QoS Routing

Jonathan Osting
Matt Hoffman
Sheng Shen

The Goal

To guarantee a predetermined level of service to certain connections, the required resources must be reserved when the connection is established.

Such resources include CPU time, buffer space, bandwidth, etc.

Resource reservations mean that a QoS connection will not be affected by other traffic sharing the same links.

However, this also means that all routers along a path for a QoS connection must maintain state.

The Goal

Unicast - One sender and one receiver

- Find a network path that meets a set of given requirements between two end users.

Multicast - One sender with many receivers

- Find a multicast tree covering all receivers rooted at the sender, with every segment satisfying a set of given requirements.

Types of Constraints

Where *f(a)* is the value of some heuristic function *f* for a link *a*.

- Propagation Delay
- Jitter
- Cost ($, wtd link, etc)

Additive Constraints
$f(a + b) = f(a) + f(b)$

- Transmission Rate

Multiplicative Constraint
$f(a + b) = f(a) * f(b)$

- Bandwidth

Concave Composition Constraint
$f(a + b) = \min[f(a), f(b)]$

Constraint Reductions

Multiple constraints introduce additional complexity to QoS routing.

- Problem: A combination of any two of the previous constraints (except for bandwidth) is an NP-Complete problem.
- Solution: Combine different types of constraints into one.
  - Widely used in existing routing
  - Combining different types of constraints does not always work well

Example: Combination of bandwidth, delay and packet loss

Metric = Bandwidth / (Delay * Loss Probability)

How would this metric be accumulated along a path?

Constraint Reductions

By approximating an unbounded real number as a bounded integer, it is possible to greatly reduce the worst-case time complexity of a routing algorithm. For example:

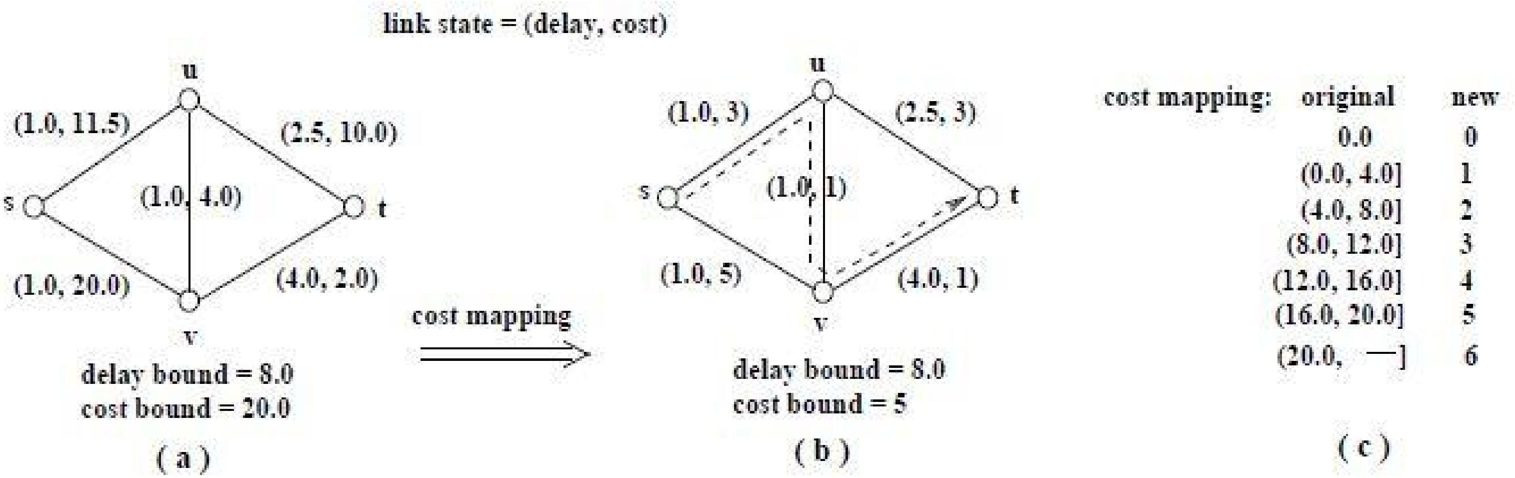


Image from "An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions"

This reduces an NP-Complete problem into an O(E + V log(V)) one, at the cost of accuracy.

QoS Problem Classes: Overview

Constrained versus Optimized

- Constrained - above or below a certain threshold
- Optimized - the min or max possible value

Path versus Link versus Tree

- Path - defined by the value across the entire end-to-end path
- Link - defined by the value between two nodes in the path
- Tree - defined by the value across the entire propagation tree (multicast)

Problem Classes: Examples

<u>Basic Examples</u>

Link-constrained: Minimum Bandwidth, reserved buffer space

Link-optimization: Best possible bandwidth

Path-constrained: Maximum latency

Path-optimization: Least-cost routing

Tree-constrained: Maximum latency multicast routing

Tree-optimization: Least-cost multicast routing **(NP-Complete)**

<u>Composite Examples</u>

Link-constrained path-optimization routing: Bandwidth-constrained least-delay

Problem Classes: Complexity

Polynomial Time Composite Classes (Unicast)

- Link-constrained + Link-optimization
- Multi-link-constrained (ex: bandwidth-buffer-constrained routing)
- Link-constrained + Path-constrained
- Path-constrained + Link-optimization

NP-Complete Problem Classes (Unicast)

- Path-constrained + Path-optimization (PCPO)
    - Delay-constrained/least-cost routing
- Multi-path-constrained (MPC)
    - Delay/delay-jitter-constrained routing (bounded delay and bounded delay-jitter)

Solution Classes

Source Routing - A complete global state is maintained at every node, the source node computes the entire path locally.

Distributed Routing - The path computation is distributed among the intermediate nodes between the source and the destination.

Hierarchical Routing - An aggregated global state is maintained instead of a complete one, retaining many advantages of both source routing and distributed routing.

Source Routing

The Good

- Very simple to conceptualize and implement.
- Avoids issues such as deadlock, loops, termination, and state desynchronization.
- More heuristics for NP-Complete routing problems are possible in comparison to distributed routing.

The Bad

- Maintaining a global state involves a large overhead that does not scale well.
- Communication overhead can be excessively high for large networks.
- Both computational time as well as packet headers are very large at the source node.

Distributed Routing

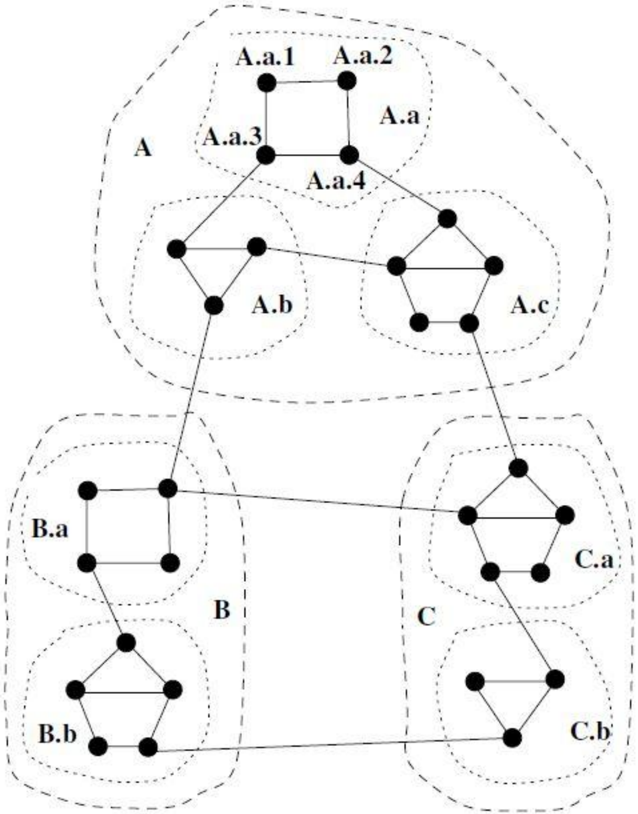
The Good

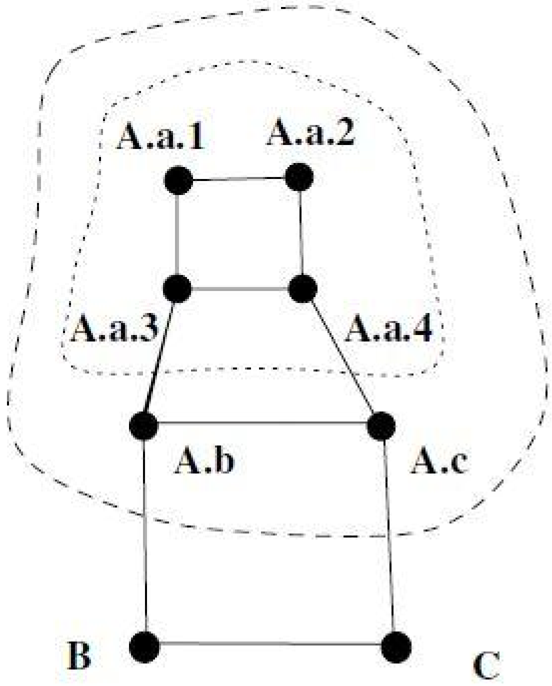
- Path computation is shared among many nodes.
- Allows multiple paths to be searched in parallel.
- Generally allows for more scalable algorithms.
- Allows a balance between amount of state information needed versus communication overhead.

The Bad

- It is very difficult to design efficient algorithms, especially for multicast routing.
- Many problems arise due to inconsistent information.
- Many algorithms depend on global state information and share many of the same issues present in source routing.

Hierarchical Routing





Images from "An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions"

Hierarchical Routing

The Good

- Allows source routing type solutions to scale in large networks.
- Allows paths to be partially abstracted and filled in by other nodes.

The Bad

- The more that the network is aggregated, the more imprecision is introduced.
- Several unsolved problems are still present in implementing QoS in a Hierarchical network.

Hierarchical Routing

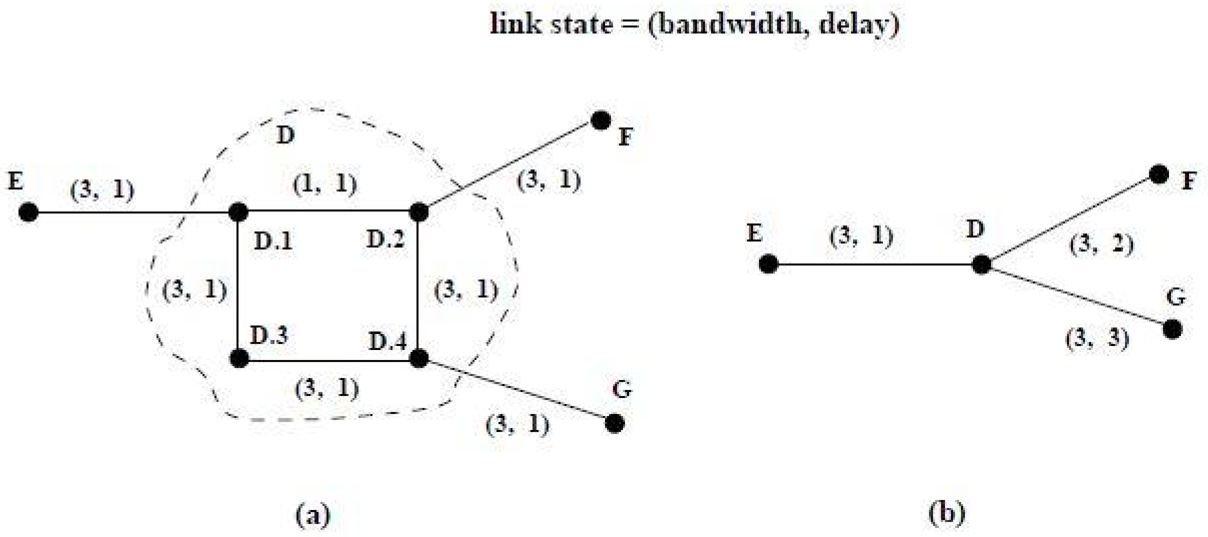


Figure 6: (a) the internal state, (b) an incorrect aggregation on link (*D*, *F*)

Image from "An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions"

Up Next: Jonathan Osting

Practical Considerations in QoS

Metric Selection

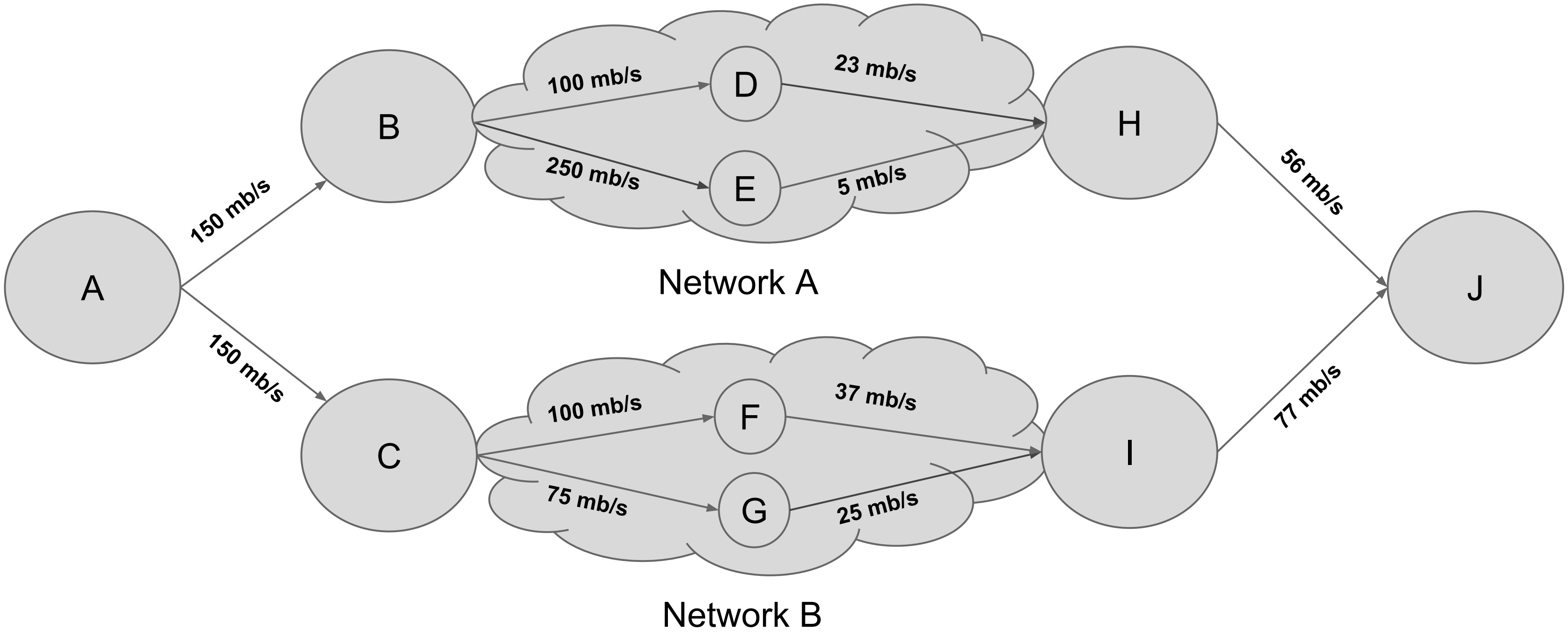
1. Metrics and constraints selected must have efficient routing algorithms
   a. Should be similar order of magnitude to current routing algorithms
   b. Should work in both a centralized and distributed framework
2. Should reflect basic network attributes and easily map to QoS requirements
3. Should be independent / orthogonal

Why is the challenging?

- Resources required are diverse and application-dependent

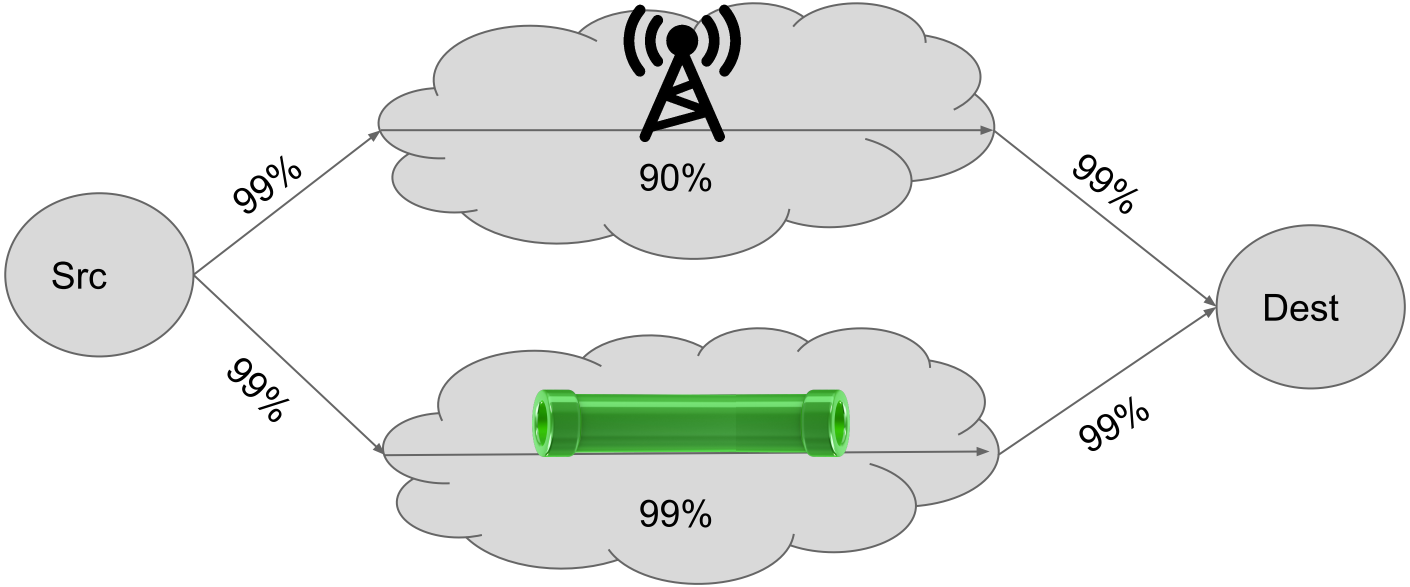
Concave Constraints

- Limiting factor is ‘weakest’ link
- Bandwidth from A-B-D-H-J = 23 mbps



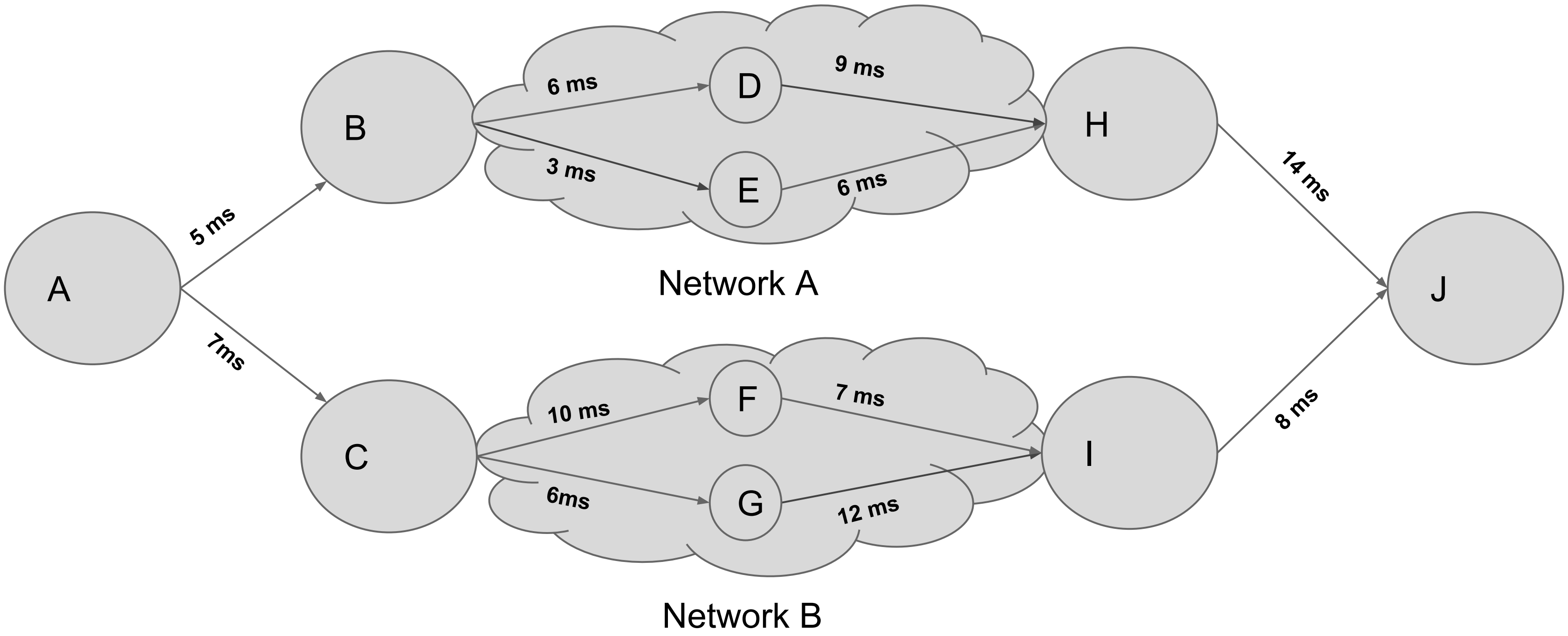
Multiplicative Constraints

- Path cost is multiple of link cost
- Loss probability can be modified to success probability
  - Success for top ‘wireless’ network = .99 * .90 * .99 = 88.21%



Additive constraints

- Path cost is added across links
- Delay from A-C-F-I-J = 32 ms



Metric Combinations

<u>Issue</u>

Finding route with any combo of additive or multiplicative metrics is NP-complete

(see Wang-Crowcroft paper for proof)

Given the metrics: bandwidth, delay, delay jitter, cost, and packet loss

<u>AND</u>

the practical limitations on metric combinations for QoS

1. What type of constraint problems can we feasibly solve?
2. Why don't we just create a new 'super' metric from a combination of these?
   a. IE - Bandwidth / (Delay * Loss Probability)

QoS Source Routing Algorithm

1. Remove links that do not satisfy the concave constraint (set distance measure to inf)
2. Initialize at the source node
3. Starting at the source node, expand the node with the shortest path
   a. If the distance to any neighbor is greater than the constraint terminate
   b. If we found a path to the destination terminate
4. Update the path lengths
5. Repeat step 3

Step 1) Set $d_{ij} = \infty$, if $b_{ij} < B$.

Step 2) Set $L = \{1\}$, $D_i = b_{1i}$ for all $i \neq 1$.

Step 3) Find $k \notin L$ so that $D_k = \min_{i \notin L} D_i$.

If $D_k > D$, no such a path can be found and the algorithm terminates.

If L contains node m, a path is found and the algorithm terminates.

$L := L \cup \{k\}$.

Step 4) For all $i \notin L$, set $D_i := \min[D_i, D_k + d_{ki}]$

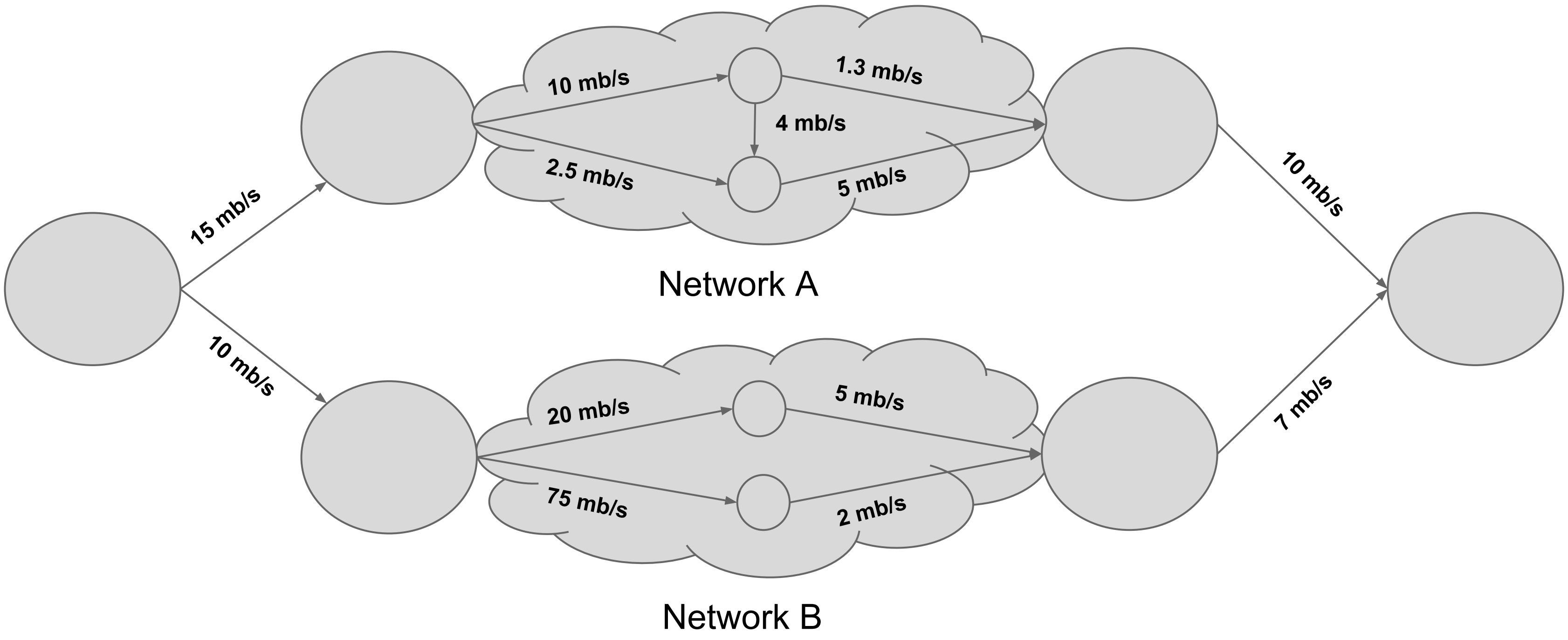
Step 5) Go to Step 3).

Step 1 est. the concave constraint

Steps 2-5 run Dijkstra’s algorithm

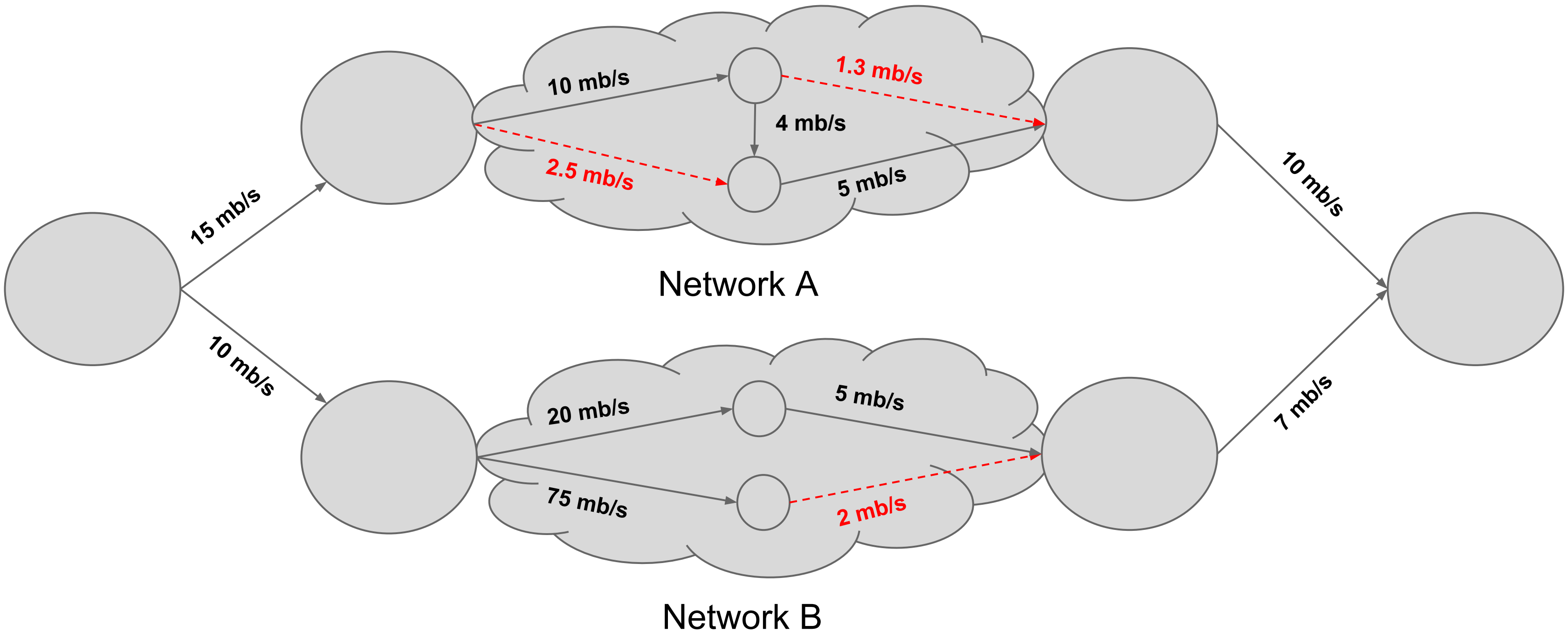
Example: QoS Source Routing

- Req: Bandwidth of 3 Mbps (Netflix)



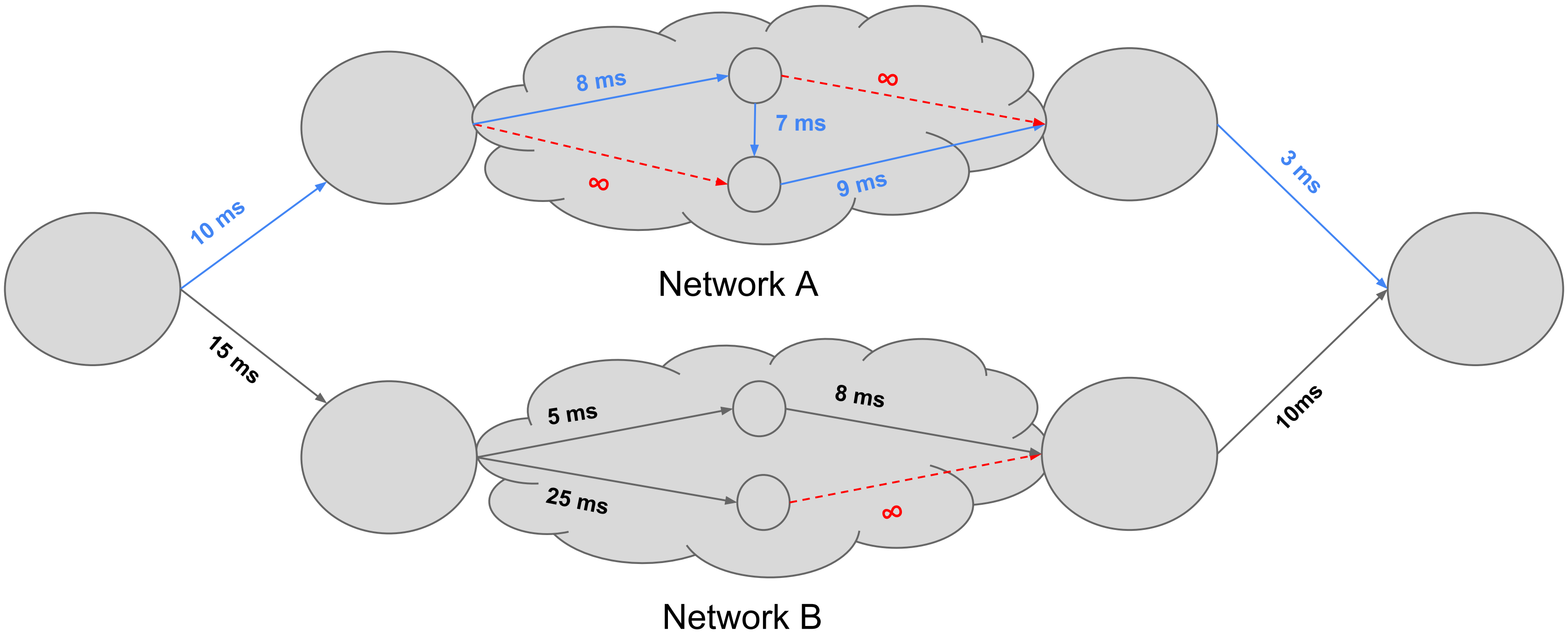
Example: Source Routing

- Eliminate routes that do not meet path (set distance/delay = ∞)



Example: Source Routing

- Find path with min shortest delay (using Dijkstra's algorithm)



Hop-by-Hop (Distributed) QoS Routing

<u>QoS Distributed Routing</u>

- Problem 1: With multiple metrics a single optimal route at a node may not exist
- Solution: Set precedence of metrics (bandwidth first, then delay)
- Problem 2: Loops may form due to distributed nature of algorithm
- Solution: Using the delay metric will avoid loops
- Result: Finds the shortest-widest route (highest bandwidth, lowest delay)

Shortest Widest Path Using Dist. Vectors

Step 1) Initially, $h = 0$ and $B_i^{(0)} = 0$, for all $i \neq 1$.

Step 2) Find set K so that $width(1, \cdots, K, i) = \max_{1 \leq j \leq N}[\min[B_j^{(h)},\ b_{ji}]], i \neq 1$.

Step 3) If K has more than one element, find $k \in K$ so that $length(1, \cdots, k, i) = \min_{1 \leq j \leq N}[D_j^{(h)} + d_{ji}], i \neq 1$.

Step 4) $B_i^{(h+1)} = width(1, \cdots, k, i)$ and $D_i^{(h+1)} = length(1, \cdots, k, i)$.

Step 5) If $h \geq A$, the algorithm is complete. Otherwise, $h = h + 1$ and go to Step 2).

$B_i^{(h)}$ - width of shortest-widest path h hops away

$D_i^{(h)}$ - length of shortest-widest path h hops away

Shortest-widest Path using Link State

Step 1) Initially, $L = \{1\}$, $B_i = b_{1i}$ and $D_i = d_{1i}$ for all $i \neq 1$.

Step 2) Find set $K \notin L$ so that $B_K = \max_{i \notin L} B_i$.

Step 3) If K has more than one element, find $k \in K$ so that $length(1, \cdots, k, i) = \min_{j \in K}[D_{(1,\cdots,j,i)}]$. $L := L \cup \{k\}$. If L contains all nodes, the algorithm is completed.

Step 4) For all $i \notin L$, set $B_i := \max[B_i, \min[B_k, b_{ki}]]$.

Step 5) Go to Step 2).

B_i - width of shortest-widest path to node i

D_i - length of shortest-widest path to node i

Discussion questions

The authors present a metric combination of bandwidth and delay in the paper.

- Given what we've presented, what other metrics could we feasibly combine using the author's method? Why might these be useful?
- What other future applications of QoS routing might become popular if it is widely available?
- Do you foresee a shift in network traffic from best-effort to QoS?
- Denial of service implications?