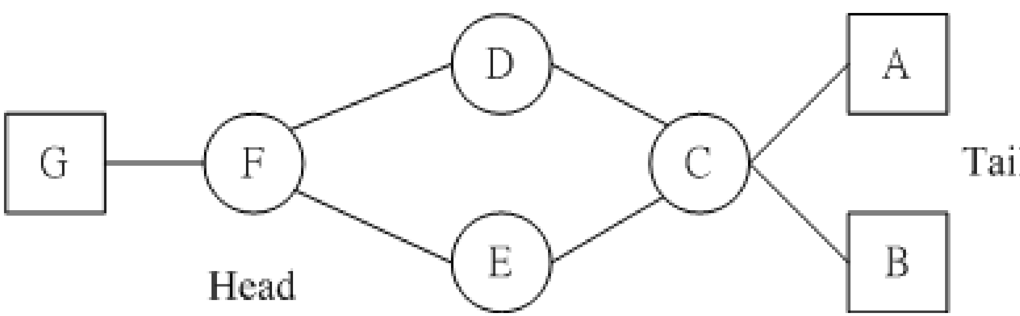
Lecture 6
Internet Traffic Engineering

Introduction of Traffic Engineering

- Traffic engineering is concerned with the performance optimization of operational networks
- The objective of traffic engineering is to reduce congestion hot spots and improve resource utilization across the network through carefully managing the traffic distribution inside a network

Fish Problem

- Extremely unbalanced traffic distribution: all traffics from A and B to G go over one of the two paths
- Reasons
  - Destination-based IP routing: all packets to the same prefix address have the same next hop
  - Local optimization-based decision making in current routing protocols
  - From the viewpoint of service providers, traffic split is preferred
    - ATM VCs with IP routing on top of VCs
    - Traffic distribution is managed by carefully mapping VCs to the physical network topology



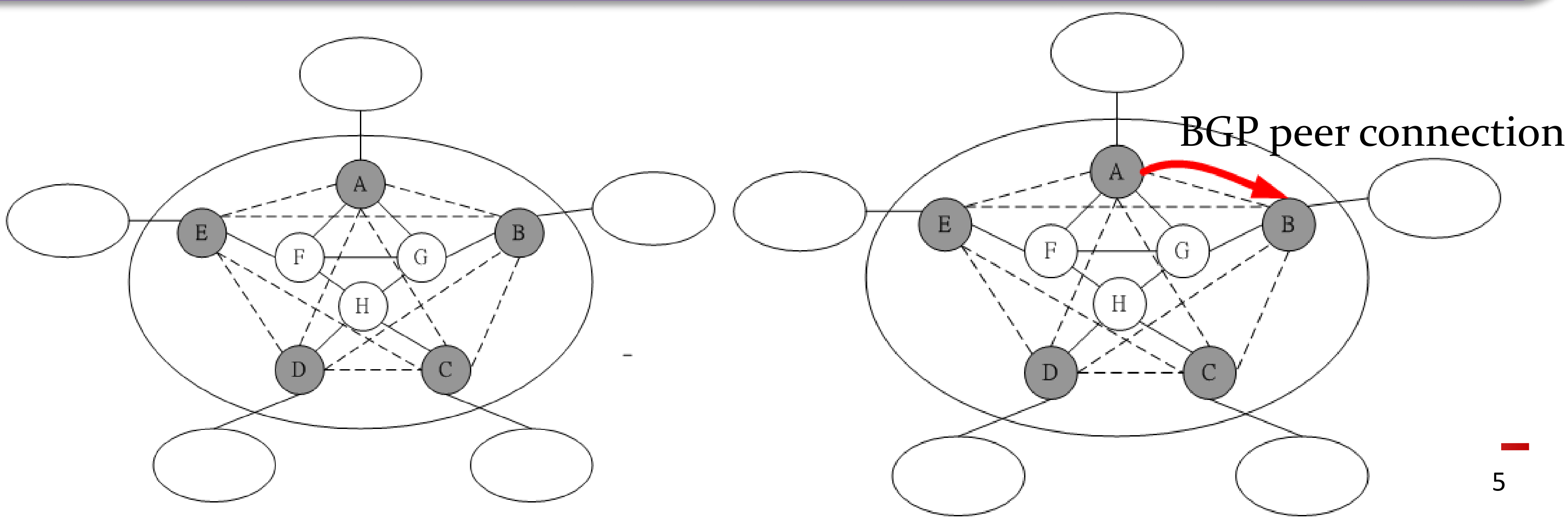
Solutions of Traffic Engineering (1)

- Overlay model
  - Allowing service providers to build arbitrary virtual topologies over the network's physical topology
  - Usually build a full mesh of logical connections between all edge nodes
    - Collecting the info of network topology and traffic demands
    - Calculating a set of optimal routes with constraint-based routing
    - These logical connections are set up as MPLS explicit routes
  - Cooperate with BGP routers
    - Dotted line: logical connection
    - Solid line: physical connection
    - A, B, C, D, E: BGP edge routers
    - The difference of with and without overlay model
      - LSPs vs. IGP-based approach

Solutions of Traffic Engineering (2)

- (+) Flexibility
- (–) Scalability (N-square problem)
- (–) Breakdown of a single physical trunk may cause multiple LSPs to fail

There are 4 paths from A to C: A→G→H→C; A→F→H→C; A→G→F→H→C; A→F→G→H→C
With overlaying, any of 4 paths can be used, and is decided based on traffic between other BGP nodes
Without overlaying, path from A to C is determined by an IGP such as OSPF

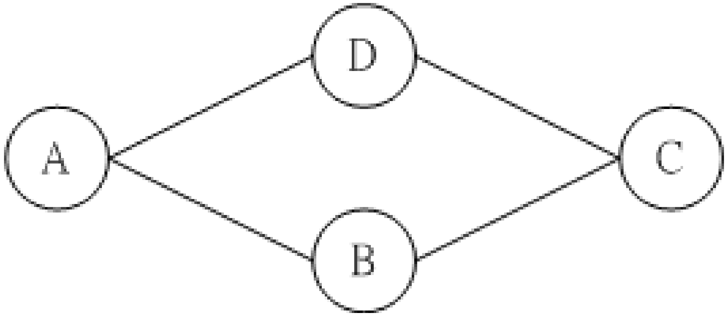


Solutions of Traffic Engineering (3)

- Peer model
  - Achieving balanced traffic distribution by manipulating link weights in OSPF routing protocol
  - Much more scalable than overlay model

Optimization Objectives (1)

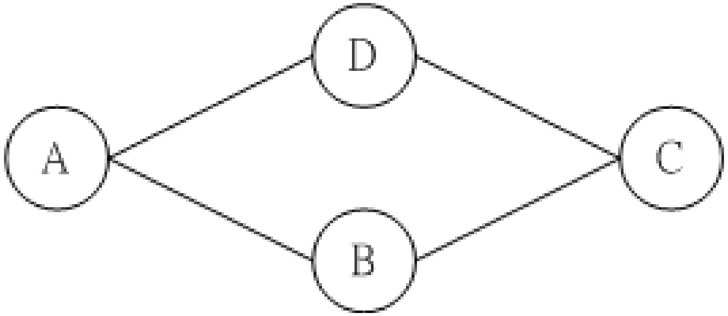
- The objectives could be:
  - Minimization of congestion and packet loss
  - Improvement of link utilization
  - Minimization of experienced delay
  - Increase of served customers



- Factors of congestion
  - Inadequate network resources
  - Unbalanced traffic distribution
    - The problem can be formulated to “minimize the maximum of link utilization in a network”
  - Interesting observation: relationship between link utilization and queueing delay
    - The maximum of link utilization is minimized, the total delay the packets experience is also minimized

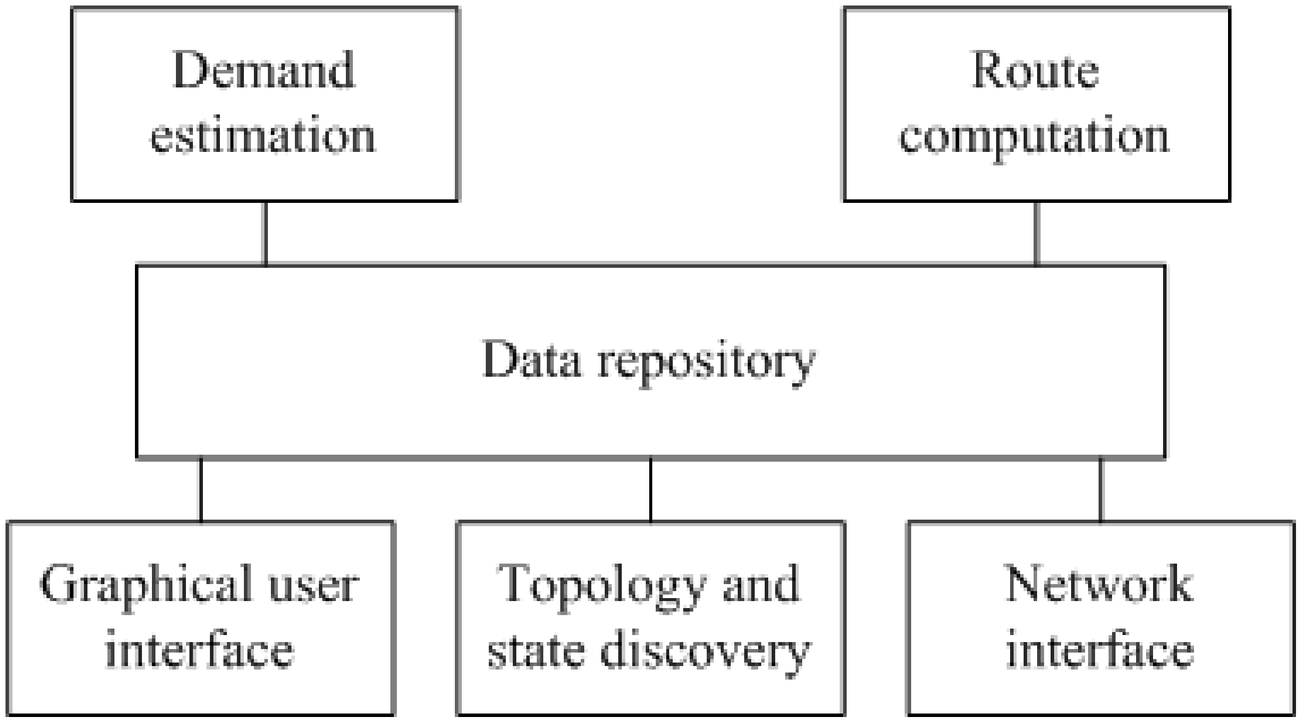
Optimization Objectives (2)

- Assume all links have the same capacity
- Node A can split traffic from itself to node C
- What's the optimal allocation among the 2 paths?
  - Queueing delay increases more rapidly as the utilization increases
  - The increase of the delay on path A→D→C is always larger than the decrease of the delay on path A→B→C



Building Blocks of Internet TE (1)

- Data repository: a central database and persistent storage for all shared data objects (e.g., network topology, link state, and traffic demand)



Building Blocks of Internet TE (2)

- Topology and state discovery: monitor topology and link state changes
  - Possible approaches
    - Static information: network configuration
    - Dynamic information: residual bandwidth, link utilization (SNMP & polling)
    - Routing protocol extension (to broadcast link state periodically in OSPF)
- Traffic demand estimation
  - Service agreement between service providers and customers
  - Traffic measurement
    - Aggregated traffic statistics (ex., traffic load between any pair of ingress and egress nodes)

Building Blocks of Internet TE (3)

- Route computation
  - Calculated based on traffic demands
  - Constraint-based routing
    - Off-line mode
      - Route computation is performed for all routes periodically with current info
      - The network cuts over to the new routes during maintenance periods
      - (+) Yielding optimal results
      - (-) Traffic flow disruption
      - (–) Extra delays for adding new traffic demands
    - On-line mode
      - Route computation is performed in an incremental fashion and only calculates for new demands
      - (+) Minimizing rerouting of existing flows
      - (-) Inefficient resource utilization
  - Utilizing both modes at different time scales alternatively

Building Blocks of Internet TE (4)

- Network interface
  - Configuring network elements in the network accordingly
  - Two approaches
    - Web-based user interface (usually vendor specific)
      - (+) Highly flexible
      - (–) Vender-specific
    - Standard-based approach: SNMP/Common open policy server protocol (COPS)
      - COPS: proposed as a standard for a traffic-engineering system to instruct an edge node to set up MPLS explicit routes

Constraint-Based Routing (1)

- Consist of two basic elements
  - Route optimization: selecting routes for traffic demands subject to a given set of constraints
  - Route placement: implement the selected routes in the network so that the traffic flows will follow them
- Mathematical formulation
  - Assumptions include
    - Network links and capacities are directional (directed graph)
    - Average traffic demand is known (either through measurement or through configuration parameters)
    - Traffic demand is directional

Constraint-Based Routing (2)

- Objectives
  - All traffic demands are fulfilled
  - Minimizing the maximum of link utilization
- Notations
  - $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$
  - $c_{i,j}$: the capacity of link $(i,j), \forall (i,j) \in \boldsymbol{E}$
  - $\boldsymbol{K}$: the set of traffic demands between a pair of edge nodes
  - (bw_k, s_k, t_k) : bandwidth demand, source node, destination node, $\forall k \in \boldsymbol{K}$
  - Note that a demand may be split over multiple paths
  - $X_{i,j}^k$: the percentage of k's bandwidth demand satisfied by link (i,j)
  - α : the maximum of link utilization among all the links

Constraint-Based Routing (3)

– Linear-programming formulation (LPF)
  - Objective is to minimize the maximum of link utilization, i.e., $\min \alpha$
  - Constraints:
    – The traffic flowing into a node must equal the traffic flowing out of the node for any node other than the source node and the destination node for each demand, i.e.,

$$\sum_{j:(i,j)\in \boldsymbol{E}} X_{i,j}^{k} - \sum_{j:(j,i')\in \boldsymbol{E}} X_{j,i'}^{k} = 0, \quad k \in \boldsymbol{K}, j \neq s_k, j \neq t_k$$

    – The net flow out of the source node is 1 (the assumed total required bandwidth), i.e.,

$$\sum_{j:(i,j)\in \boldsymbol{E}} X_{i,j}^{k} - \sum_{j:(j,i)\in \boldsymbol{E}} X_{j,i}^{k} = 1, \quad k \in \boldsymbol{K}, i = s_k, j \neq t_k$$

$$\Rightarrow \sum_{j:(i,j)\in \boldsymbol{E}} X_{i,j}^{k} = 1, \quad k \in \boldsymbol{K}, i = s_k, j \neq t_k$$

Constraint-Based Routing (4)

– The total bandwidth consumed by all logical connections on a link should not exceed the max utilization rate times the total capacity of the link, i.e.,

$$\sum_{\forall k} bw_k X_{i,j}^k \leq c_{i,j}\alpha, (i,j) \in \boldsymbol{E}$$

– All variables are nonnegative real numbers and no more than 1, i.e.,

$$0 \leq \alpha \leq 1$$

$$0 \leq X_{i,j}^k \leq 1$$

Constraint-Based Routing (5)

- Constraint-based routing in overlay model
  - Shortest path (SP)
    - When there is a demand, the shortest-path algorithm is used to set up an LSP to the destination
    - Link metric for link (i,j) is inversely proportional to the bandwidth
  - Minimum hop (MH)
    - Link metric is set to 1 uniformly for each hop
    - Still run shortest path algorithm
  - Shortest-widest path (SWP)
    - Link metric is set to as bandwidth
    - Always selecting the path with largest bottleneck bandwidth
    - The one with minimum hops or shortest distance is chosen when multiple paths are available

Constraint-Based Routing (6)

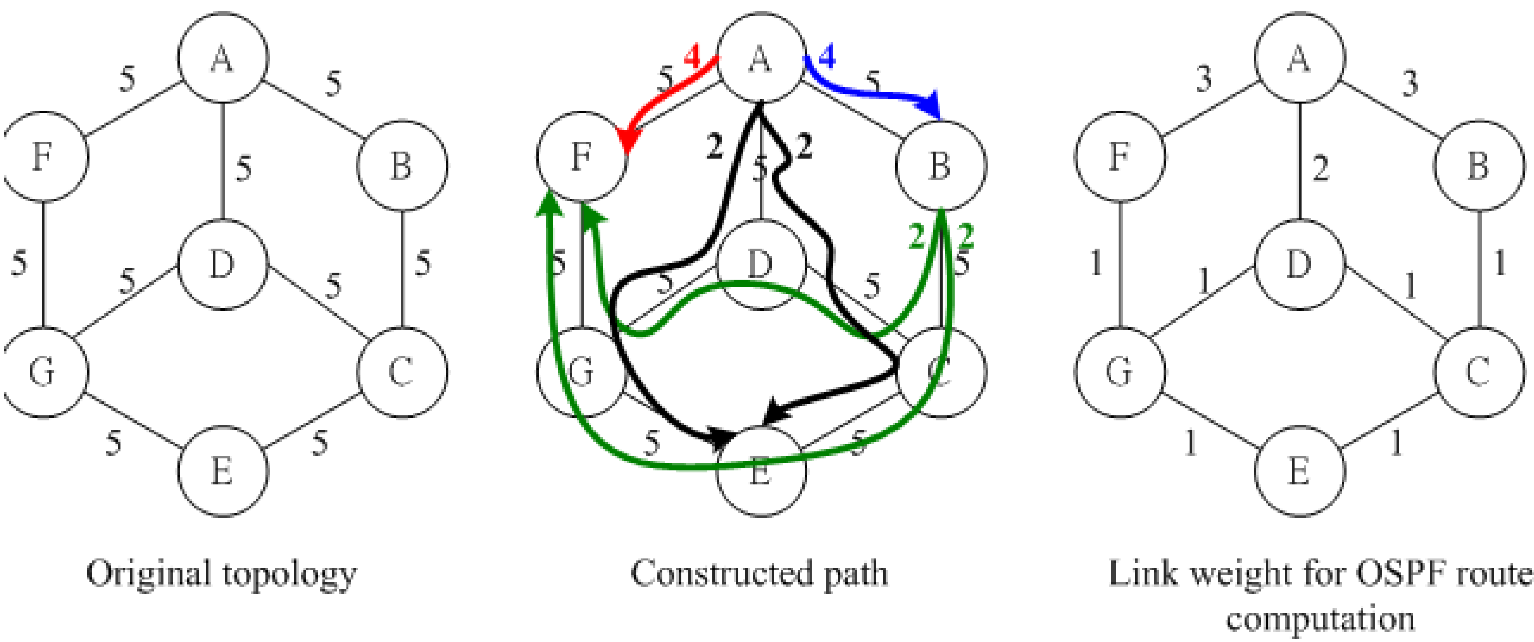
- Hybrid algorithm
  - Concerning the basic tradeoff between avoiding bottleneck links and taking a longer path, and trying to balance the two objectives
  - How? Assigning appropriate weights for them
  - Weight: link utilization, instead of residual bandwidth
  - Metric combines the cost of a path and the contribution to the maximum of link utilization
  - Notations
    - $f_{i,j}$: current load (used capacity) of link (i, j); initial value is 0
    - $c_{i,j}$: total capacity of link (i, j)
    - α : current maximum link utilization; initial value is 0
    - $\alpha_{i,j}$: link (i, j) cost metric [link (i, j) utilization]

$$\alpha_{i,j} = \frac{f_{i,j} + bw_k}{c_{i,j}} + T \times \max\left[0, \frac{f_{i,j} + bw_k}{c_{i,j}} - \alpha\right],$$

where T is a tunable parameter

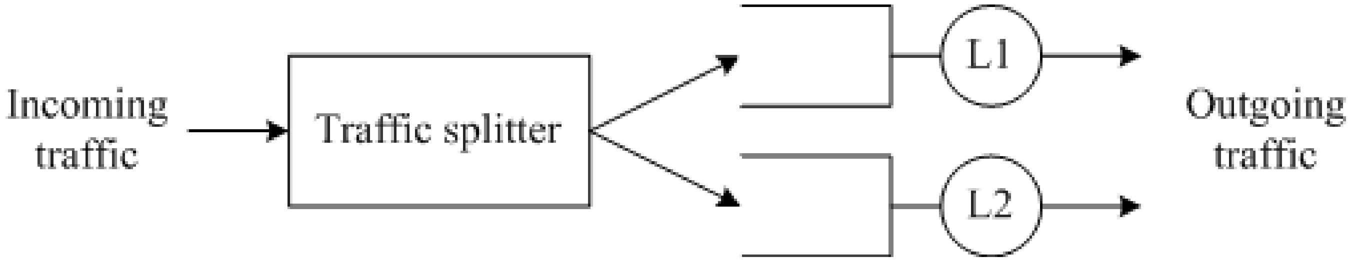
Constraint-Based Routing (7)

- Constraint-based routing in <u>peer model</u>
  - The basic idea behind peer model is link cost adjustment
  - Challenge: how to find a systematic approach of link weight assignment
  - An example
    - Traffic demands: (A→B, 4), (A→F, 4), (B→F, 4), (A→E, 4)
    - Link utilization: 36/45=80%



Multipath Load Sharing (1)

- Load-balancing system: distribute traffic equally or proportionally
- Basic requirements
  - Traffic splitting is in the packet-forwarding path, and executed for every packet
  - To reduce implementation complexity, the system should preferably keep no or little state info
  - Traffic-splitting schemes produce stable traffic distribution across multiple outgoing links with minimum fluctuation
  - Traffic-splitting algorithms must maintain per-flow packet ordering



Multipath Load Sharing (2)

- Packet-by-packet round robin
  - (+) Low overheads
  - (+) Good performance
  - (-) Per-flow ordering (sequence numbers/states)
- Hashing-based traffic-splitting schemes
  - (+) Stateless
  - (+) Easy to implement
  - (+) Even traffic load distribution

Multipath Load Sharing (3)

- Direct hashing
  - Hashing of destination address
    - $H(\bullet)$=DestIP mod N
    - N: the number of outgoing links
  - Hashing using XOR folding of source/destination addresses
    - $H(\bullet)=(S1 \otimes S2 \otimes S3 \otimes S4 \otimes D1 \otimes D2 \otimes D3 \otimes D4) \bmod N$
    - $\otimes$: XOR operation
    - S_i: the i^{th} octet of the source address
    - D_i: the i^{th} octet of the destination address

Multipath Load Sharing (4)

- Table-based hashing
  - Split a traffic stream into M bins.
  - The *M* bins are mapped to *N* outgoing links based on an allocation table, i.e., compute the corresponding hash value
  - By changing the allocation of the bins to the outgoing links, we can distribute traffic in a predefined ratio