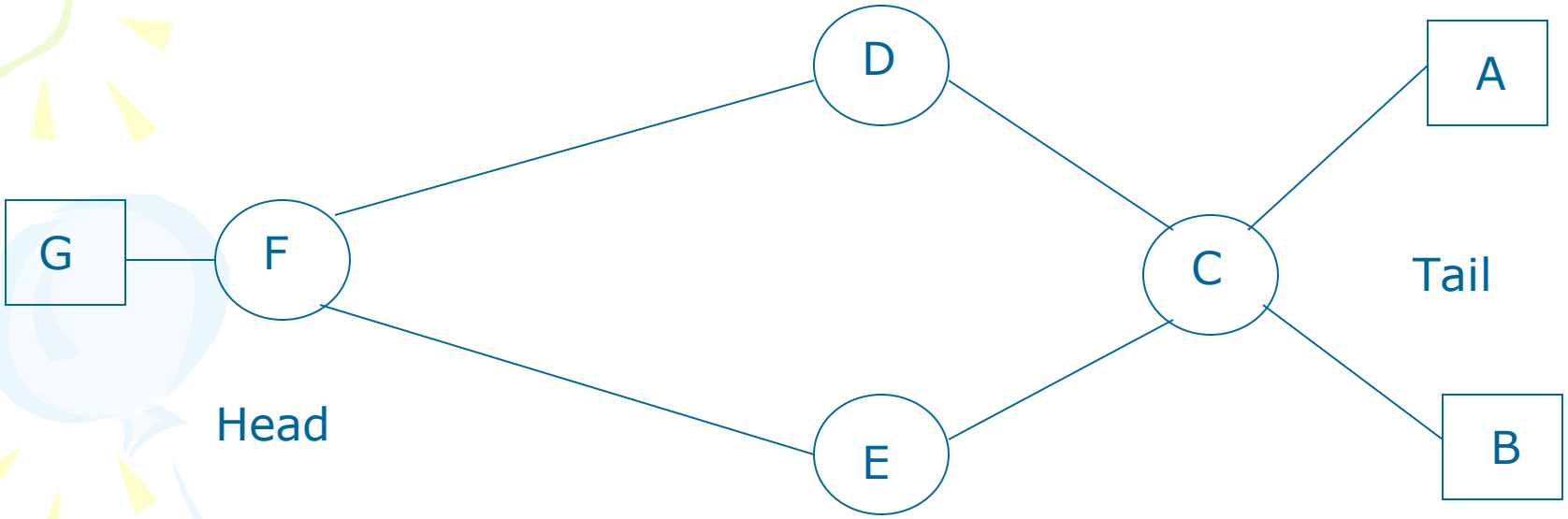
Traffic Engineering

- Concerned with the performance optimization of operational networks
- Main objective is to reduce congestion hot spots and improve resource utilization across the network through carefully managing the traffic distribution inside the network
- Cost savings that results in more efficient use of resources (e.g. bw) helps to reduce overall cost of operation for service providers.

The Fish Problem

- IP routing is based on destination and used simple metrics such as hop count or link cost for making routing decisions
- IP routing can lead to poor resource utilization
  - Can be illustrated with so called *fish problem*

The Fish Problem



The Fish Problem (cont.)

- There are two paths from A and B to G.
- But only one of the two paths (shortest path) will be used for traffic
- Leads to unbalanced traffic distribution
- Problem caused by two properties of IP routing
  - IP routing is destination based. Thus for each destination network there is typically only one path in the routing table : traffic distribution tends to be unbalanced

The Fish Problem (cont.)

- Decision making in current routing is based on local optimization : any node simply selects a path that is best from its own perspective. It does not take into account the overall system objective and does not have a global view of the network in terms of traffic distribution

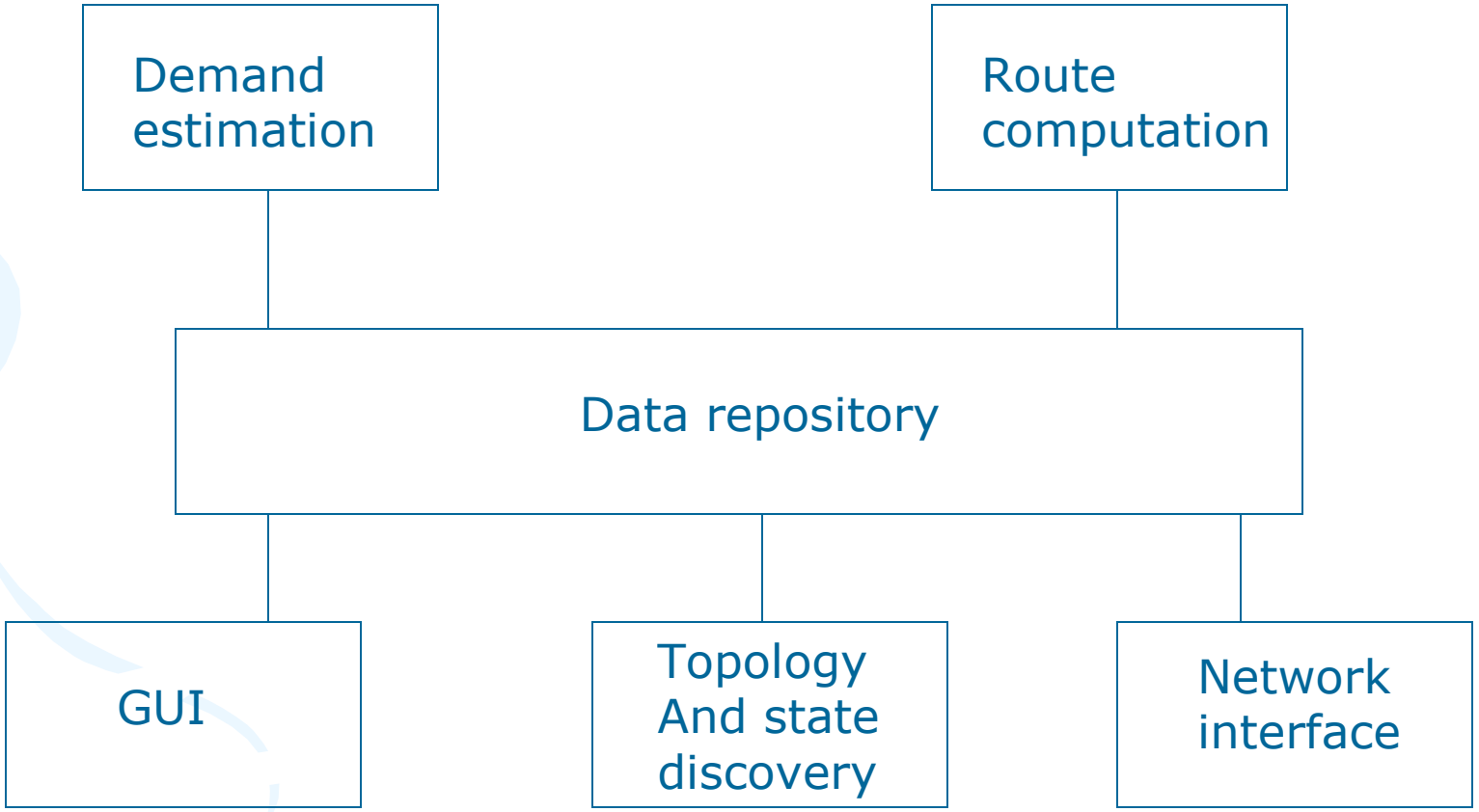
Optimization Objectives

- The main aim of TE is to improve network performance through optimization of resource utilization in the network.
- Common optimization objectives are
  - Minimizing congestion and packet losses in the network
  - Improving link utilization
  - Minimizing total delay experienced by packets
  - Increasing number of customers with the current assets

Optimization Objectives (cont.)

- ISPs would like to avoid hot spots in the network
  - Mathematically means minimize the maximum link utilization
  - Means lower total delay and loss
  - Leaves more space for future traffic growth since available bandwidth is maximized

Building Blocks



Block diagram of TE system

Data Repository

- Central DB of all shared data objects such as network topology, link state, traffic demands etc.
- Other modules in the system can store, access and exchange information thru the DB

Topology and State Discovery

- For TE system, changes in network topology and link state should be monitored
- Dynamic info. such as available bw and link utilization may be collected thru network mgmt system such as SNMP traps and polling
- Another approach is to extend routing protocol such as OSPF to broadcast link states periodically

Traffic Demand Estimation

- TE should have reasonably accurate information about the traffic demands of users
  - May be available via SLAs
  - May be estimated based on measurement
- TE typically uses aggregated traffic statistics
  - For route optimization it is necessary to know the traffic load between any pair of ingress and egress nodes

Route Computation

- Central to TE system
  - TE system should calculate routes based on traffic demands
- Route selection must be subject to multiple constraints : *constraint based routing*
- Constraint based routing can be done in two modes : off-line or on-line
- Off-line mode
  - Route computation performed for all routes periodically with current information

Route Computation (cont.)

  - Network switches over to new routes during maintenance periods
  - All routes are systematically reoptimized after changes
  - Extra delays result from adding new traffic demands to the network since route computation is only done periodically
- On-line mode
  - Route computation done in an incremental fashion as traffic demands arrive
  - Route computation module calculates the optimal route for the new demand only
  - Routes of existing demands remains the same

Route Computation (cont.)

  - Rerouting of existing traffic is minimized, but resource utilization may not be as efficient
- The two modes can be implemented together
  - Routes for new demands can be placed incrementally and after some time interval complete reoptimization is performed for all demands

Network Interface

- Network interfaces used to configure network elements, once route computation is done
- If network elements have embedded web servers, config can be done via web
- SNMP can also be used for configuration

Constraint-Based Routing

- Conventional IP routing is based on an algorithm that optimizes a particular scalar metric
- With constraint based routing path is optimal w.r.t. some scalar metric, at the same time it does not violate a set of constraints :
  - Performance constraint
    - Path with certain minimum available bw
  - Administrative constraints
    - Path that excludes certain links in the network

Constraint-Based Routing (cont.)

- Plain IP routing cannot support constraint based routing
  - Constraint-based routing requires path calculation at the source
    - Because different source may have different constraints for a path to the same destination
    - Constraints associated with a particular source router are only known to that router
    - In plain IP routing paths are computed in a distributed fashion by every router; does not take into account constraints of different sources

Constraint-Based Routing (cont.)

- When a path is determined by the source, forwarding along such a path cannot be provided using the destination-based IP forwarding
- Path computation at the source needs to have information about attributes associated with individual links (e.g. link utilization).
  - There is no mechanism to distribute this information in the network through plain IP routing

- IP routing protocol can be augmented to support these functionality

Constraint-Based Routing (cont.)

- Augmentation would include ability to compute a (constrained) path at the source
- Source would need information available locally and also from other routers in the network
- Information needed from other routers include network topology and attributes of links in the network required for constraint test

Constraint-Based Routing (cont.)

- Since any node may potentially originate traffic, this info. has to be available at every node
- Once constrained path is determined, we need a way to support forwarding along such a path : explicit routing
- May need reservation of resources along the constrained path : so a need for resource reservation mechanism along the path

Mathematical Formulation

K = set of bandwidth demands between a source and destination pair (pair of edge nodes)

X_{ij}^{k} = *percentage of k's bandwidth demand satisfied by link*(i, j)

α = maximum link utilization among all links

d_k= k^{th} bandwidth demand

s_k = source node for k^{th} demand

t_k = destination node for k^{th} demand

c_{ij}=capacity of link(i,j)

Then the following LP needs to be solved

Mathematical Formulation

Minimize α

$$\sum_{j:(i,j)\varepsilon E} X_{ij}^{k} - \sum_{j:(j,i)\varepsilon E} X_{ji}^{k} = 0 \quad for \text{ int } ermediate\ nodes$$

$$\sum_{j:(i,j)\varepsilon E} X_{ij}^{k} - \sum_{j:(j,i)\varepsilon E} X_{ji}^{k} = 1 \quad for\ i = s_k$$

$$\sum d_k X_{ij}^{k} \leq c_{ij}\alpha \qquad (i,j) \in E$$

$$0 \leq X_{ij}^{k} \leq 1, \alpha \geq 0$$

Constrained Shortest Path First (CSPF)

- CSPF finds path which satisfies the following
  - Path is optimal w.r.t. some scalar metric
  - Path does not violate a set of constraints
- Plain SPF can be easily modified to get CSPF

SPF Algorithm

- Step 1 : Initialization. Set the set of *candidate* nodes to empty. Set the SPF tree to only the root S. For each node adjacent to the root, set its path metric to the metric of the link between the root and the node. For all other nodes, set this metric to infinity
- Step 2 : Denote node just added to the SPF tree as V. For each link attached to that node, examine the node at the other end of the link. Let this other node be W.

SPF Algorithm (cont.)

- Step 2a : If W is already in the SPF tree, examine the next link attached to V
- Step 2b : else (W is not in the SPF tree), compute the path from the root to W (metric from root to V + metric from V to W). If W is not in the candidate list, then add W and set the metric from root to W to the distance computed. If W is on the candidate list and its current path metric is greater than the newly computed metric, set the path metric to the new one

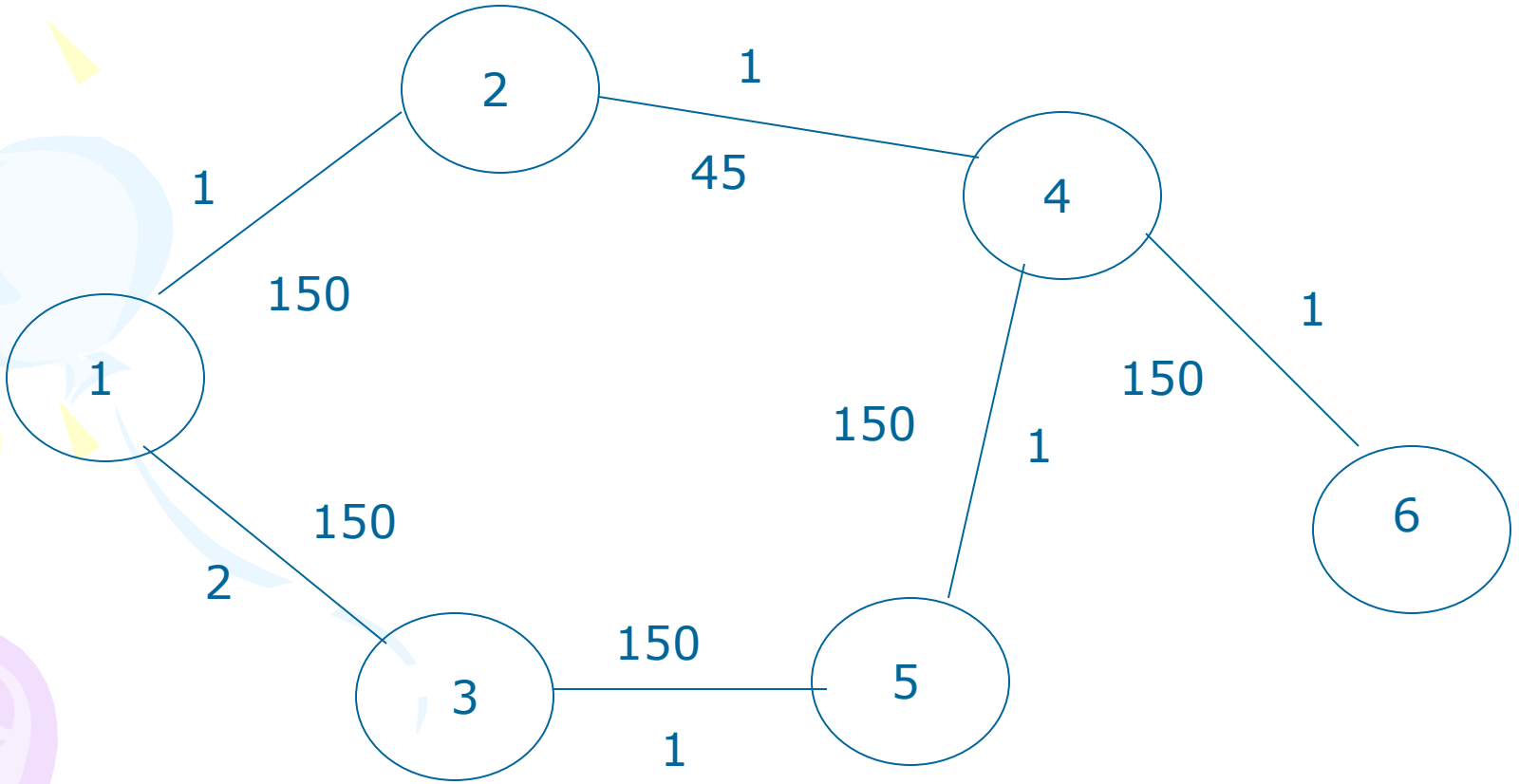
SPF Algorithm (cont.)

- Step 3: Among all nodes in candidate list, select the one with smallest path metric. Add this to the SPF tree and remove this node from candidate list. If this node is D, done. Else go back to step 2.

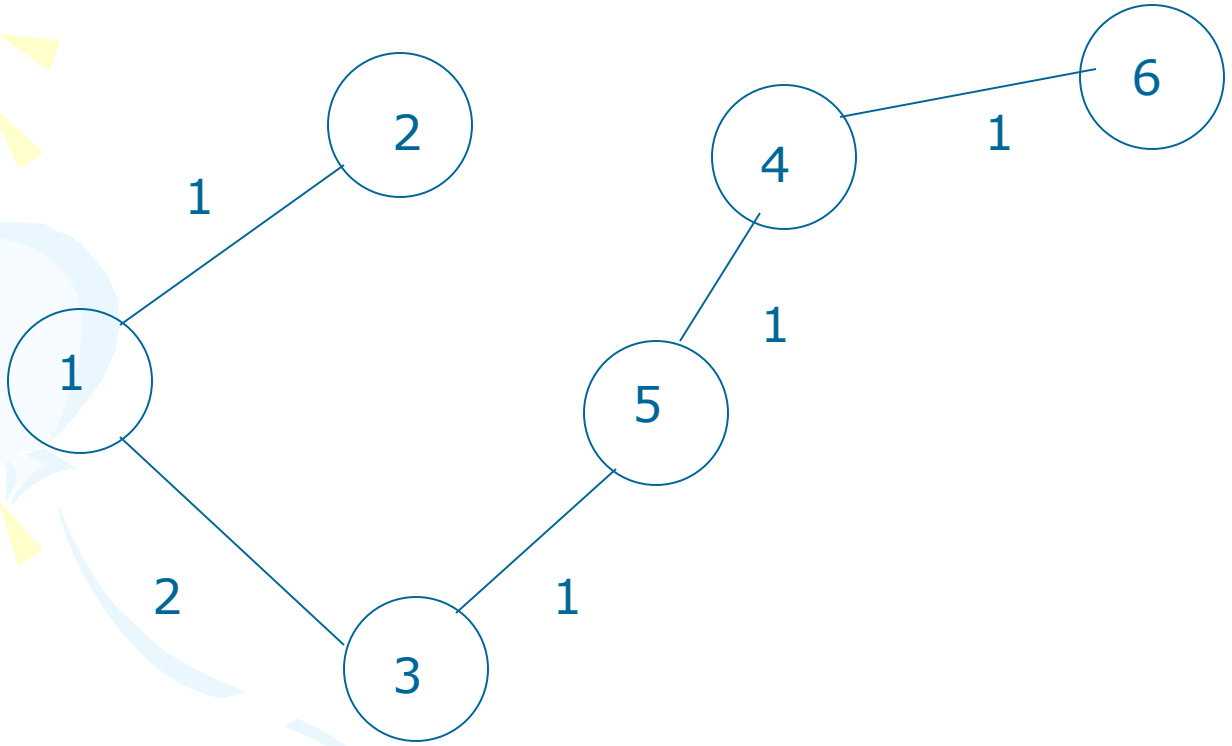
CSPF Algorithm

- SPF algo. easily modified to get CSPF
- In step 2, as links attached to V are examined, we first check whether the link satisfies the constraints.
- If all the constraints are satisfied then we examine the node W at the other end

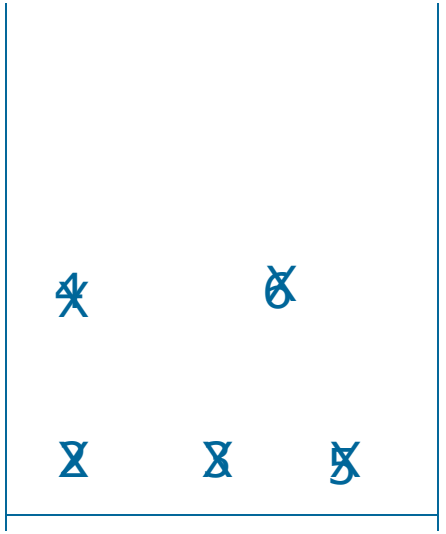
Example



Example (cont.)



CSPF tree



Candidate list

Minimum Hop (MH)

- A simple variation from shortest path routing
- All link costs are set to 1 and SPF is run
- Essentially finds the path with minimum hop

Shortest-Widest Path (SWP)

- Uses bandwidth as a metric
- Selects the paths that have largest bottleneck bandwidth
- Bottleneck bandwidth of a path is the minimum unused capacity of all the links on the path
- In case of ties (same bottleneck bw), path with minimum hops or shortest distance

Hybrid Algorithm

- Shortest path algorithm (link cost as inverse of BW) minimize total resource consumption per route
- Optimization is local : does not consider other and future demands
  - e.g. taking a longer path to avoid a bottleneck may consume more bw because of extra hops. But it would leave more bw at critical bottleneck links for future demands
- SWP is the other extreme
  - Avoids overloading any bottleneck links by maximizing the residual capacity across the network
  - May result in taking longer paths to avoid a bottleneck

Hybrid Algorithm (cont.)

- So there is a tradeoff between avoiding bottleneck links and taking a longer path
- Hybrid approach tries to balance the two objectives by assigning appropriate weights for them
- Routes are selected as the least cost paths based on the link cost metric as shown below

Hybrid Algorithm (cont.)

$$\alpha_{ij} = \frac{f_{ij} + d_k}{c_{ij}} + T * MAX[0, \frac{f_{ij} + d_k}{c_{ij}} - \alpha\ ]$$

f_{ij} = current load

c_{ij} = total capacity

α= current maximal link utilization

d_k = current demand

T = tunable parameter

α_{ij} = link cost metric

Hybrid Algorithm (cont.)

- First term : increase in link utilization
- Second term : increase in maximum utilization α caused by the route selection
- T allows different weights to the two terms

OSPF-TE

- Provides a way of describing *traffic engineering* topology and distributing them in an ospf area.
- Provides mechanism to advertise extended link attributes and build a TE database
- This TE database can be used for
  - monitoring the extended link attributes
  - local constraint-based source routing
  - global traffic engineering.

OSPF-TE

- Makes use of opaque LSA
- A new TE LSA is defined
- New Link TLVs defined for traffic engineering
  - Traffic engineering metric
  - Maximum bandwidth
  - Maximum reservable bandwidth
  - Unreserved bandwidth

OSPF-TE

- Traffic engineering metric
  - Link metric for TE purpose
    - May be different from standard ospf link metric
  - Typically assigned by network administrator
- Maximum Bandwidth
  - Max bw that can be used on a link
    - True link capacity
- Maximum reservable bandwidth
  - Max bw that may be reserved on a link
- Unreserved bandwidth
  - Bw not yet reserved
  - Initial value set to max reservable bw

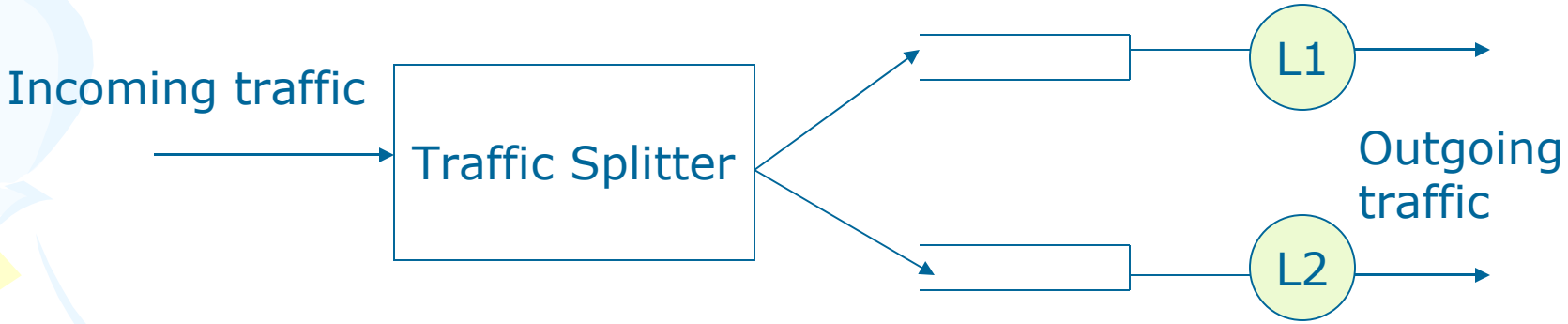
OSPF-TE

- TE LSAs advertised whenever the LSA contents change or otherwise when required by OSPF (e.g., LSA refresh)
- Implementation may set thresholds that will trigger update
  - Should be rate limited to at most one in MinLSInterval

Multipath Load Sharing

- Load sharing can substantially improve the performance of a TE scheme
- When traffic demands can be split into smaller sizes, there is more flexibility in managing them

Traffic splitting for load sharing



Traffic Splitting

- Packets are dispatched to multiple outgoing links
- May be split equally or with some proportion
- Per-packet overhead should be small (since splitting is done in packet forwarding path)
- Must maintain per-flow packet ordering
- Simple scheme of packet-by-packet round-robin has low overhead, but may cause per-flow reordering
  - May add sequence number or states to reordering, but will increase complexity

Traffic Splitting (cont.)

- Hashing based traffic splitting are stateless and easy to implement
- Hashing functions that use any combination of the five-tuple as input, per-flow ordering can be preserved

Direct Hashing

- Traffic splitter applies a hash function with a set of fields of five-tuple and uses the hash value to select the outgoing link
- Hashing of destination address : If there are N outgoing links
  - H(•) = DestIP MOD N
  - If $N = 2^k$ then effectively use the last k bits of the dest addr as an index of the outgoing link
- XOR folding of source/dest addr

Direct Hashing (cont.)

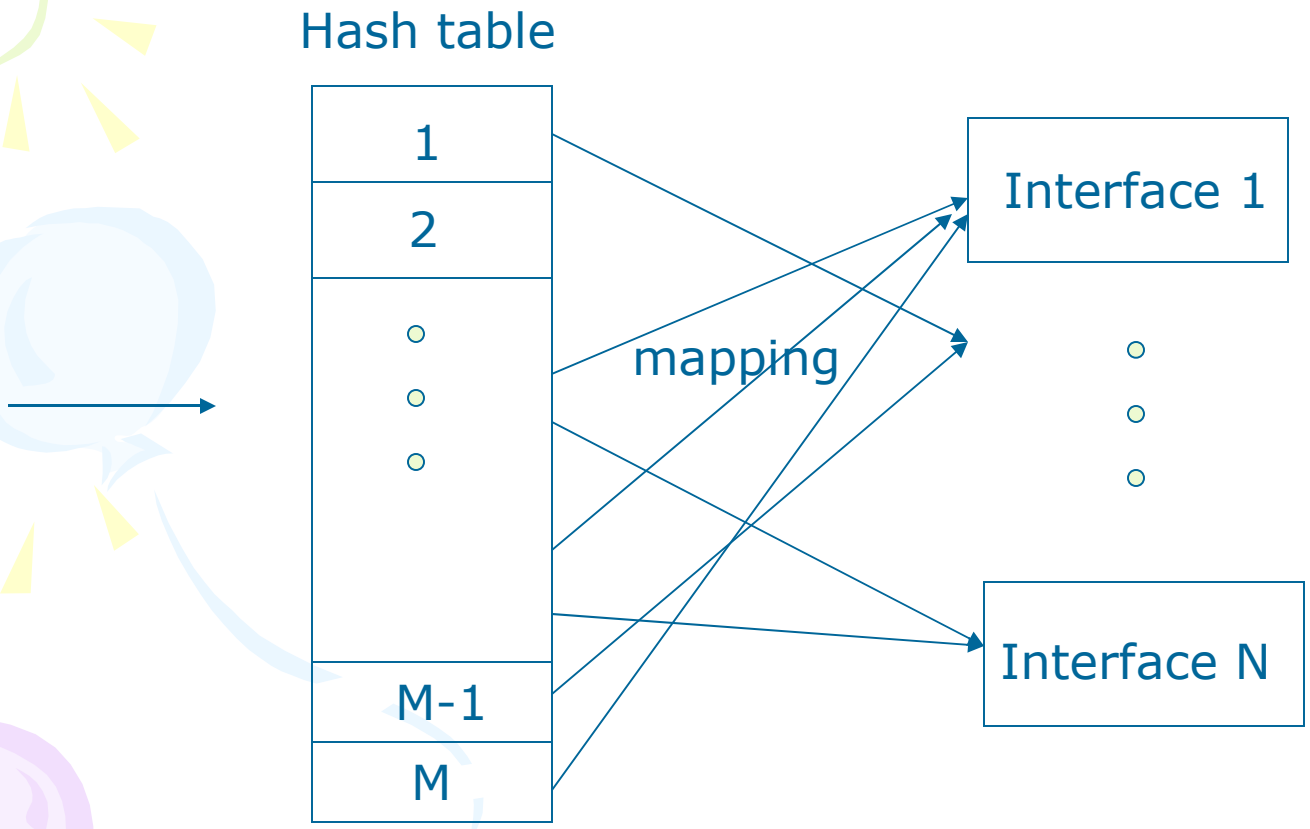
$$H(\bullet) = (D_1 \oplus D_2 \oplus D_3 \oplus D_4) \bmod N$$

- Similarly source and dest addr can be xor'ed
- CRC algorithm also can be used for hashing
  - H(•) = CRC16(5-tuple) mod N

Table-Based Hashing

- Direct hashing, though simple, can only split traffic into equal amounts to multiple outgoing paths
- If two links are of different bw, then a proportional division is desirable
- Table hashing scheme first splits the traffic into M bins (using a hash function).
- The M bins are then mapped into N outgoing links
  - Each entry in those M bins have a outbound interface entry
- Proportionality of splitting can be changed by changing the number of entries of a particular outbound interface in the M bins
- Typically M is one or two orders of magnitude larger than N
- When M=N, one-to-one mapping : becomes direct hashing

Table Based Hashing



References

- Requirements for Traffic Engineering Over MPLS – RFC 2702
- Awduche D., "MPLS and Traffic Engineering in IP Networks" – IEEE Communication magazine, Dec 1999
- Ghanwani A., et al., "Traffic Engineering Standards in IP Networks Using MPLS" - IEEE Communication magazine, Dec 1999
- Swallow G., "MPLS Advantages for Traffic Engineering" - IEEE Communication magazine, Dec 1999
- Cao Z., et al. "Performance of Hashing-Based Schemes for Internet Load Balancing – IEEE Infocom 2000.
- Jain R., "A Comparison of Hashing Schemes for Address Lookup in Computer Networks" – IEEE Trans. On Communications, Oct. 1992.
- Katz D. et al., "Traffic Engineering (TE) Extensions to OSPF Version 2" – RFC 3630