R41
- R41 will only know one path towards 10/7
Community–based
AS–Path prepending

- R22 announces 10/7 differently to R32 and R41
- R41 will prefer path via R32 to reach 10/7
Interdomain traffic control
Summary

- Control of the outgoing traffic
  - AS has almost full control on this traffic
  - ... but it depends on the received announcements

- Control of the incoming traffic
  - Requires tuning of the announcements ...
  - ... never perfect since each AS can use localpref...

- Issues to take into account
  - Internet is composed of about 13.000 AS
    - size and stability of BGP routing tables are key concern
  - Avoid changes that affect unnecessarily everyone
  - Avoid too rapid changes that could affect stability
Outline

- IP–based intradomain traffic engineering
- MPLS–based intradomain traffic engineering
- BGP–based interdomain traffic engineering
  - BGP principles
  - BGP–based traffic engineering
  - BGP restoration in case of failures
- Issues and challenges
BGP restoration in case of failures

- How to detect the failed link?
  - Layer 2 detection can be fast
  - BGP-based detection relies on Keep-Alive
    - Default keep-alive period: 30 seconds

- How quickly can BGP find alternate route?
  - Measurements done in 1998–99 by injecting BGP advertisements indicating failures on the Internet
    - Tdown event needs more than 2 minutes to be propagated
      - One event can trigger multiple BGP messages
    - Tup event needs about one minute to be propagated
      - One event can trigger multiple BGP messages
    - Average restoration time: three minutes
    - Failure may trigger oscillations lasting fifteen minutes
The reasons for the slow convergence

- The BGP protocol itself

Routers will process the withdraw and advertise alternate routes

This example is based on:

The reasons for the slow convergence (2)

- C sends announcements

Routing table of C
R via A (Path: A−R)
R via B (Path: B−R) (best)

Routing table of A
R via B (Path: B−R) (best)
R via C (Path: C−R)

Routing table of B
R via A (Path: A−R)
R via C (Path: C−R) (best)

- A learns a worse (but valid) route towards R
- B learns that the route via C is a loop

It should be noted that some BGP implementations support sender-side loop detection (aka split horizon). In this case, router C would not advertise to B a path containing B inside its AS−Path. However, this feature is not compulsory in BGP.
The reasons for the slow convergence (3)

- B sends announcements

- C learns a longer (but valid) path towards R
- A learns that the route via B is a loop
The reasons for the slow convergence (4)

- A sends announcements

Routing table of C
R via A (Path: A−R) (best)

Update : R Path: A C B R

Routing table of A
R via C (Path: C−B−R) (best)
- C learns that route via A is a loop
- C will withdraw its route and inform A
- B learns that route via A is a loop

Routing table of B
R via A (Path: A−R) (best)

In practice, the convergence of BGP will also depend on the MRAI timer and on the utilization of the BGP route dampening features.

For a more detailed discussion of those issues, see:


Route Flap Damping Exacerbates Internet Routing Convergence. Zhuoqing Morley, Mao (UC Berkeley), Ramesh Govindan (ICSI), George Varghese(UCSD) and Randy Katz (UC Berkeley), SIGCOMM 2002


RIPE has installed beacons to study the convergence of BGP in the global Internet, as part of their RIS project see:
http://www.ripe.net/ris/beacon.html
Outline

- IP–based intradomain traffic engineering
- MPLS–based intradomain traffic engineering
- BGP–based interdomain traffic engineering

---

Issues and challenges
IP–based intradomain traffic engineering

- **Solved problems**
  - Flow–based and link–based link monitoring
  - Load balancing at router level
  - Computation of the IGP link weights based on a known traffic matrix for best–effort traffic

- **Development issues**
  - Improving the convergence of IGP
    - IGP implementation needs to be redesigned with more focus on fast convergence without losing scalability
    - Target could be around 100 milliseconds
  - Network design guidelines to improve convergence
  - Better management tools
    - Synchronous update of costs on all routers of a network
IP–based intradomain traffic engineering (2)

- Research issues
  - The traffic matrix
    - Techniques to efficiently estimate the traffic matrix
    - More measurements to determine its variability
  - Computation of metrics taking into account possible link failures and QoS
    - Traffic should still be engineered after one failure
    - Benefits of using a second IGP metric for VoIP?
  - Interactions between intradomain and interdomain traffic engineering for large ISPs offering transit service
    - A small change in BGP routes might move lots of traffic

The following paper studies how a setting of the link weights behaves in cases of failures or changes in the traffic matrix:


A first step in the setting of OSPF/IS–IS weights to survive failures is described in:


Several papers have been published recently on the evaluation of the traffic matrix, see:

Traffic Matrix Estimation: Existing Techniques and New Directions. A. Medina (Sprint Labs, Boston University), N. Taft (Sprint Labs), K. Salamatian (University of Paris VI), S. Bhattacharyya, C. Diot (Sprint Labs), SIGCOMM 2002.
MPLS–based intradomain traffic engineering

- Development issues
  - Operational experience in managing traffic engineering in MPLS–based networks
    - Currently more complex than operating an IP network
  - Tuning of the reaction in case of link failures
  - Support of traffic engineering across IGP areas

- Research issues
  - Algorithms to setup LSPs across IGP areas with
    - Bandwidth constraints and QoS constraints
    - Fast restoration constraints (primary and secondary)
  - Methods to map traffic onto LSPs
  - Methods to reduce the number of LSP required
    - Scalability issues with full–mesh of LSPs in large ISPs

For discussions of operational issues, see the presentations at the recent NANOG meetings, http://www.nanog.org
Interdomain traffic engineering

- **Research issues**
  - Can we improve the convergence of BGP?
    - Can we improve the convergence of interdomain routing by replacing BGP with a different protocol?
  - **Stability of BGP with interdomain TE**
    - What would happen if each AS started to dynamically change its BGP advertisements to engineer its traffic?
    - Can we design stable interdomain TE methods?
  - **Interactions between intradomain and interdomain traffic engineering**
  - Could MPLS play a role across interdomain boundaries?
    - MPLS could provide a better control on the traffic flow and a faster convergence in case of failure

The IRTF has started a working group on routing where the development of new protocols is briefly discussed, see http://www.irtf.org/charters/routing.html

Several vendors have implemented devices that allow to control interdomain routes. See

Thank you

Questions and comments can be sent to

Olivier Bonaventure

Department of Computing Science and Engineering
Université catholique de Louvain (UCL)
Place Sainte–Barbe, 2, B–1348, Louvain–la–Neuve (Belgium)

Email: Bonaventure@info.ucl.ac.be
URL: http://www.info.ucl.ac.be/people/OBO

This half–day tutorial is largely based on a two–days tutorial on "Traffic control and QoS in IP networks" that was given several times in Belgium and in France during the last few years. To obtain more information about those previous tutorials (some have been recorded and are available on CD–ROM or on the web) or to be informed about updates to this tutorial, send an email to Bonaventure@info.ucl.ac.be.