3. Advanced Routing Mechanisms

PA159: Net-Centric Computing I.

Eva Hladká

Slides by: Tomáš Rebok

Faculty of Informatics Masaryk University

Autumn 2014

Lecture Overview I

1. Routing: Recapitulation
   - Distributed Routing
   - Autonomous Systems
2. Distance Vector Routing Protocols
   - RIP protocol
   - IGRP protocol
   - EIGRP protocol
   - Comparison
3. Link State Routing Protocols
   - OSPF Protocol
   - IS-IS Protocol
4. Path Vector Routing Protocols
   - BGP Protocol
5. Router Architectures
   - Router Introduction
   - IP Address Lookup Algorithms
   - IP Packet Filtering and Classification

Lecture Overview II

6. Traffic Engineering in IP Networks
   - Introduction
   - Discovering Network Utilization
   - Discovering Network Topology
   - Links' Weights Computation

7. Multiprotocol Label Switching
   - MPLS
   - Generalized MPLS
   - Grid-enabled GMPLS

8. QoS-Based Routing

9. Advanced Routing Mechanisms: Literature

Lecture Overview I

Lecture Overview II

6. Traffic Engineering in IP Networks
   - Introduction
   - Discovering Network Utilization
   - Discovering Network Topology
   - Links' Weights Computation

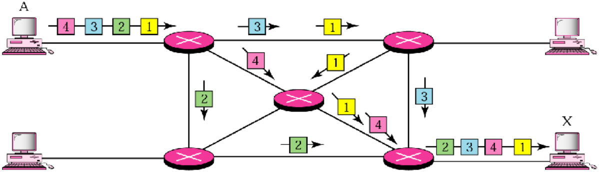
7. Multiprotocol Label Switching
   - MPLS
   - Generalized MPLS
   - Grid-enabled GMPLS

8. QoS-Based Routing

9. Advanced Routing Mechanisms: Literature

Routing in General

- Internet on the L3 – datagram apporach to packet switching
  - upper layer data are encapsulated into datagrams
  - datagrams (their fragments) travel through the network independently on each other
  - the global knowledge of the network's topology is problematic



- **Routing** = the process of finding a path in the network between two communicating nodes
  - the route/path has to satisfy certain constraints
  - influenced by several factors:
    - *static ones:* network topology
    - *dynamic ones:* network load

A Real Network Example

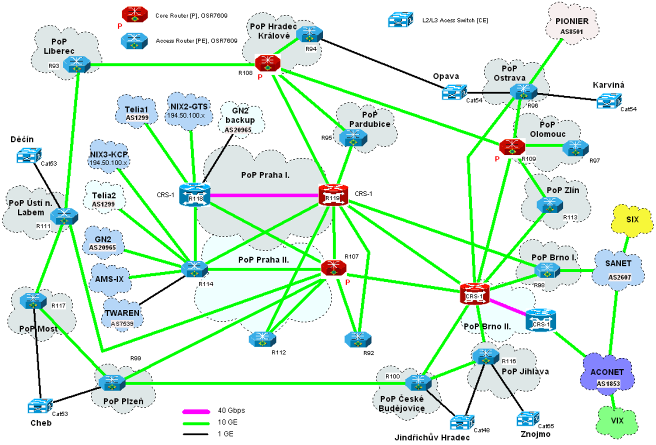


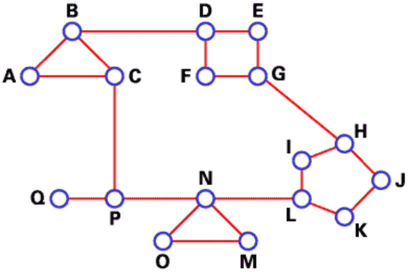
Figure: The topology of the IP/MPLS layer of the CESNET2 network.

Routing – the goal

- the main goal of routing is:
    - to find optimal paths
        - the optimality criterion is a *metric* – a cost assigned for passing through a network
    - to deliver a data packet to its receiver
- the routing *usually* does not deal with the whole packet path
    - the router deals with just a single step – to whom should be the particular packet forwarded
        - somebody "closer" to the recipient
        - so-called *hop-by-hop* principle
    - the next router then decides, what to further do with the received packet

Routing – Mathematical View

- the routing can be seen as a problem of graph theory
- a network can be represented by a graph, where:
  - nodes represent routers (identified by their IP addresses)
  - edges represent routers' interconnection (a data link)
  - edges' value = the communication cost
    - based on the employer metric – hop count, links' delay, links' usage, etc.
  - *the goal:* to find paths having minimal costs between any two nodes in the network



Routing – Mathematical View

Graph Theory Algorithms

Two very important algorithms have profound impact on data networks: **Bellman-Ford algorithm** and **Dijkstra's algorithm**

- both allow to compute shortest paths from a single source
  - to a single destination – Bellman-Ford, complexity $O(LN)$
  - to all the destinations – Dijkstra, complexity $O(N^2)$ (can be improved to $O(L + NlogN)$)
- both of them have *centralized* and *distributed* variants
- variants for *widest-path computation* also exist
  - so-called *widest-path routing algorithms*
    - algorithms, that use a *non-additive concave property* to define distance cost between two nodes
    - e.g., bandwidth – the bandwidth of a path is determined by the link with the minimum available bandwidth
    - i.e., if $m(P) = min\{m(n_1, n_2), m(n_2, n_3), ..., m(n_i, n_j)\} \Rightarrow$ *concave property*
- further details:
  - PB165: Graphs and networks (prof. Matyska, doc. Hladká, doc. Rudová)

Routing – basic approaches

distributed vs. centralized

hop-by-hop vs. source-based

deterministic vs. stochastic

single-path vs. multi-path

dynamic path selection vs. static path selection

INTERNET

Distributed Routing – Basic Approaches

Basic approaches to distributed routing:

- *Distance Vector (DV)* – Bellman-Ford algorithm
  - the neighboring routers periodically (or when the topology changes) exchange complete copies of their routing tables
  - based on the content of received updates, a router updates its information and increments its *distance vector number*
    - a metric indicating the number of hops in the network
  - i.e., *"all pieces of information about the network just to my neighbors"*
- *Link State (LS)* – Dijkstra's algorithm
  - the routers periodically exchange information about states of the links, to which they are directly connected
  - they maintain complete information about the network topology – every router is aware of all the other routers in the network
  - once acquired, the Dijkstra algorithm is used for shortest paths computation
  - i.e., *"information about just my neighbors to everyone"*

Distributed Routing – Link State vs. Distance Vector

Link State

- *Complexity:*
  - every node has to know the cost of every link in the network ⇒ $O(nE)$ messages
  - once a link state changes, the change has to be propagated to *every* node
- *Speed of convergence:*
  - $O(n^2)$ alg., sends $O(nE)$ messages
  - sustains from oscillations
- *Robustness:*
  - wrongly functional/compromised router spreads wrong information just about the links it is directly connected to
  - every router computes routing tables on its own ⇒ separated from routing information propagation ⇒ a form of robustness
- *Usage:*
  - suitable for large networks

Distance Vector

- *Complexity:*
  - once a link state changes, the change has to be propagated just to the *closest neighbors*; it is further propagated just in cases, when the changed state leads to a change in the current shortest paths tree
- *Speed of convergence:*
  - may converge more slowly than LS
  - problems with routing loops/cycles, *count-to-infinity* problem
- *Robustness:*
  - bad computation is spread through the network ⇒ may lead to a “confusion” of other routers (bad routing tables)
- *Usage:*
  - suitable just for smaller networks

Distributed Routing – Path Vector

Path Vector (PV)

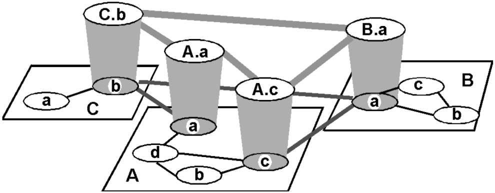
- a variant of DV routing
- in comparison with the DV, whole paths are sent in the PV (not only the end nodes)
  - allows a simple detection of loops
  - allows a definition of rules/policies (friendly vs. non-friendly ASs)

Autonomous Systems

- the goal of Internet's division into *Autonomous Systems* is:
  - a reduction of routing overhead
    - simpler routing tables, a reduction of exchanged information, etc.
  - a simplification of the whole network management
    - particular internets are managed by various institutions/organizations
- autonomous systems = domains
  - a 16bit identifier is assigned to every AS/domain
    - *Autonomous System Number (ASN)* – RFC 1930
    - assigned by *ICANN (Internet Corporation For Assigned Names and Numbers)*
  - correspond to administrative domains
    - networks and routers inside a single AS are managed by a single organization/institution
    - e.g., CESNET, PASNET, ...
  - a distinction according to the way an AS is connected to the Internet:
    - *Stub AS*
    - *Multihomed AS*
    - *Transit AS*

Autonomous Systems – routing

- separated routing because of scalability reasons:
  - *interior routing*
    - routing inside an AS
    - under the full control of AS's administrator(s)
    - the primary goal is the performance
    - so-called *Interior Gateway Protocols (IGP)* (e.g., RIP, OSPF, (E)IGRP, IS-IS)
  - *exterior routing*
    - routing among ASs
    - the primary goal is the support of defined policies and scalability
    - so-called *Exterior Gateway Protocols (EGP)* (e.g., BGP-4)
  - a cooperation of interior and exterior routing protocols is necessary



Autonomous Systems – routing

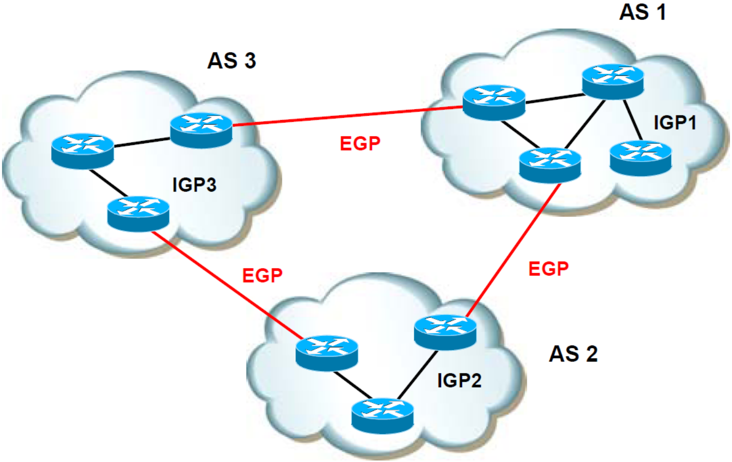


Figure: Interior (IGP) vs. Exterior (EGP) routing protocols.

Lecture Overview I

Lecture Overview II

6 Traffic Engineering in IP Networks
- Introduction
- Discovering Network Utilization
- Discovering Network Topology
- Links' Weights Computation

7 Multiprotocol Label Switching
- MPLS
- Generalized MPLS
- Grid-enabled GMPLS

8 QoS-Based Routing

9 Advanced Routing Mechanisms: Literature

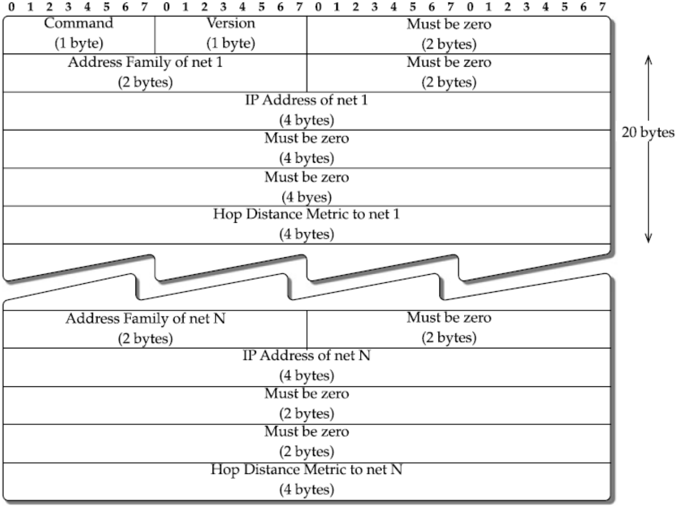
RIP protocol

Routing Information Protocol (RIP)

- the principal actor of the DV routing
    - RIPv1 (RFC 1058) – the first routing protocol used in TCP/IP-based network in an intradomain environment
    - RIPv2 (RFC 1723) – adds several features (e.g., explicit masking and an authentication of routing information)
    - RIPng (RFC 2081) – RIPv2's extension to support IPv6 addresses/networks
- the number of hops is used as a metric
    - transfer of a packet between two neighboring routers = 1 hop
- the routers send the information periodically every 30 seconds
    - messages sent over UDP protocol
    - supports triggered updates when a state of a link changes
    - timeout 180s (detection of connection errors)
- usage:
    - suitable for small networks and stable links
    - not advisable for redundant networks

RIP protocol – version 1

Message Format I.



Command (1 byte) | Version (1 byte) | Must be zero (2 bytes)

Address Family of net 1 (2 bytes) | Must be zero (2 bytes)

IP Address of net 1 (4 bytes)

Must be zero (4 bytes)

Must be zero (4 byes)

Hop Distance Metric to net 1 (4 bytes)

Address Family of net N (2 bytes) | Must be zero (2 bytes)

IP Address of net N (4 bytes)

Must be zero (2 bytes)

Must be zero (2 bytes)

Hop Distance Metric to net N (4 bytes)

RIP protocol – version 1

Message Format II.

- **Command** – indicates, whether the message is a request (a router is asking its neighbor for DV information) or a response
- **Version** – RIP version
- **Address family identifier** – identifies the address family (set to 2 for the IP address family)
- **IP address** – the destination network (identified by a subnet or a host)
- **Metric** – hop count to the destination (a number in the range $(1..16)$, $16 =$ infinity)

RIPv1 messages are *broadcast*.

RIP protocol – version 1

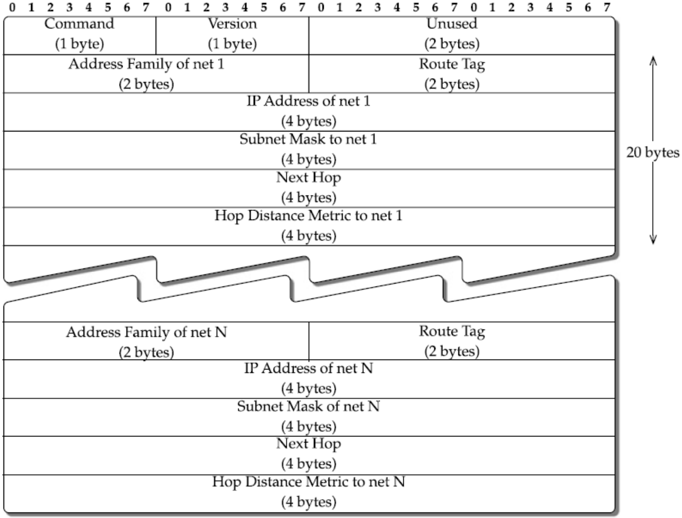
Problems Analysis

RIPv1 suffers from several problems:

- slow convergence and problems with routing loops/cycles – imposed by DV approach
- infinity $= 16 \Rightarrow$ the RIPv1 cannot be used for networks with minimal amount of hops between any two routers > 15
- has no way (no field in the messages) to indicate anything specific about the network being addressed
    - RIPv1 assumes that an address included follows a Class A, Class B, or Class C boundary implicitly
    - $\Rightarrow$ it *does NOT support variable length subnet masking*

RIP protocol – version 2

Message Format I.



Command (1 byte) | Version (1 byte) | Unused (2 bytes)

Address Family of net 1 (2 bytes) | Route Tag (2 bytes)

IP Address of net 1 (4 bytes)

Subnet Mask to net 1 (4 bytes)

Next Hop (4 bytes)

Hop Distance Metric to net 1 (4 bytes)

Address Family of net N (2 bytes) | Route Tag (2 bytes)

IP Address of net N (4 bytes)

Subnet Mask of net N (4 bytes)

Next Hop (4 bytes)

Hop Distance Metric to net N (4 bytes)

RIP protocol – version 2

Message Format II.

New fields introduced by RIPv2:

- **Route tag** – used to differentiate internal routes within a RIP routing domain from external routes (the ones obtained from an external routing protocol)
- **Subnet mask** – allows routing based on subnet instead of doing classful routing (eliminates a major limitation of RIPv1)
- **Next hop** – an advertising router might want to indicate a next hop that is different from itself

RIPv2 messages are *multicast* on `224.0.0.9`.

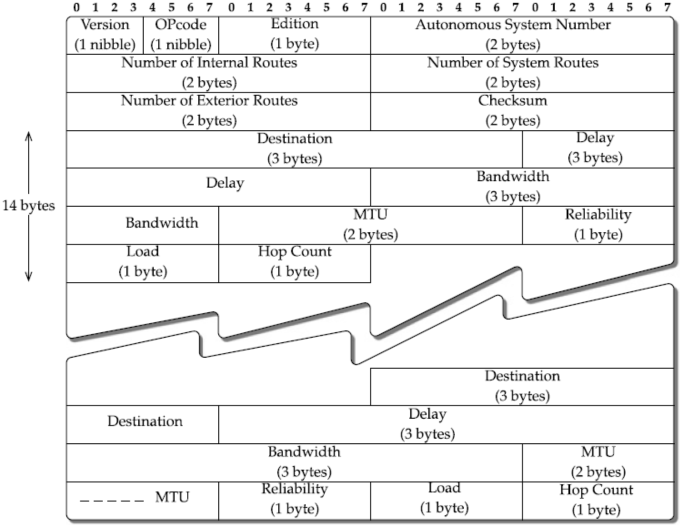
Interior Gateway Routing Protocol (IGRP)

Interior Gateway Routing Protocol (IGRP):

- developed by Cisco primarily to overcome the hop count limit and hop count metric of RIPv1
- differs from the RIPv1 in the following ways:
  - *DV updates include five different metrics for each route*
  - runs directly over IP with protocol (type field set to 9)
  - allows multiple paths for a route for the purpose of load balancing
  - external routes can be advertised
- *does NOT support variable length subnet masking*

Interior Gateway Routing Protocol (IGRP)

Message Format I.



0 1 2 3	4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Version (1 nibble)	OPcode (1 nibble)	Edition (1 byte)	Autonomous System Number (2 bytes)	
Number of Internal Routes (2 bytes)			Number of System Routes (2 bytes)	
Number of Exterior Routes (2 bytes)			Checksum (2 bytes)	
Destination (3 bytes)				Delay (3 bytes)
Delay			Bandwidth (3 bytes)	
Bandwidth		MTU (2 bytes)		Reliability (1 byte)
Load (1 byte)		Hop Count (1 byte)		
			Destination (3 bytes)	
Destination		Delay (3 bytes)		
Bandwidth (3 bytes)				MTU (2 bytes)
_ _ _ _ _ MTU		Reliability (1 byte)	Load (1 byte)	Hop Count (1 byte)

Interior Gateway Routing Protocol (IGRP)

Message Format II.

- **Version** – set to 1
- **Opcode** – ≈ *Command* field in RIPv1
- **Edition** – counter incremented by the sender (prevents from receiving an old update)
- **Autonomous system number** – ID number of an IGRP process
- **Number of interior routes** – a field to indicate the number of routing entries in an update message that are subnets of a directly connected network
- **Number of system routes** – a counterpart of the number of interior routes
- **Number of exterior routes** – the number of route entries that are default networks
- **Checksum** – value calculated on the entire IGRP packet (header + entries)

- **Destination** – the destination network for which the distance vector is generated (just *3 B are used!*)
- **Delay**, **Bandwidth**, **Reliability**, **Load** – fields for *composite metric computation*
- **Hop count** – a number between 0 and 255 used to indicate the number of hops to the destination
- **MTU** – the smallest MTU of any link along the route to the destination

IGRP messages are *multicast* on `224.0.0.10`.

Interior Gateway Routing Protocol (IGRP)

Composite Metric Computation I.

The IGRP uses a composite metric to compute a link cost:

- included to provide flexibility to compute better or more accurate routes from a link cost rather than just using a hop count
- based on four factors: *bandwidth (B)*, *delay (D)*, *reliability (R)*, and *load (L)*
  - along with five nonnegative real-number coefficients (K1, K2, K3, K4, K5) for **weighting these factors**
    - set on the routers
- The composite metric, C ("cost of a link"), is given as follows:

$$C = \begin{cases} (K_1 \times B + K_2 \times \dfrac{B}{256 - L} + K_3 \times D) \times (\dfrac{K_5}{R + K_4}), & \text{if } K_5 \neq 0 \quad (1) \\ K_1 \times B + K_2 \times \dfrac{B}{256 - L} + K_3 \times D, & \text{if } K_5 = 0 \quad (2) \end{cases}$$

Interior Gateway Routing Protocol (IGRP)

Composite Metric Computation II.

- example: $\frac{K_5}{R+K_4}$ considers the reliability of a link
  - i.e., if $K_5 = 0$ (the above part is not included), all the links have the same level of reliability
- the default, often used case: $K_1 = K_3 = 1$ and $K_2 = K_4 = K_5 = 0$
  - the composite metric reduces: $C_{default} = B + D$
  - How can we compare bandwidth (kbps, Mbps) with delay (sec, milisec)?
    - a transformation process is necessary to map the raw parameters to a comparable level
    - see the literature
- further details:
  Medhi, D. and Ramasamy, K.: Network Routing: Algorithms, Protocols, and Architectures.

Interior Gateway Routing Protocol (IGRP)

Analysis

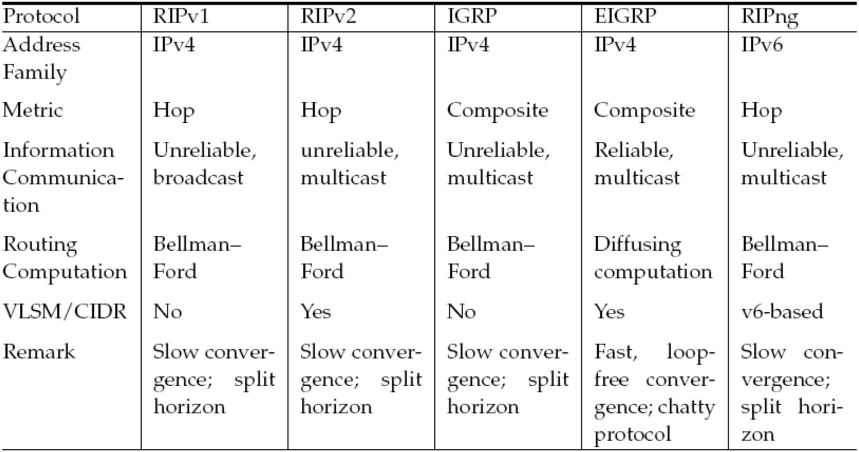
- the protocol message includes all the different metric components rather than the composite metric
    - ⇒ the composite metric is left to a router to be computed
- it is extremely important to ensure that each router is configured with the same value of the coefficients K_1, K_2, K_3, K_4, K_5
    - if NOT set equally, the routers' view of the shortest paths would be different
        - may cause routing problems

Enhanced Interior Gateway Routing Protocol (EIGRP)

Enhanced Interior Gateway Routing Protocol (EIGRP):

- another routing protocol developed by Cisco
- it enhances IGRP in many ways (e.g., it provides loop-free routing, provides reliable delivery, allows variable length subnet masking, etc.)
- the composite metric remains the same as in IGRP
- originally designed for IPv4 only, IPv6 version proposed afterwards

DV Protocols Comparison



Protocol	RIPv1	RIPv2	IGRP	EIGRP	RIPng
Address Family	IPv4	IPv4	IPv4	IPv4	IPv6
Metric	Hop	Hop	Composite	Composite	Hop
Information Communication	Unreliable, broadcast	unreliable, multicast	Unreliable, multicast	Reliable, multicast	Unreliable, multicast
Routing Computation	Bellman–Ford	Bellman–Ford	Bellman–Ford	Diffusing computation	Bellman–Ford
VLSM/CIDR	No	Yes	No	Yes	v6-based
Remark	Slow convergence; split horizon	Slow convergence; split horizon	Slow convergence; split horizon	Fast, loop-free convergence; chatty protocol	Slow convergence; split horizon

Figure: Comparison of protocols in the distance vector protocol family.

Lecture Overview I

1. Routing: Recapitulation
   - Distributed Routing
   - Autonomous Systems
2. Distance Vector Routing Protocols
   - RIP protocol
   - IGRP protocol
   - EIGRP protocol
   - Comparison
3. Link State Routing Protocols
   - OSPF Protocol
   - IS-IS Protocol
4. Path Vector Routing Protocols
   - BGP Protocol
5. Router Architectures
   - Router Introduction
   - IP Address Lookup Algorithms
   - IP Packet Filtering and Classification

Lecture Overview II

6. Traffic Engineering in IP Networks
   - Introduction
   - Discovering Network Utilization
   - Discovering Network Topology
   - Links' Weights Computation

7. Multiprotocol Label Switching
   - MPLS
   - Generalized MPLS
   - Grid-enabled GMPLS

8. QoS-Based Routing

9. Advanced Routing Mechanisms: Literature

Open Shortest Path First (OSPF) I.

Open Shortest Path First (OSPF)

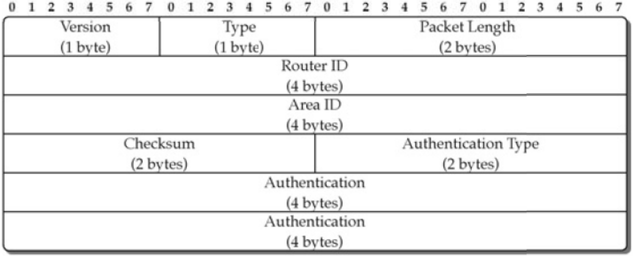
- currently the mostly used LS protocol
  - gathers link state information from available routers and constructs a topology map of the network
- metric: *cost*
  - NO hop-count
  - a number (in the range between 1 and 65535) assigned to each router's network interface
  - the lower the number is, the better the link/path is (i.e., will be preferred)
  - by default, every interface is automatically assigned a cost derived from the link's throughput
    - $cost = 100000000/bandwidth$ (bw in bps)
    - might be manually edited

Open Shortest Path First (OSPF) II.

- features:
  - *message authentication*
    - up to OSPFv2
    - OSPFv3 (running on IPv6) no longer supports protocol-internal authentication (instead, it relies on IPv6 protocol security (IPsec))
  - *routing areas*
    - next layer of hierarchy – autonomous systems can be divided into subdomains (*routing areas*)
    - to simplify administration and optimize traffic and resource utilization (lower amount of messages exchanged among same-area routers)
  - *load-balancing*
    - OSPF can make use of more outgoing links with the same (lowest) cost
    - so-called *Equal-Cost MultiPath (ECMP)*
  - *CIDR/Variable Length Subnet Mask support*
- OSPF messages are encapsulated directly in IP datagrams (protocol number 89)
  - OSPF handles its own error detection and correction functions
  - multicast is used for OSPF messages delivery (`224.0.0.5` and `224.0.0.6` for IPv4, `FF02::5` and `FF02::6` for IPv6)

Open Shortest Path First (OSPF) III.

Message Format I.



0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Version (1 byte)	Type (1 byte)	Packet Length (2 bytes)	
Router ID (4 bytes)			
Area ID (4 bytes)			
Checksum (2 bytes)		Authentication Type (2 bytes)	
Authentication (4 bytes)			
Authentication (4 bytes)			

Figure: OSPF packet common header.

OSPF messages:

- *Hello Packet*
- *Database Description Packet*
- *Link State Request Packet*
- *Link State Update Packet*
- *Link State Acknowledgement Packet*

Intermediate System To Intermediate System (IS-IS) I.

- **Intermediate System To Intermediate System (IS-IS)**
  - standardized by the ISO as a mechanism for communication between network devices (termed *Intermediate Systems*)
    - developed at the same time as the OSPF
  - originally designed for ISO-developed OSI Network Layer service called *CLNS (Connectionless Network Service)*
  - later extended to support routing of IP datagrams – called *Integrated IS-IS* or *Dual IS-IS*
    - RFC 1195
- *key similarities with the OSPF:*
  - both protocols provide network hierarchy through two-level areas
  - both protocols use *Hello packets* to initially form adjacencies and then continue to maintain them
  - both protocols support variable length subnet masks
  - both protocols maintain a link state database and perform shortest path computation using the Dijkstra's algorithm

Intermediate System To Intermediate System (IS-IS) II.

- *key differences with the OSPF:*
  - while OSPF packets are encapsulated in IP datagrams, IS-IS packets are encapsulated directly in link layer frames
  - IS-IS's run on top of layer 2 makes it relatively safer from spoofs or attacks
  - IS-IS is neutral regarding the type of network addresses for which it can route
    - easily adapted to support IPv6
    - OSPF needed a major overhaul (OSPFv3) in order to support IPv6
  - IS-IS allows overload declaration – an overloaded router may not be considered in path computation
  - OSPF's link metric value is in the range 1 to $65,535$, while IS-IS's metric value is in the range 0 to 63 (narrow metric)
    - further extended to the range 0 to $16,777,215$ (wide metric)
  - OSPF provides a richer set of extensions and added features
  - IS-IS is less "chatty" and can scale to support larger networks

Lecture Overview I

1. Routing: Recapitulation
   - Distributed Routing
   - Autonomous Systems
2. Distance Vector Routing Protocols
   - RIP protocol
   - IGRP protocol
   - EIGRP protocol
   - Comparison
3. Link State Routing Protocols
   - OSPF Protocol
   - IS-IS Protocol
4. Path Vector Routing Protocols
   - BGP Protocol
5. Router Architectures
   - Router Introduction
   - IP Address Lookup Algorithms
   - IP Packet Filtering and Classification

Lecture Overview II

6 Traffic Engineering in IP Networks
- Introduction
- Discovering Network Utilization
- Discovering Network Topology
- Links' Weights Computation

7 Multiprotocol Label Switching
- MPLS
- Generalized MPLS
- Grid-enabled GMPLS

8 QoS-Based Routing

9 Advanced Routing Mechanisms: Literature

Border Gateway Protocol (BGP) I.

Border Gateway Protocol (BGP)

- currently version 4 (*BGP-4*)
  - RFC 1771
- proposed due to Internet's grow and demands on complex topologies support
  - supports redundant topologies, deals with loops/cycles, etc.
- used to communicate information about networks currently residing in an autonomous system to other autonomous systems
  - the exchange is done by setting up a communication session between bordering autonomous systems
  - the communication channel is set on top of the TCP protocol
    - the BGP relies on a fully reliable transport protocol
- allows a definition of routing rules (policies)
- uses a hop count metric
- uses CIDR for paths' aggregation

Border Gateway Protocol (BGP) II.

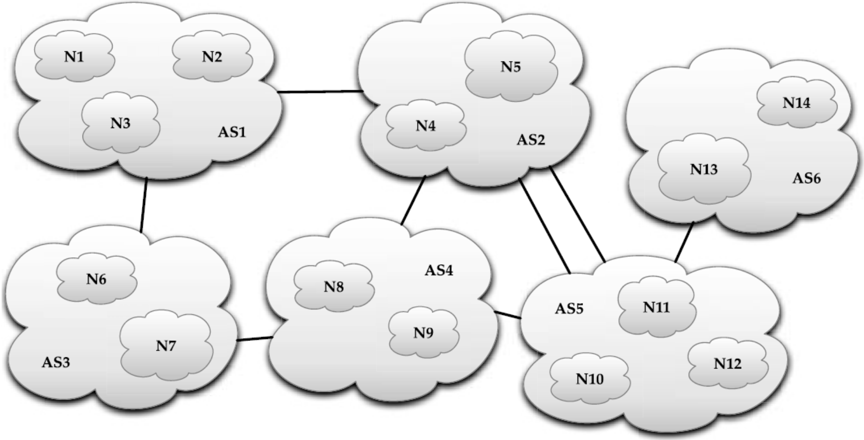


Figure: The BGP's view of the Internet architecture.

Border Gateway Protocol (BGP) III.

Advertisements

- the BGP basis upon *advertisements* sent among BGP peers:
    - sent through reliable point-to-point communication channels
        - TCP, port 179
    - an advertisement consists of:
        - a destination network address (using CIDR notation)
        - path attributes (e.g., the ASs on the path, next-hop router, etc.)
- once paths are advertised to an AS, a *routing policy* takes place
    - a routing policy defines, which ASs are allowed to transit data through the particular AS, to which ASs the data are allowed to be forwarded, etc.
    - peering contracts are big bussiness (no standards exist)
    - if a routing policy is not defined, the shortest path is chosen

Border Gateway Protocol (BGP) III.

Message Types

- *OPEN* – initiates a BGP session between a pair of BGP routers
    - allows routers to introduce themselves and to announce their capabilities
    - includes router's authentication information
- *UPDATE*
    - used to advertise routing information from one BGP router to another ("push model")
    - used to withdraw a previously announced advertisment
        - the advertised information is valid *until being explicitly withdrawn!*
- *KEEPALIVE*
    - exchanged when there is no other traffic
    - allows the BGP routers to distinguish between a failed connection and a BGP peer that has nothing to say
- *NOTIFICATION* – used to close a session or to report an error
    - e.g., rejecting an OPEN message or reporting a problem with UPDATE message
- *ROUTE-REFRESH* – a specific request to re-advertise all of the routes in router's routing table using UPDATE messages
    - not defined in the original BGP-4 (RFC 1771), but added by RFC 2918

Border Gateway Protocol (BGP) IV.

Routing table size

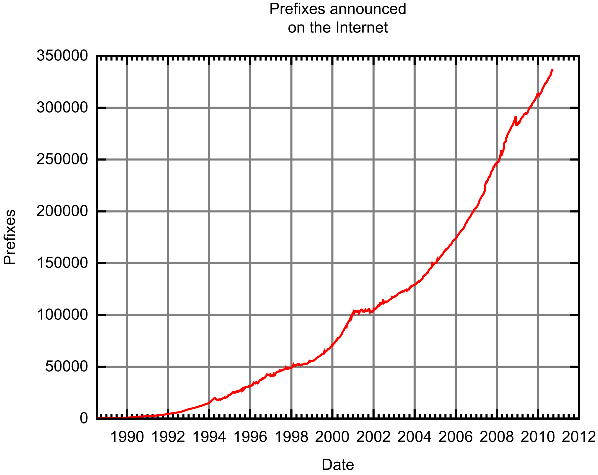


Figure: The growth of the BGP Table.

Border Gateway Protocol (BGP) IV.

Number of ASs on the Internet

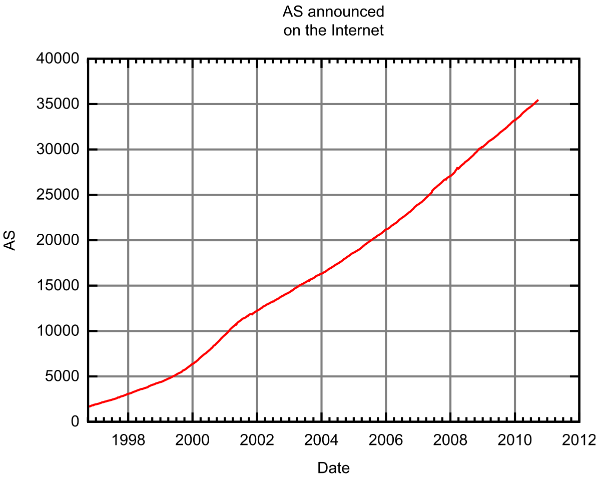


Figure: The number of autonomous systems on the Internet.

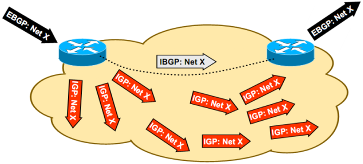
Border Gateway Protocol (BGP) V.

Internal BGP (IBGP)

The basic problem: *How to make external destinations (ASs) reachable from all the routers within an AS?*

⇒ **Internal BGP (IBGP)**

- a mechanism to provide information about adjacent ASs to internal routers of a particular AS
    - all IBGP peers within a same AS are fully meshed
    - peer announces routes received via eBGP (external BGP) to IBGP peers
    - **but:** IBGP peers do not announce routes received via IBGP to other IBGP peers
    - the learned routes are further distributed via interior routing protocol (IGP)



Lecture Overview I

1. Routing: Recapitulation
    - Distributed Routing
    - Autonomous Systems
2. Distance Vector Routing Protocols
    - RIP protocol
    - IGRP protocol
    - EIGRP protocol
    - Comparison
3. Link State Routing Protocols
    - OSPF Protocol
    - IS-IS Protocol
4. Path Vector Routing Protocols
    - BGP Protocol
5. Router Architectures
    - Router Introduction
    - IP Address Lookup Algorithms
    - IP Packet Filtering and Classification

Lecture Overview II

6 Traffic Engineering in IP Networks
- Introduction
- Discovering Network Utilization
- Discovering Network Topology
- Links' Weights Computation

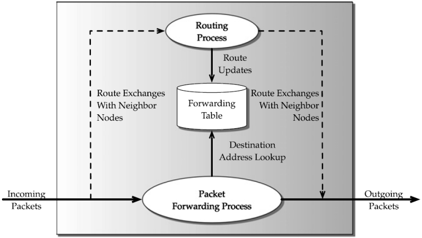
7 Multiprotocol Label Switching
- MPLS
- Generalized MPLS
- Grid-enabled GMPLS

8 QoS-Based Routing

9 Advanced Routing Mechanisms: Literature

Router Functions

- a router must perform two fundamental tasks: *routing* and *packet forwarding*
    - the **routing process** constructs a view of the network topology and computes the best paths
        - based on the information exchanged between neighboring routers using routing protocols
        - the best paths are stored in a data structure called a *forwarding table*
    - the **packet forwarding process** moves a packet from an input interface ("ingress") to the appropriate output interface ("egress")
        - based on the information contained in the forwarding table
        - the performance of the forwarding process determines the overall performance of the router



Router Functions

Basic forwarding functions I.

IP Header Validation

- every IP packet arriving at a router needs to be validated
    - e.g., the version number of the protocol is correct, the header length is valid, checksum is correct, etc.

Packet Lifetime Control

- decrementing the TTL field to prevent packets from getting caught in the routing loops forever
- it the TTL is zero or negative, the packet is discarded
    - and an ICMP message is generated and sent to the original sender

Checksum Recalculation

- since the value of the TTL has been modified, the header checksum needs to be updated

Router Functions

Basic forwarding functions II.

Route Lookup

- packet destination address is used to search the forwarding table for determining the output port

Fragmentation

- the router needs to split the packet into multiple fragments when the MTU of the outgoing link is smaller than the size of the packet that needs to be transmitted

Handling IP Options

- a packet may indicate that it requires special processing needs at the router

Router Functions

Complex forwarding functions

Packet Classification

- for distinguishing packets, a router might need to examine not only the destination IP address but also other fields
    - such as source address, destination port, and source port, etc.

Packet Translation

- a router that acts as a gateway to a NAT network needs to support network address translation

Traffic Prioritization

- a router might need to guarantee a certain quality of service to meet service level agreements

Router Functions

Routing process functions

Routing Protocols

- routers need to implement different routing protocols (e.g., OSPF, BGP, and RIP) for maintaining peer relationships by sending and receiving route updates from adjacent routers

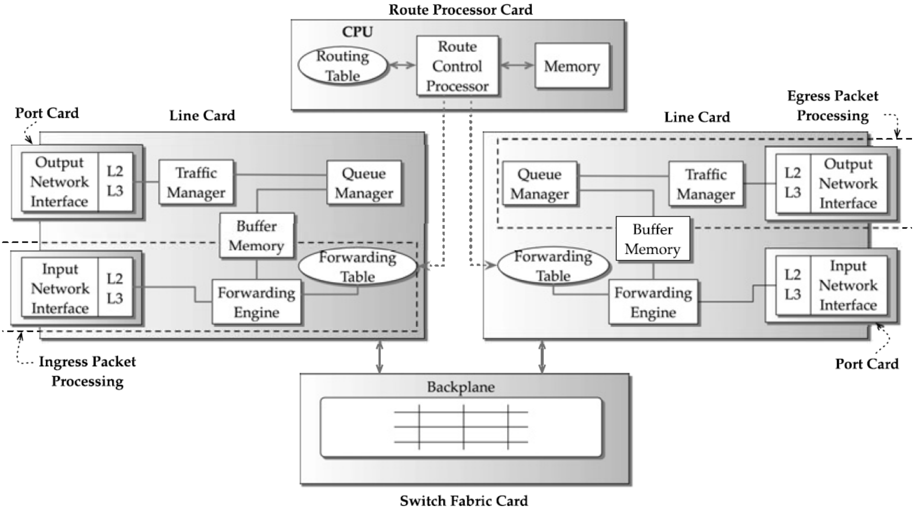
System Configuration

- a router needs to implement various functions enabling the operators to configure various administrative tasks
  - configuring the interfaces, routing protocol keep alives, rules for classifying packets, etc.

Router Management

- in addition to the configuration tasks, the router needs to be monitored for continuous operation
  - e.g., SNMP support

Router Elements



Router Elements II.

Network Interfaces

- a network interface contains many ports that provide the connectivity to physical network links
    - a port is specific to a particular type of network physical medium (Ethernet, Sonet, etc.)

Forwarding Engines

- responsible for deciding to which network interface the incoming packet should be forwarded
    - by consulting a *forwarding table* = **Address/Route Lookup**

Queue Manager

- provides buffers for temporary storage of packets when an outgoing link from a router is overbooked
- when these buffer queues overflow due to congestion, the queue manager selectively drops packets

Traffic Manager

- responsible for prioritizing and regulating the outgoing traffic, depending on the desired level of service

Router Elements III.

Backplane

- provides connectivity for the network interfaces
  - packets from an incoming network interface can be transferred to the outgoing network interface

Route Control Processor

- responsible for implementing and executing routing protocols
  - maintains a *routing table* that is updated whenever a route change occurs
    - based on the contents of the routing table, the *forwarding table* is computed and updated
- runs the software to configure and manage the router
- performs complex packet-by-packet operations
  - e.g., handling errors during packet processing
    - e.g., sending an ICMP message to the origin when packet's destination address cannot be found in the forwarding table

(a) Routing table

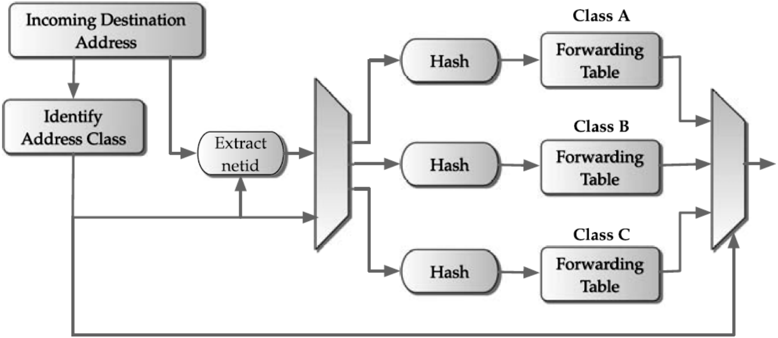
IP prefix	Next hop
10.5.0.0/16	192.168.5.254

(b) Forwarding table

IP prefix	Interface	MAC address
10.5.0.0/16	eth0	00:0F:1F:CC:F3:06

Address Lookup with Classful Addressing

- with the classful addressing scheme, the forwarding of packets is straightforward
  - routers need to examine only the network part of the destination address
  - ⇒ the forwarding table needs to store just a single entry for routing the packets destined to all the hosts attached to a given network



Address Lookup with CIDR – Longest Prefix Matching

- address lookup with CIDR is *more difficult* since:
  1. a destination IP address does not explicitly carry the netmask information
  2. the prefixes in the forwarding table against which the destination address needs to be matched can be of arbitrary lengths

Address Lookup with CIDR – Longest Prefix Matching

Requirements I.

Lookup Speed

- Internet traffic measurements show that roughly 50 % of the packets that arrive at a router are TCP-acknowledgment packets, which are typically 40-byte long
- thus, the prefix lookup has to happen in the time it takes to forward such a minimum-size packet (40 bytes)
  - known as *wire-speed forwarding*
- wire-speed forwarding for:
  - 1 Gbps link ⇒ prefix lookup should not exceed 320 nanosec
  - 10 Gbps link ⇒ prefix lookup should not exceed 32 nanosec
  - 40 Gbps link ⇒ prefix lookup should not exceed 8 nanosec

1 Gbps computed as: $\frac{40\ bytes \times 8\ bits/byte}{1 \times 10^9\ bps} = 320\ nanosec$

Address Lookup with CIDR – Longest Prefix Matching

Requirements II.

Memory Usage

- i.e., the amount of memory consumed by the data structures of the algorithm
- a memory-efficient algorithm can effectively use the fast but small cache memory

Scalability

- algorithms are expected to scale both in speed and memory as the size of the forwarding table increases

Updatability

- route changes occur fairly frequently
    - rates varying from a few prefixes per second to a few hundred prefixes per second
- ⇒ the route changes require updating the forwarding table data structure in the order of milliseconds or less

Address Lookup with CIDR – Longest Prefix Matching

Algorithms I.

Naive Algorithms

- the simplest algorithm for finding the best matching prefix is a *linear search of prefixes*
- time complexity is $O(N)$
  - N . . . number of prefixes in a forwarding table
  - useful if there are very few prefixes to search; otherwise the search time degrades as N becomes large

Trie-based Algorithms

- *note: "trie" comes from "retrieval"*, not from "tree"
- several variants proposed:
  - Binary Tries
  - Multibit Tries
  - Compressed Multibit Tries

Address Lookup with CIDR – Longest Prefix Matching

Algorithms II.

P_1	0*
P_2	00001*
P_3	001*
P_4	1*
P_5	1000*
P_6	1001*
P_7	1010*
P_8	1011*
P_9	111*

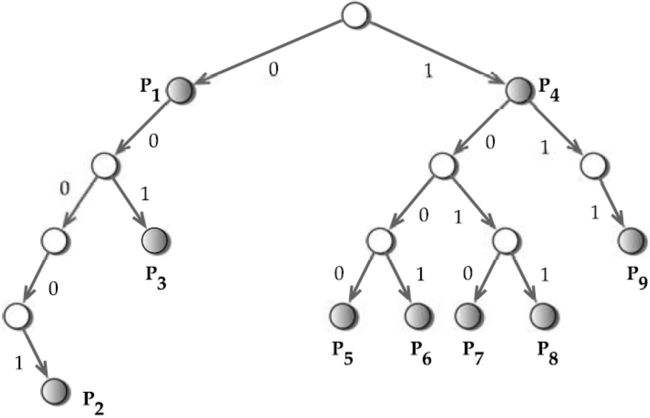


Figure: Binary trie data structure example.

Address Lookup with CIDR – Longest Prefix Matching

Algorithms II.

Other Approaches

- Search by Length Algorithms
- Search by Value Approaches
- Hardware Algorithms
  - RAM-Based Lookup, Ternary CAM-Based Lookup, Multibit Tries in Hardware, etc.

Further details:

- *Medhi, D. and Ramasamy, K.: Network Routing: Algorithms, Protocols, and Architectures.*

IP Packet Filtering and Classification I.

Importance of **Packet Classification/Filtering**:

- *Providing preferential treatment for different types of traffic*
  - to provide different service guarantees for different types of traffic, an ISP might maintain different paths for the same source and destination addresses
- *Flexibility in accounting and billing*
  - an ISP needs flexible accounting and billing based on the traffic type
    - ⇒ different traffic can be charged at different prices
- *Preventing malicious attacks*
  - the ability to identify malicious packets and drop them at the point of entry
- *etc.*

IP Packet Filtering and Classification II.

The criteria for classification are expressed in terms of *rules* or *policies*

- using the header fields of the packets
  - ⇒ the forwarding engine needs to examine packet fields other than the destination address to identify the context of the packets
  - and to perform required processing/actions in order to satisfy user requirements
- a collection of such rules/policies – *rule/policy database*, *flow classifier* or simply *classifier*
- each rule specifies:
  - a *flow* to which a packet may belong (based on expressed conditions)
    - exact match, prefix match, range match, regular expression match, etc.
  - an *action* which has to be applied to packets belonging to the flow
    - like permit, deny, encrypt, etc.
- a packet may match more than one rule in the classifier
  - a *cost* is associated with each rule to determine an unambiguous match
  - ⇒ the goal is to find the rule with the least cost that matches a packet's header
    - when the rules are placed in the order based on their cost → the goal is to find the *earliest matching rule*

IP Packet Filtering and Classification

Algorithms

- *Naive Algorithms*
  - storing the rules in a linked list in the order of increasing cost
  - storage efficient, but seach-time inefficient (does not scale)
- *Two-dimensional Solutions*
  - Hierarchical Tries, Set Pruning Tries, Grid-of-Tries
- *d-dimensional Solutions*
- *Divide and Conquer Approaches*
  - Lucent Bit Vector, Aggregated Bit Vector, Cross-Producting, Recursive Flow Classification
- *Tuple Space Approaches*
- *Decision Tree Approaches*
  - Hierarchical Intelligent Cuttings (HiCuts), HyperCuts,
- *Hardware-Based Solutions*
  - Ternary Content Addressable Memory (TCAM)

Further details:

Medhi, D. and Ramasamy, K.: Network Routing: Algorithms, Protocols, and Architectures.

Lecture Overview I

1. Routing: Recapitulation
   - Distributed Routing
   - Autonomous Systems
2. Distance Vector Routing Protocols
   - RIP protocol
   - IGRP protocol
   - EIGRP protocol
   - Comparison
3. Link State Routing Protocols
   - OSPF Protocol
   - IS-IS Protocol
4. Path Vector Routing Protocols
   - BGP Protocol
5. Router Architectures
   - Router Introduction
   - IP Address Lookup Algorithms
   - IP Packet Filtering and Classification

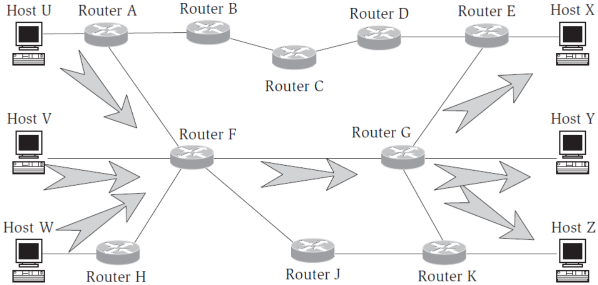
Lecture Overview II

6. Traffic Engineering in IP Networks
    - Introduction
    - Discovering Network Utilization
    - Discovering Network Topology
    - Links' Weights Computation
7. Multiprotocol Label Switching
    - MPLS
    - Generalized MPLS
    - Grid-enabled GMPLS
8. QoS-Based Routing
9. Advanced Routing Mechanisms: Literature

Traffic Engineering in IP Networks

Introduction I.

- (interior) routing protocols used in IP networks are based on *Shortest Path First (SPF)* routing
- in an unused network, the SPF is ideal:
  - datagrams are delivered expeditiously with the least use of network resources
- **Problem statement:** once traffic increases, a link/router on the shortest path may become saturated
  - while longer paths remain unused/underused
  - *Equal-Cost MultiPath (ECMP)* is usable, but NOT problem-solving solution



Traffic Engineering in IP Networks

Introduction II.

Traffic Engineering

Traffic Engineering is all about discovering what other paths and links are available in the network, what the current traffic usage is within the network, and directing traffic to routes other than the shortest so that optimal use of the resources in the network is made.

- achieved by a combination of:
    - extensions to existing IGP protocols
    - traffic monitoring tools
    - traffic routing techniques
- occurs *outside* the actual network
- does not address issues such as traffic surge lasting a few seconds/minutes

Traffic Engineering in IP Networks

Introduction III.

Performed steps:

1. traffic measurements are collected to estimate the traffic matrix
2. topology and configuration is obtained from the network
3. a link weight determination process determines link weights
    - the computed link weights for each link are injected into the network
        - i.e., each router receives a metrics for its outgoing links
    - once injected, using a normal OSPF/IS-IS flooding process the metrics are disseminated through link-state advertisements

Question: How often should the TE system update the link weights?

- up to the network provider/administrator
- usually once a day or once a week
    - to avoid short-term traffic fluctuations
    - since traffic matrix determination is a fairly complex and time-consuming process

Traffic Engineering in IP Networks

Introduction IV.

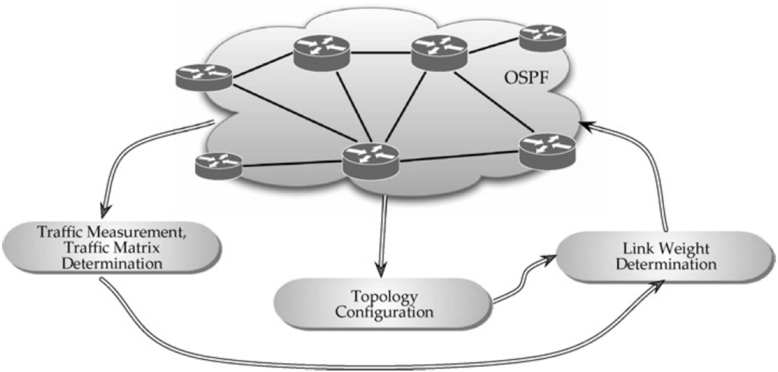
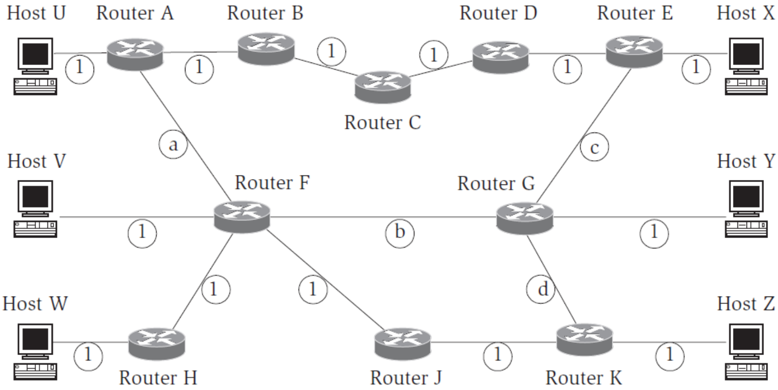


Figure: IP Traffic Engineering architectural framework.

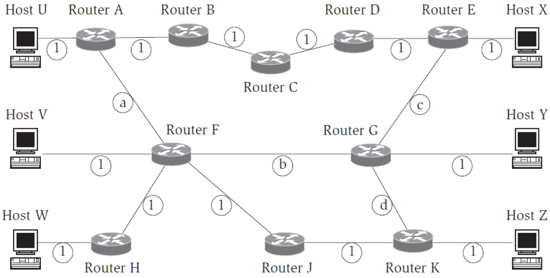
TE – Complexity illustration I.



Which costs should be assigned to a, b, c, and d?

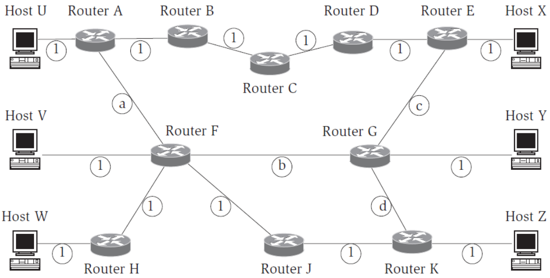
- not trivial even in such a simple network

TE – Complexity illustration II.



- $a = b = c = d = 1 \Rightarrow$ all traffic tends toward the link FG
- $a = c = d = 1,\ b = 7$
  - $U \rightarrow X$ routed through B, C, D (total cost 6)
  - $W \rightarrow Z$ routed through J (total cost 5)
  - $V \rightarrow Y$ routed through J (total cost 5)
    - not ideal – some congestion is moved to router J

TE – Complexity illustration III.



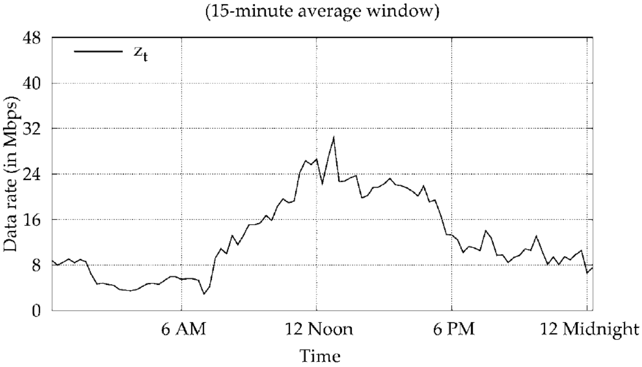
- $a = c = 2$, $b = 7$, $d = 10$ achieves the desired result
  - $U \rightarrow X$ routed through B, C, D (cost 6), $V \rightarrow Y$ routed through F, G (cost 9), and $W \rightarrow Z$ routed through J (cost 5)
  - **But:** imagine $W \rightarrow X$ traffic – takes the path WHFABCDEX (cost 9) instead of the shorter path WHFGEX (cost 12)
    - we can increase a to 6
    - but what about $U \rightarrow Y$ traffic?
      (will prefer UABCDEGY over shorter UAFGY ☹)

TE – Discovering Network Utilization I.

- a network-wide view of resource utilization is needed
  - a challenging problem
- several methods to collect and consolidate network usage information exist:
  1. *Simple Network Management Protocol (SNMP)*
     - an application polls each router and converts the returned information into a view of usage across the network
     - does not determine, which flows need to be redistributed to ease any congestion (just an absolute measure of the traffic load is obtained)
  2. *NetFlow*
     - Cisco's tool collecting the information at key points within the network
     - includes aggragation points (*NetFlow collectors*) consolidating the information from a subset of the network
  3. *sFlow*, *ntop*, etc.

TE – Discovering Network Utilization II.

- network traffic is nonstationary and (usually) time-dependent
  - data rate is different depending on the time of the day
  - ⇒ usually, a *peak* of the traffic data rate (or, say 90% of the peak) over the 24-hour window is considered as a traffic volume needed for traffic engineering considerations

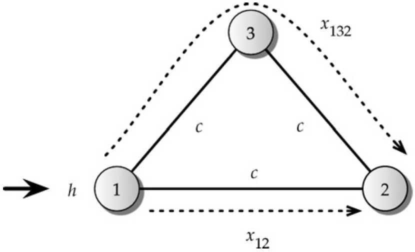


TE – Discovering Network Topology

- the application making TE decisions must have a clear view of the topology and capabilities of the links within the network
- small, static networks ⇒ manual configuration is sufficient
- large and dynamic networks ⇒ an automatic system has to be used
  - naturally, extending the IGP routing protocols to distribute additional information about the links will do the job
  - both OSPF and IS-IS have been extended to provide (for each link):
    - traffic engineering metric, maximum bandwidth, maximum reservable bandwidth, unreserved bandwidth, etc.

Network Flow Modeling – Single-Commodity Network Flow

- *single-commodity* – just a single node pair in the network has positive demand volume
  - commodity ≈ demand for a link's capacity
- let's assume the following network:

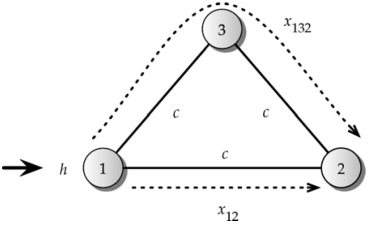


Let's denote:

- c . . . a capacity of each link (here the same for all the links)
- h . . . the demand volume for node pair $1:2$
- x_{12}, x_{132} . . . the amount of the demand volume to be routed over the path $1-2$ (resp. $1-3-2$)

Network Flow Modeling – Single-Commodity Network Flow

Problem Constraints

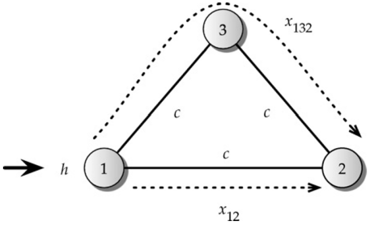


Then the following constraints have to be satisfied:

- the demand volume h has to be carried over these two paths:

Network Flow Modeling – Single-Commodity Network Flow

Problem Constraints

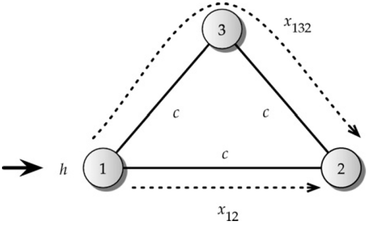


Then the following constraints have to be satisfied:

- the demand volume h has to be carried over these two paths:
  - $\Rightarrow x_{12} + x_{132} = h$

Network Flow Modeling – Single-Commodity Network Flow

Problem Constraints

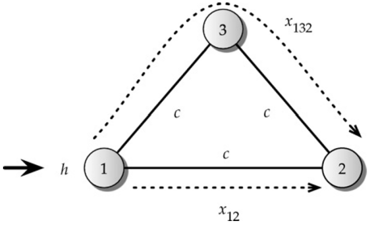


Then the following constraints have to be satisfied:

- the demand volume h has to be carried over these two paths:
  - $\Rightarrow x_{12} + x_{132} = h$
- a path may not carry any negative demand:

Network Flow Modeling – Single-Commodity Network Flow

Problem Constraints

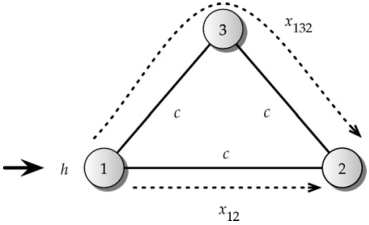


Then the following constraints have to be satisfied:

- the demand volume *h* has to be carried over these two paths:
  - $\Rightarrow x_{12} + x_{132} = h$
- a path may not carry any negative demand:
  - $\Rightarrow x_{12} \geq 0,\ x_{132} \geq 0$

Network Flow Modeling – Single-Commodity Network Flow

Problem Constraints

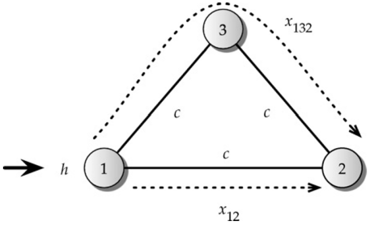


Then the following constraints have to be satisfied:

- the demand volume h has to be carried over these two paths:
    - $\Rightarrow x_{12} + x_{132} = h$
- a path may not carry any negative demand:
    - $\Rightarrow x_{12} \geq 0, \ x_{132} \geq 0$
- any flow on the path cannot exceed the capacity on any of the links the path uses:

Network Flow Modeling – Single-Commodity Network Flow

Problem Constraints

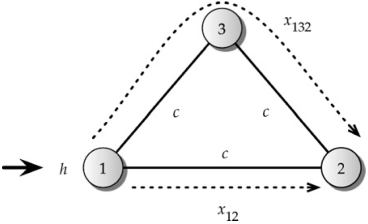


Then the following constraints have to be satisfied:

- the demand volume h has to be carried over these two paths:
  - $\Rightarrow x_{12} + x_{132} = h$
- a path may not carry any negative demand:
  - $\Rightarrow x_{12} \geq 0, \; x_{132} \geq 0$
- any flow on the path cannot exceed the capacity on any of the links the path uses:
  - $\Rightarrow x_{12} \leq c, \; x_{132} \leq c$ (same capacity on each link)

Network Flow Modeling – Single-Commodity Network Flow

The Goal – Minimize the cost of routing I.

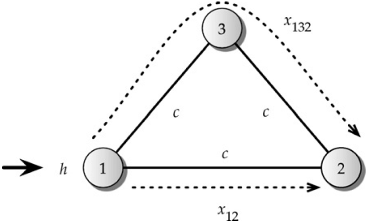


Let's assume **the goal** of minimizing the cost of routing flows:

- let's introduce a cost *per unit* of flow on each path: ξ_{12} and ξ_{132}, both ≥ 0
  - $\approx$ a price payed for data transferred over the path
- $\Rightarrow$ $Total\_cost = \xi_{12}x_{12} + \xi_{132}x_{132}$
  - $=$ the **objective function** (in general denoted by F)

Network Flow Modeling – Single-Commodity Network Flow

The Goal – Minimize the cost of routing II.



The complete problem could be written as follows:

$$\begin{array}{ll} \textbf{\textit{minimize}}_{\{x_{12},\,x_{132}\}} & F = \xi_{12}x_{12} + \xi_{132}x_{132} \\ \textbf{\textit{subject to}} & x_{12} + x_{132} = h \\ & x_{12} \le c,\ x_{132} \le c \\ & x_{12} \ge 0,\ x_{132} \ge 0. \end{array}$$

The above system solves a goal of minimizing the cost (price) of routing for the above topology when a traffic demand h is given.

- it finds proper values of x_{12} and x_{132} satisfying the given conditions

Network Flow Modeling – Single-Commodity Network Flow

The Goal – Load Balancing

Another goals could be also considered:

- load balancing – minimization of maximum link utilization
- average delay – minimization of the average packet delay

Example: *Minimization of maximum link utilization:*

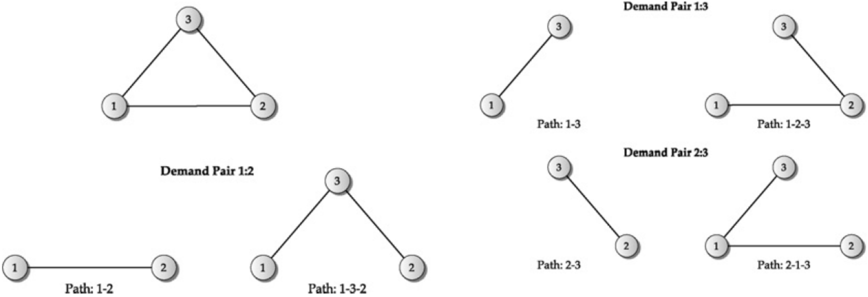
- utilization of the link $1-2$: $\frac{x_{12}}{c}$
- utilization of the links $1-3$ or $3-2$: $\frac{x_{132}}{c}$
- maximum utilization over all links: $max\{\frac{x_{12}}{c}, \frac{x_{132}}{c}\}$

$$\begin{array}{ll} \textbf{\textit{minimize}}_{\{x\}} & F = \max\{\frac{x_{12}}{c},\ \frac{x_{132}}{c}\} \\ \textbf{\textit{subject to}} & x_{12} + x_{132} = h \\ & x_{12} \le c,\ x_{132} \le c \\ & x_{12} \ge 0,\ x_{132} \ge 0. \end{array}$$

The above system solves a goal of balancing the load over paths $1-2$ *and* $1-3-2$ *when a traffic demand* h *is given.*

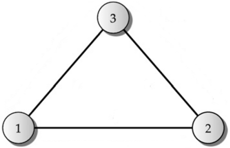
Network Flow Modeling – Multicommodity Network Flow

- *multicommodity* – all the three demand pairs can have positive demand volumes
    - h_{12}, h_{13}, h_{23}
- for each demand pair, the volume of demand can be accommodated using two paths:



Network Flow Modeling – Multicommodity Network Flow

Problem Constraints

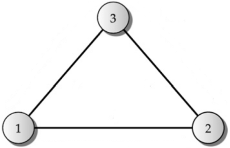


Then the following constraints have to be satisfied:

- the demand volume for each node pair may be carried over two paths:

Network Flow Modeling – Multicommodity Network Flow

Problem Constraints

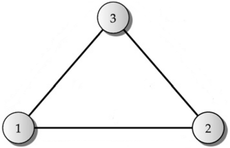


Then the following constraints have to be satisfied:

- the demand volume for each node pair may be carried over two paths:
    - $\Rightarrow x_{12} + x_{132} = h_{12}$
    - $\Rightarrow x_{13} + x_{123} = h_{13}$
    - $\Rightarrow x_{23} + x_{213} = h_{23}$

Network Flow Modeling – Multicommodity Network Flow

Problem Constraints

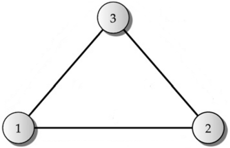


Then the following constraints have to be satisfied:

- the demand volume for each node pair may be carried over two paths:
  - $\Rightarrow x_{12} + x_{132} = h_{12}$
  - $\Rightarrow x_{13} + x_{123} = h_{13}$
  - $\Rightarrow x_{23} + x_{213} = h_{23}$
- links' capacity limits must also be satisfied:

Network Flow Modeling – Multicommodity Network Flow

Problem Constraints

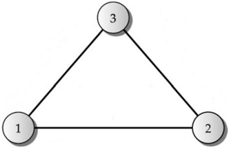


Then the following constraints have to be satisfied:

- the demand volume for each node pair may be carried over two paths:
  - $\Rightarrow x_{12} + x_{132} = h_{12}$
  - $\Rightarrow x_{13} + x_{123} = h_{13}$
  - $\Rightarrow x_{23} + x_{213} = h_{23}$
- links' capacity limits must also be satisfied:
  - $\Rightarrow x_{12} + x_{123} + x_{213} \leq c_{12}$
  - $\Rightarrow x_{13} + x_{132} + x_{213} \leq c_{13}$
  - $\Rightarrow x_{23} + x_{132} + x_{123} \leq c_{23}$

Network Flow Modeling – Multicommodity Network Flow

Problem Constraints

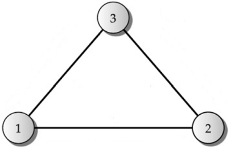


Then the following constraints have to be satisfied:

- the demand volume for each node pair may be carried over two paths:
  - $\Rightarrow x_{12} + x_{132} = h_{12}$
  - $\Rightarrow x_{13} + x_{123} = h_{13}$
  - $\Rightarrow x_{23} + x_{213} = h_{23}$
- links' capacity limits must also be satisfied:
  - $\Rightarrow x_{12} + x_{123} + x_{213} \leq c_{12}$
  - $\Rightarrow x_{13} + x_{132} + x_{213} \leq c_{13}$
  - $\Rightarrow x_{23} + x_{132} + x_{123} \leq c_{23}$
- Total cost:
  - $\Rightarrow Total\_cost = \xi_{12}x_{12} + \xi_{132}x_{132} + \xi_{13}x_{13} + \xi_{123}x_{123} + \xi_{23}x_{23} + \xi_{213}x_{213}$

Network Flow Modeling – Multicommodity Network Flow

The Goal – Minimize the cost of routing



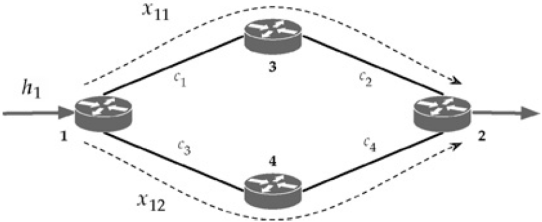
Then, **the goal** of minimizing the cost of routing can be formulated as follows:

$$\begin{array}{ll} \textbf{\textit{minimize}}_{\{x\}} & F = \xi_{12}x_{12} + \xi_{132}x_{132} + \xi_{13}x_{13} + \xi_{123}x_{123} + \xi_{23}x_{23} + \xi_{213}x_{213} \\ \textbf{\textit{subject to}} & x_{12} + x_{132} = h_{12} \\ & x_{13} + x_{123} = h_{13} \\ & x_{23} + x_{213} = h_{23} \\ & x_{12} + x_{123} + x_{213} \le c_{12} \\ & x_{13} + x_{132} + x_{213} \le c_{13} \\ & x_{23} + x_{132} + x_{123} \le c_{23} \\ & x_{12} \ge 0,\ x_{132} \ge 0,\ x_{13} \ge 0,\ x_{123} \ge 0,\ x_{23} \ge 0,\ x_{213} \ge 0. \end{array}$$

(Another goals (load balancing, average delay, etc.) can be formulated as well.)

TE – Shortest Path Routing and Network Flow

- in an IP network based on OSPF or IS-IS, the shortest paths are computed based on **links' weights**
  - this computation does NOT consider traffic volume or (usually) capacity of the network
  - the previous examples did NOT consider the links' weights
- *How is the shortest path routing related to network flow modeling?*
  - link weights drive the flows
  - let's denote w to be an array of link weights of all links in the network
    - $w = (w_1, w_2, w_3, \dots)$
    - a dependency of a flow x_{11} on the link weights will be denoted as $x_{11}(w)$



TE – Shortest Path Routing and Network Flow

MCSPRF optimization problem I.

The goal: *to determine link weights for given traffic volume demand and capacity limits where a certain objective is optimized.*

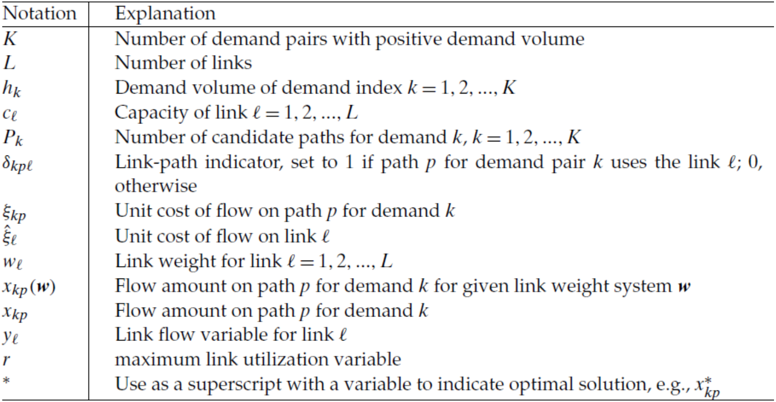
The *Multicommodity shortest path-based routing flow (MCSPRF)* optimization problem having the objective to *minimize the maximum link utilization (load balancing)* can be formulated as follows:

$$
\begin{array}{lll}
\textbf{\textit{minimize}}_{\{\boldsymbol{w},r\}} & F = r & \\
\textbf{\textit{subject to}} & \sum_{p=1}^{P_k} x_{kp}(\boldsymbol{w}) = h_k, & k = 1, 2, ..., K \\
 & \sum_{k=1}^{K} \sum_{p=1}^{P_k} \delta_{kp\ell}\, x_{kp}(\boldsymbol{w}) = y_\ell, & \ell = 1, 2, ..., L \\
 & y_\ell \le c_\ell\, r, & \ell = 1, 2, ..., L \\
 & w_1, w_2, ..., w_L \in \mathcal{W} & \\
 & x_{kp}(\boldsymbol{w}) \ge 0, & p = 1, 2, ..., P_k, \quad k = 1, 2, ..., K \\
 & y_\ell \ge 0, & \ell = 1, 2, ..., L \\
 & r \ge 0. &
\end{array}
$$

TE – Shortest Path Routing and Network Flow

MCSPRF optimization problem II.

Where



Notation	Explanation
K	Number of demand pairs with positive demand volume
L	Number of links
h_k	Demand volume of demand index $k = 1, 2, ..., K$
c_ℓ	Capacity of link $\ell = 1, 2, ..., L$
P_k	Number of candidate paths for demand k, $k = 1, 2, ..., K$
$\delta_{kp\ell}$	Link-path indicator, set to 1 if path p for demand pair k uses the link ℓ; 0, otherwise
ξ_{kp}	Unit cost of flow on path p for demand k
$\hat{\xi}_\ell$	Unit cost of flow on link ℓ
w_ℓ	Link weight for link $\ell = 1, 2, ..., L$
$x_{kp}(\boldsymbol{w})$	Flow amount on path p for demand k for given link weight system $\boldsymbol{w}$
x_{kp}	Flow amount on path p for demand k
y_ℓ	Link flow variable for link ℓ
r	maximum link utilization variable
$*$	Use as a superscript with a variable to indicate optimal solution, e.g., x^*_{kp}

The weights are determined by solving a *dual problem*.

- *details:* PA163: Constraint programming (dr. Rudová)

TE – Shortest Path Routing and Network Flow

MCSPRF – Minimum cost objective

$$
\begin{array}{lll}
\textbf{\textit{minimize}}_{\{\mathbf{w}\}} & F = \sum_{k=1}^{K} \sum_{p=1}^{P_k} \xi_{kp} x_{kp}(\mathbf{w}) & \\
\textbf{\textit{subject to}} & \sum_{p=1}^{P_k} x_{kp}(\mathbf{w}) = h_k, & k = 1, 2, ..., K \\
 & \sum_{k=1}^{K} \sum_{p=1}^{P_k} \delta_{kp\ell} x_{kp}(\mathbf{w}) \leq c_\ell, & \ell = 1, 2, ..., L \\
 & w_1, w_2, ..., w_L \in \mathcal{W} & \\
 & x_{kp}(\mathbf{w}) \geq 0, & p = 1, 2, ..., P_k, \quad k = 1, 2, ..., K.
\end{array}
$$

TE – Shortest Path Routing and Network Flow

MCSPRF – Minimum cost AND load balancing objective

$$
\begin{array}{lll}
\textbf{\textit{minimize}}_{\{x,r\}} & F = \alpha \sum_{k=1}^{K} \sum_{p=1}^{P_k} \left(\sum_{\ell=1}^{L} \hat{\xi}_\ell \delta_{kp\ell} \right) x_{kp} + \beta r & \\
\textbf{\textit{subject to}} & \sum_{p=1}^{P_k} x_{kp} = h_k, & k = 1, 2, \ldots, K \\
 & - \sum_{k=1}^{K} \sum_{p=1}^{P_k} \delta_{kp\ell} x_{kp} + c_\ell r \geq 0, & \ell = 1, 2, \ldots, L \\
 & x_{kp} \geq 0, & p = 1, 2, \ldots, P_k, \quad k = 1, 2, \ldots, K. \\
 & r \geq 0. &
\end{array}
$$

Lecture Overview I

1. Routing: Recapitulation
   - Distributed Routing
   - Autonomous Systems
2. Distance Vector Routing Protocols
   - RIP protocol
   - IGRP protocol
   - EIGRP protocol
   - Comparison
3. Link State Routing Protocols
   - OSPF Protocol
   - IS-IS Protocol
4. Path Vector Routing Protocols
   - BGP Protocol
5. Router Architectures
   - Router Introduction
   - IP Address Lookup Algorithms
   - IP Packet Filtering and Classification

Lecture Overview II

6. Traffic Engineering in IP Networks
    - Introduction
    - Discovering Network Utilization
    - Discovering Network Topology
    - Links' Weights Computation

7. Multiprotocol Label Switching
    - MPLS
    - Generalized MPLS
    - Grid-enabled GMPLS

8. QoS-Based Routing

9. Advanced Routing Mechanisms: Literature

Multiprotocol Label Switching (MPLS)

Introduction I.

Multiprotocol Label Switching (MPLS)

- a new forwarding mechanism originally presented as a way of improving the forwarding speed of core IP routers
- in MPLS network, packets are forwarded based on *labels*
  - a label is added in front of a packet (i.e., as another header so that routers know how to act based on this label)
    - assigned when packet enters the MPLS-capable network
  - internal MPLS routers don't inspect packet's IP address
    - short and fixed-length label lookup is much faster than longest-prefix match performed on every router
  - labels usually correspond to IP destination networks
    - but can also correspond to other parameters, such as QoS or source address
  - requires new protocols to distribute label information
    - or extensions to existing protocols

Multiprotocol Label Switching (MPLS)

Introduction II.

Multiprotocol Label Switching (MPLS) – cont'd.

- MPLS flows are *connection-oriented* and packets are routed along pre-configured *Label Switched Paths (LSPs)*
  - the MPLS connection (LSP) is *unidirectional*
  - ⇒ two-way communication requires a pair of LSPs to be established
    - the paths for forward and reverse directions may differ
- MPLS allows new forwarding paradigms not available with conventional IP routing
  - e.g., the ability of network operators to dictate the path that traffic takes through their network, Virtual Private Network support, etc.
    - for example, low-priority data may be sent on a longer path to keep the shortest path clear for higher-priority traffic
- MPLS has emerged into a crucial standard technology for large-scale IP networks

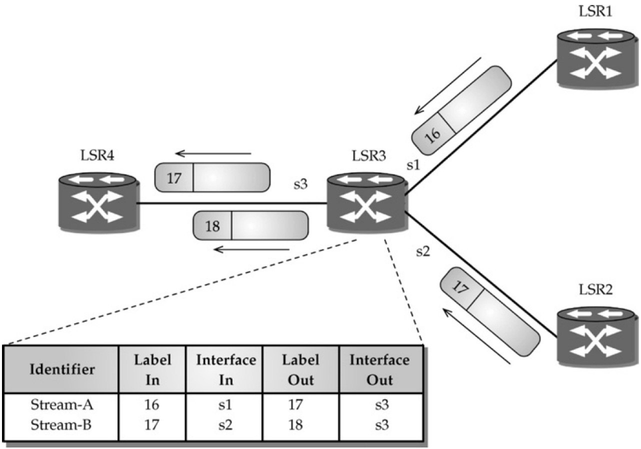
Multiprotocol Label Switching (MPLS)

Basic functionality

- an analysis of packets entering the network
    - and their classification to *FEC classes (Forward Equivalence Class)*
        - the classification may be based on more information than just on the destination address
        - for example, type of service, VPN, etc.
- labels' creation for all the FEC classes
- determination/creation of *Label Switched Paths (LSPs)*
- labels' distribution
- setting the forwarding information tables in the routers
    - the tables are known as *Label Information Base (LIB)* or *Label Forwarding Information Base (LFIB)*
    - the tables map $\{incoming\_interface, incoming\_label\}$ to $\{outgoing\_interface, outgoing\_label\}$
        - each MPLS core router maintains a valid mapping from the label of an incoming packet ("incoming label") to a label to be attached to the packet before being sent out ("output label")
- packets' forwarding (based on the label)
- MPLS header (called *shim header*) creation

Multiprotocol Label Switching (MPLS)

MPLS Example



Identifier	Label In	Interface In	Label Out	Interface Out
Stream-A	16	s1	17	s3
Stream-B	17	s2	18	s3

Figure: Label swapping and label switched paths.

Multiprotocol Label Switching (MPLS)

MPLS Network Components I.

Edge Label-Switched Routers (Edge-LSRs) = border routers

- *Ingress-LSR*
  - analyses information in IP packet header
  - based on analysed information, the packet is assigned to particular FEC
  - depending on the assigned FEC, a proper label is inserted into MPLS header
- *Egress-LSR*
  - removes MPLS header and forwards original IP packet to an egress link
  - decrements packet's TTL field

Core Label-Switched Routers (Core-LSRs)

- ensures packets' forwarding based on the assigned label
- the IP header is neither modified nor analysed by the Core-LSRs
  - just MPLS labels are analysed and modified, if necessary

Multiprotocol Label Switching (MPLS)

MPLS Network Components II.

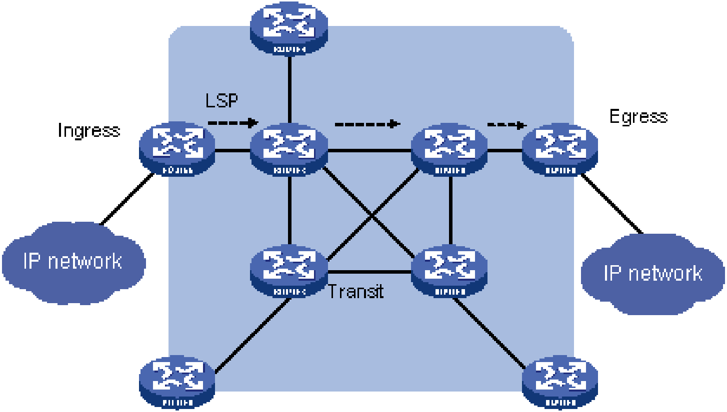
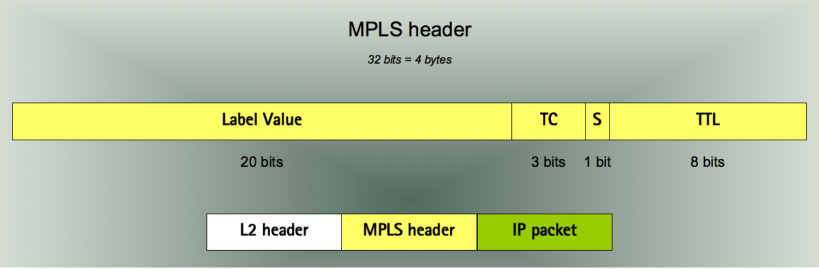


Figure: Structure of the MPLS network.

Multiprotocol Label Switching (MPLS)

MPLS Shim Header



- *Label* – carries the actual value of the Label
- *Traffic Class field* – previously named as *Experimental*
- *Stack* – set to one for the last entry in the label stack, and zero for all other label stack entries
  - receiving router examines the top label only
- *TTL* – used to encode a time to live value

Multiprotocol Label Switching (MPLS)

MPLS Labels

- usually, just a single MPLS label is assigned to a packet
- scenarios, that may produce more than one label:
  - MPLS VPNs – 2 labels
    - the top label points to the egress router and the second label identifies the VPN
  - MPLS Traffic Engineering – 2 labels
    - the top label points to the endpoint of the traffic engineering tunnel and the second label points to the destination
  - MPLS TE combined with MPLS VPNs – 3 or more labels
  - etc.

Multiprotocol Label Switching (MPLS)

MPLS Label Distribution

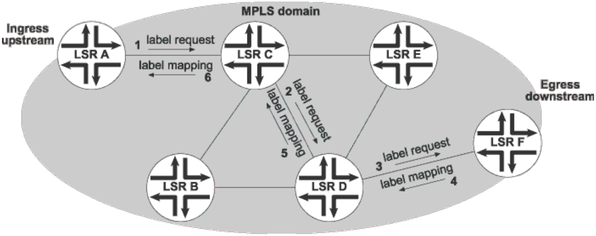
- before an LSP can be used, the LFIBs must be populated at each LSR along the path
  - ⇒ a *label distribution protocol* has to be used
- several protocols could be used:
  - *BGP (Border Gateway Protocol)* – its extension allowing labels' distribution
  - *RSVP-TE (RSVP-Traffic Engineering)* – a modified version of the RSVP protocol
  - *LDP (Label Distribution Protocol)* – a specialized protocol for MPLS networks
  - *TDP (Tag Distribution Protocol)* – Cisco's specialized protocol for MPLS networks
  - *LDP/CR (Label Distribution Protocol/Constrained Routing)* – LDP's extension for QoS support
  - etc.

Multiprotocol Label Switching (MPLS)

MPLS Label Distribution – Basic approaches I.

Downstream-on-demand, ordered control approach

- MPLS devices do not signal a FEC-to-label binding until requested to do so by an upstream device
- an LSR does not advertise a label for a FEC unless it is the egress LSR for the FEC or until it has received a label for the FEC from its downstream peer
- the same label has to be used only between adjacent LSRs!

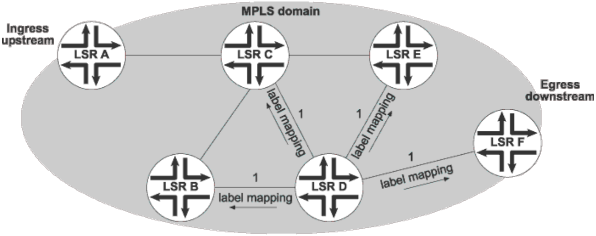


Multiprotocol Label Switching (MPLS)

MPLS Label Distribution – Basic approaches II.

Downstream-unsolicited, independent control approach

- MPLS devices do not wait for a request from an upstream device before signaling FEC-to-label bindings
  - as soon as the LSR learns a route, it sends a binding for that route to all peer LSRs, both upstream and downstream
- the LSR sending the label acts independently of its downstream peer
  - it does not wait for a label from the downstream LSR before it sends a label to its peers



Multiprotocol Label Switching (MPLS)

MPLS Label Distribution – LDP protocol

Label Distribution Protocol (LDP)

- a protocol defined by the IETF (RFC 5036) for the purpose of distributing labels in an MPLS environment
- relies on the underlying routing information provided by an IGP in order to forward label packets
- makes use of the TCP or UDP transport protocols
- can operate in both Downstream-on-demand and Downstream-unsolicited modes
- main protocol activities:
    - discovery of LDP-capable LSRs that are "adjacent"
        - LDP's *Discovery* message
    - establishment of a control conversation between adjacent LSRs, and negotiation of capabilities and options
        - LDP's *Adjacency* message
    - advertisement of labels
    - withdrawal of labels
        - both performed by LDP's *Label Advertisement* message
    - error notifications
        - LDP's *Notification* message

Multiprotocol Label Switching (MPLS)

Traffic Engineering in MPLS I.

MPLS is able to supply much of the function of the traffic engineered overlay model in an integrated manner:

- MPLS has the ability to establish an LSP that follows a path other than the one offered as “preferred” by the routing protocol and forwarding algorithm
- resources within the network can be dynamically reserved as LSPs are established and can be dynamically updated as the needs of the LSPs change
  - traffic flows can be guaranteed a level and quality of service
- traffic can be groomed onto “parallel” LSPs
  - multiple LSPs can be established between a pair of source and destination end points
  - traffic can be distributed over the LSPs by a defined algorithm
- recovery procedures can be defined describing how traffic can be transferred to alternate LSPs in the event of a failure
  - indicating how and when backup and standby LSPs should be set up and routed
- load-sharing and traffic grooming decisions need to be made just once (at the entry point into the LSP) rather than at each node within the network
- etc.

Multiprotocol Label Switching (MPLS)

Traffic Engineering in MPLS II.

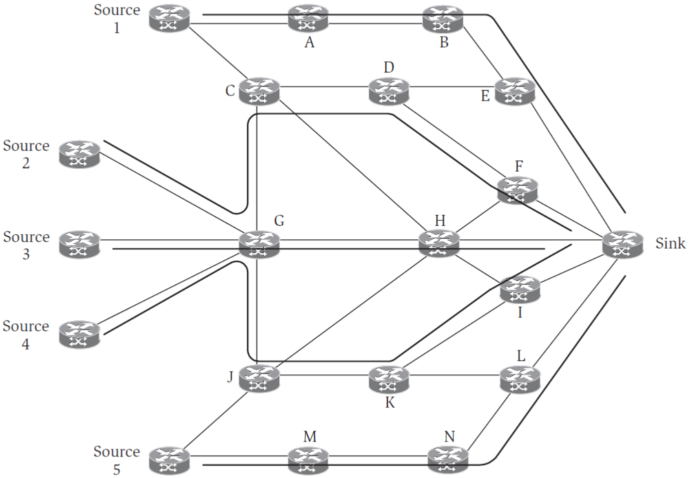


Figure: Explicit path control in an MPLS network.

Generalized MPLS (GMPLS)

Generalized MPLS (GMPLS)

- MPLS has been designed to switch packets using a labeling mechanism
- however, there is the need for an MPLS control-type functionality for controls that is beyond just switching packets
  - e.g., wavelength switching, time division multiplexing, fiber (port) switching, etc.
    - traditionally referred to as *circuit switching* or *circuit routing* (a dedicated path and physical resources must be allocated for a service from one end to another)
- GMPLS thus intended for the following switching capabilities:
  - *Packet-Switch Capable* – (i.e., GMPLS encompasses MPLS)
  - *Time-Division Multiplexing Capable* – for timeslot-based circuit switching
  - *Lambda-Switch Capable* – for wavelength switching at optical cross-connects
  - *Fiber-Switch Capable* – for fiber-level switching at optical cross-connects

Grid-enabled GMPLS (G^{2}MPLS)

Grid-enabled GMPLS (G^{2}MPLS)

- a network control plane solution that enhances the GMPLS and provides a single-step resource reservation, co-allocation, and maintenance of both network and Grid resources
  - designed by IST *Phosphorus* project
- seamlessly serves Grid jobs by co-allocating and provisioning network and Grid resources in a single-step
- not widely used (yet)

Lecture Overview I

1. Routing: Recapitulation
   - Distributed Routing
   - Autonomous Systems
2. Distance Vector Routing Protocols
   - RIP protocol
   - IGRP protocol
   - EIGRP protocol
   - Comparison
3. Link State Routing Protocols
   - OSPF Protocol
   - IS-IS Protocol
4. Path Vector Routing Protocols
   - BGP Protocol
5. Router Architectures
   - Router Introduction
   - IP Address Lookup Algorithms
   - IP Packet Filtering and Classification

Lecture Overview II

6. Traffic Engineering in IP Networks
    - Introduction
    - Discovering Network Utilization
    - Discovering Network Topology
    - Links' Weights Computation

7. Multiprotocol Label Switching
    - MPLS
    - Generalized MPLS
    - Grid-enabled GMPLS

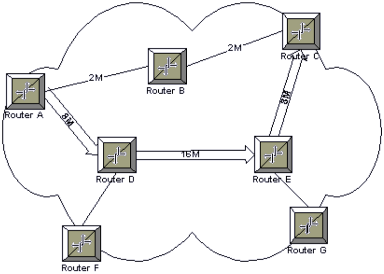
8. QoS-Based Routing

9. Advanced Routing Mechanisms: Literature

QoS-Based Routing – Introduction I.

QoS-Based Routing is defined as:

- *a routing mechanism under which paths for flows are determined based on some knowledge of resource availability in the network as well as the QoS requirement of the flows*, or
- *a dynamic routing protocol that has expanded its path-selection criteria to include QoS parameters such as available bandwidth, link and end-to-end path utilization, node resources consumption, delay and latency, and induced jitter*



QoS-Based Routing – Introduction II.

Objectives of QoS-based Routing:

- *to meet the QoS requirements of end users*
  - QoS-based routing is supposed to dynamically find a path from source to destination which can satisfy user's requirements on bandwidth, end-to-end delay, etc.
- *to optimize the network resource usage*
  - QoS-based routing is expected to direct network traffic in an efficient way that can maximize the total network throughput
- *to gracefully degrade network performance when things like congestion happen*
  - when network is in heavy load, QoS-based routing is expected to give better performance (e.g., better throughput) than the best-effort routing, which can degrade the performance dramatically

QoS-Based Routing – Issues I.

Metric and path computation

- How to measure and collect network state information?
- How to compute routes based on the information collected?
- a suitable *metric* has to be chosen (e.g., available bandwidth, delay, jitter, etc.)
- path computation is also closely related to resource reservation
  - once a feasible path is chosen, the corresponding resources (bandwidth, buffer space in routers etc.) must be reserved for the traffic flow thus are not available to other flows

Knowledge propagation and maintenance

- How often is the routing information exchanged between the routers?
  - more information has to be exchanged than in the case of best-effort routing
    - QoS information (available BW) has to be exchanged along with common routing information like connection topology changes
  - the metrics used by QoS-based routing could be changing very quickly
    - if the routing information is exchanged every time the values of metrics change, it will cause a great burden for the network links and routers $\Rightarrow$ a common way is to set a *threshold* to distinguish significant changes from minor changes (routing information accuracy becomes lower, however)

QoS-Based Routing – Issues II.

Scaling by hierarchical aggregation

- QoS-based routing is expected to be scalable
  - in order to keep the complexity of path computation and the amount of information need to be exchanged and maintained under control, a *hierarchical aggregation* is used
    - however, such aggregation brings inaccuracy in regard of routing information

Administrative Control

- different flows in the network should have different priorities
- in the framework having multiple service classes (e.g., DiffServe), the resources should be allocated fairly among all the classes
  - to avoid starvation of lower priority classes

Integration of QoS-based routing and Best-effort routing

- for compatibility, QoS-based routing must be able to support best-effort routing
  - i.e., both routing schemes must be able to coexist

QoS-Based Routing – Routing Algorithms

Basic types

- QoS-based routing algorithms classified according to the way how the state information is maintained and how the search of feasible paths is carried out
    - *source-based routing algorithms*
    - *hop-by-hop routing algorithms* (also called *distributed routing algorithms*)
    - *hierarchical routing algorithms*

QoS-Based Routing – Routing Algorithms

Basic types – Source-based routing

Source-based routing algorithms

- every router has global state information about the network, and the path is locally selected based on the state information
- once the path is determined, the source router notifies the other router along that path how to forward the traffic flow
- *features:*
  - simpler in the sense that it's decided solely by the source
- *drawbacks:*
  - requires that each router has complete state information of the network (hard to maintain)
  - the computation overhead at the source routers is very high
  - ⇒ **scalability problems** (not suitable for large networks)

QoS-Based Routing – Routing Algorithms

Basic types – Hop-by-hop routing

Hop-by-hop routing algorithms

- each router just knows the next hop towards the destination
- *features:*
  - used by most current “best-effort” routing protocols ⇒ it’s more natural to design and more compatible with existing routing protocols
  - the routing computation burden is distributed among all the routers along the path
- *drawbacks:*
  - it has the routing loop problem (when the routing state information in different routers is not consistent)
  - besides, it also has the scalability problem

QoS-Based Routing – Routing Algorithms

Basic types – Hierarchical routing I.

Hierarchical routing algorithms

- the routing structure consists of multiple levels
  - the bottom level contains the actual routers
  - these routers are organized into some logical groups, which in turn form the *next level*
  - the groups can be further organized into some higher level groups
- the routing information is integrated at the border nodes of each groups
  - every node contains the detailed information about its group and integrated information about other groups
- *features:*
  - scalability ⇒ it's suitable for large networks
- *drawbacks:*
  - aggregation decreases the accuracy of the routing state information

QoS-Based Routing – Routing Algorithms

Basic types – Hierarchical routing II.

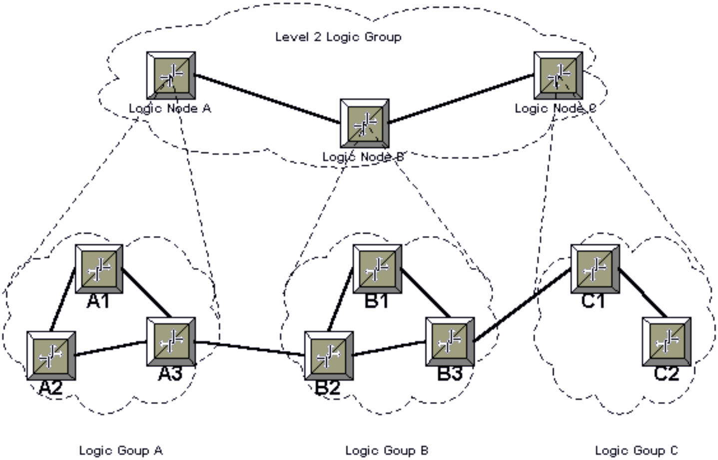


Figure: Hierarchical routing structure.

QoS-Based Routing – Routing Protocols

PNNI (Private Network-Network Interface)

Private Network-Network Interface (PNNI)

- a hierarchical, dynamic routing protocol for ATM networks
- based on link-state algorithm
  - topology information (including information about nodes, links, addresses) is flooded through the network
  - network resources are defined by metrics and attributes (delay, available bandwidth, jitter, etc.)
    - grouped by supported traffic classes
  - threshold algorithms are used to determine if the change in a metric or attribute is significant enough to require propagation of updated information
- hierarchical ⇒
  - PNNI has the concepts of levels and logical nodes
  - supports aggregation of topology and reachability information
- *drawbacks:*
  - doesn't support multicast and policy routing, and control of alternate routing
  - inherits the common problems with link state QoS-based routing
    - an issue with efficient broadcast of state information (especially for dynamic metrics)

QoS-Based Routing – Routing Protocols

QOSPF (QoS routing extensions to OSPF)

QoS routing extensions to OSPF (QOSPF)

- QoS extension to OSPF
- hierarchical protocol based on link-state algorithm
- supposed to be working in an environment in which both QoS-based routing and best-effort routing are needed
- for simplicity, **link bandwidth** and **propagation delay** are the only metrics extension added to *Link State Advertisements (LSAs)*
- in order to decrease protocol overhead, LSAs are triggered only when there is a significant change in the value of the metrics since the last advertisement
- a concept of QoS paths pre-computation is used:
  - for every possible destination, the algorithm pre-computes a "widest-shortest path" (a minimum hop count path with maximum bandwidth available)
    - a widest path version of the Bellman–Ford is used for pre-computations
    - a widest shortest path version of Dijkstra's algorithm is used for on-demand computations

Advanced Routing Mechanisms: Literature

- relevant RFCs
- Medhi, D. and Ramasamy, K.: *Network Routing: Algorithms, Protocols, and Architectures.* Morgan Kaufmann Publishers (Elsevier), 2007.
- Farrel, A.: *The Internet and Its Protocols: A Comparative Approach.* Morgan Kaufmann Publishers (Elsevier), 2004.
- Beijnum, I.: *BGP.* O'Reilly Media, Inc., 2002.
- Moy, J. T.: *OSPF: anatomy of an Internet routing protocol.* Addison-Wesley, 1998.
- Zhang, R. and Bartell, M.: *BGP design and implementation.* Cisco Press, 2004.
- Black, U.D.: *IP routing protocols: RIP, OSPF, BGP, PNNI, and Cisco routing protocols.* Prentice Hall PTR, 2000.
- `http://www.tcpipguide.com`