QoS routing

Background and Motivation

- Multimedia application have stringent QoS requirements
- Network must make resource reservation
- Requires finding paths with QoS guarantees
- Traditionally,single metric such as hop-count or delay
- QoS routing problem: Find a path that satisfies multiple constraints
- Important consideration: Scalability and Complexity

Selection of QoS routing metrics

- Efficient algorithms for path computation
  - Scalability
  - Complexity
  - Distributed computation

- Reflection of basic network characteristics
  - Mapping from QoS requirements to constraints on metrics along paths

- Orthogonality
  - To avoid redundant information
  - Redundancy may lead to interdependence among the metrics

Limitations of Single Mixed Metric

- Mix various information into a single measure, e.g.,
  - An indicator at best (Heuristic approach)
  - Different composition rules

- Multiple Metrics
  - Inherently hard problem
  - Shortest weight-constrained path is NP-Complete

Metric Composition Rules

- Computational complexity is determined by metric composition rules
- Three basic composition rules

Definition: Let $d(i,j)$ be a metric for link (i,j). For any path $p = (i,j,k,\cdots,l,m)$, we say metric d is additive if

$$d(p) = d(i,j) + d(j,k) + \cdots + d(l,m).$$

We say metric d is multiplicative if

$$d(p) = d(i,j) \times d(j,k) \times \cdots \times d(l,m).$$

We say metric d is concave if

$$d(p) = \min[d(i,j), d(j,k), \cdots, d(l,m)].$$

Examples of Routing Metrics

- Delay
- Delay Jitter
- Bandwidth
- Loss probability

- Delay and delay jitter follow additive composition rule ? (Crowcroft96)
- Bandwidth: Concave composition rule
- What about Loss probability composition rule?

Complexity Analysis

- n-additive metrics problem (Crowcroft - JSAC96)

Theorem 1: Give a network $G = (N, A)$, n additive metrics $d_1(a), d_2(a), \cdots, d_n(a)$ for each $a \in A$, two specified nodes i, m, and n positive integers $D_1, D_2, \cdots, D_n, (n \geq 2, d_i(a) \geq 0, D_i \geq 0$ for $i = 1, 2, \cdots, n)$, the problem of deciding if there is a simple path $p = (i, j, k, \cdots, l, m)$ which satisfies the following constraints $d_i(p) \leq D_i$ where $i = 1, 2, \cdots, n$ (the n additive metrics problem) is NP-complete.

- Proof by induction and reduction from Partition

Complexity Analysis

- n-Multiplicative Metrics Problem (Crowcroft - JSAC96)

Theorem 2: Give a network $G = (N, A)$, n multiplicative metrics $d_1(a), d_2(a), \cdots, d_n(a)$ for each $a \in A$, two specified nodes i, m, and n positive integers $D_1, D_2, \cdots, D_n, (n \geq 2, d_i(a) \geq 1, D_i \geq 1$ for $i = 1, 2, \cdots, n)$, the problem of deciding if there is a simple path $p = (i, j, k, \cdots, l, m)$ which satisfies the following constraints $d_i(p) \leq D_i$ where $i = 1, 2, \cdots, n$ (the n *Multiplicative Metrics Problem*) is NP-complete.

Complexity Analysis

- n-Additive and k-Multiplicative Metrics Problem (Crowcroft - JSAC96)

Theorem 3: Give a network $G = (N, A), n$ additive and k multiplicative metrics $d_1(a),\ d_2(a), \cdots, d_{n+k}(a)$ for each $a \in A$, two specified nodes i, m, and $n + k$ positive integers $D_1, D_2, \cdots, D_{n+k}, (n \geq 1, k \geq 1, d_i(a) \geq 1,\ D_i \geq 0$ for $i = 1, 2, \cdots, n,\ D_i \geq 1$ for $i = n + 1, 2, \cdots, n + k)$, the problem of deciding if there is a simple path $p = (i, j, k, \cdots, l, m)$ which satisfies the following constraints $d_i(p) \leq D_i$ where $i = 1, 2, \cdots, n+k$ (the n *Additive and k Multiplicative Metrics Problem*) is NP-complete.

- Proof by reduction from n+k Additive Metrics Problem

The Bigger Picture

- Any combination of two or more of delay, delay jitter, cost, loss probability are NP-Complete
- Only feasible combination is
  - Bandwidth and one of the four (delay, jitter, cost, loss-probability)
- Crowcroft 96:
  - Bottleneck bandwidth (Residual bandwidth, “width” of the path)
  - Propagation Delay (“length” of the path)
- Transformed QoS routing problem: Find a path in a network given the constraints on its “length” and “width”
- Classic example of tradeoff between Optimality & Complexity

Path Computation Algorithms

- Source Routing
  - Path computed on demand at the source
  - Packets forwarded according to the path in the packet
  - Centralized scheme
  - Access to full network information (Scalability ?)
  - Larger packet header
  - Initial computation delay

- Hop-by-Hop routing
  - Routing tables at each node (dynamic updates)
  - Distributed computation

Source Routing Algorithm

- Finds a path between node 1 and m that has a bandwidth no less than B and a delay no more than D
- A variation of Dijkstra's Shortest Path Algorithm

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).

- Computational Complexity: $O(N^2)$

Hop-by-Hop Routing Algorithm

- Compute the “best” path to every destination
- “Best” path may not exist at all
- Precedence among metrics
- Finding a shortest-widest path: Find a path with maximum bottleneck bandwidth (“widest” path) , and when there are more than one widest path, choose the one with shortest propagation delay.

Shortest-widest Path Algorithm (Distance-Vector)

- node1: source node
- h: number of arcs away from the source node
- $B_i^{(h)}$: width of the shortest-widest path from node 1 to node ‘i’ within ‘h’ hops
- $D_i^{(h)}$: length of the shortest-widest path from node 1 to node ‘i’ within ‘h’ hops
- $B_1^{(h)}$ = inf, $D_1^{(h)}$ = 0

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

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

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

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

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

Shortest-widest Path Algorithm (Link-states)

- node1: source node
- h: number of arcs away from the source node
- B_i : width of the shortest-widest path from node 1 to node 'i'
- D_i : length of the shortest-widest path from node 1 to node 'i'
- B_1 = inf, D_1 = 0

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).

Generalized QoS Routing with Resource Allocation

- Bashandy et al (JSAC, Feb. 2005)
  - QoS parameters as functions rather than static metrics
  - Adaptation of QoS parameters and allocated resources during path search
  - Dynamic programming algorithm that finds optimal path between source and destination and computes the amount of resources needed at each node
  - Jitter and data droppage analyses of various rate-based service disciplines

QoS Requirements and Resources

- QoS requirements
  - Maximum end-to-end jitter delay variation (J_{req})
  - Minimum percentage of data droppage /reliability (Q_{req})
  - Minimum long term average bandwidth (R_{req})
  - Maximum end-to-end delay variation (D_{req})
- Resources:
  - Buffer space
  - Bandwidth

Jitter Graph Model

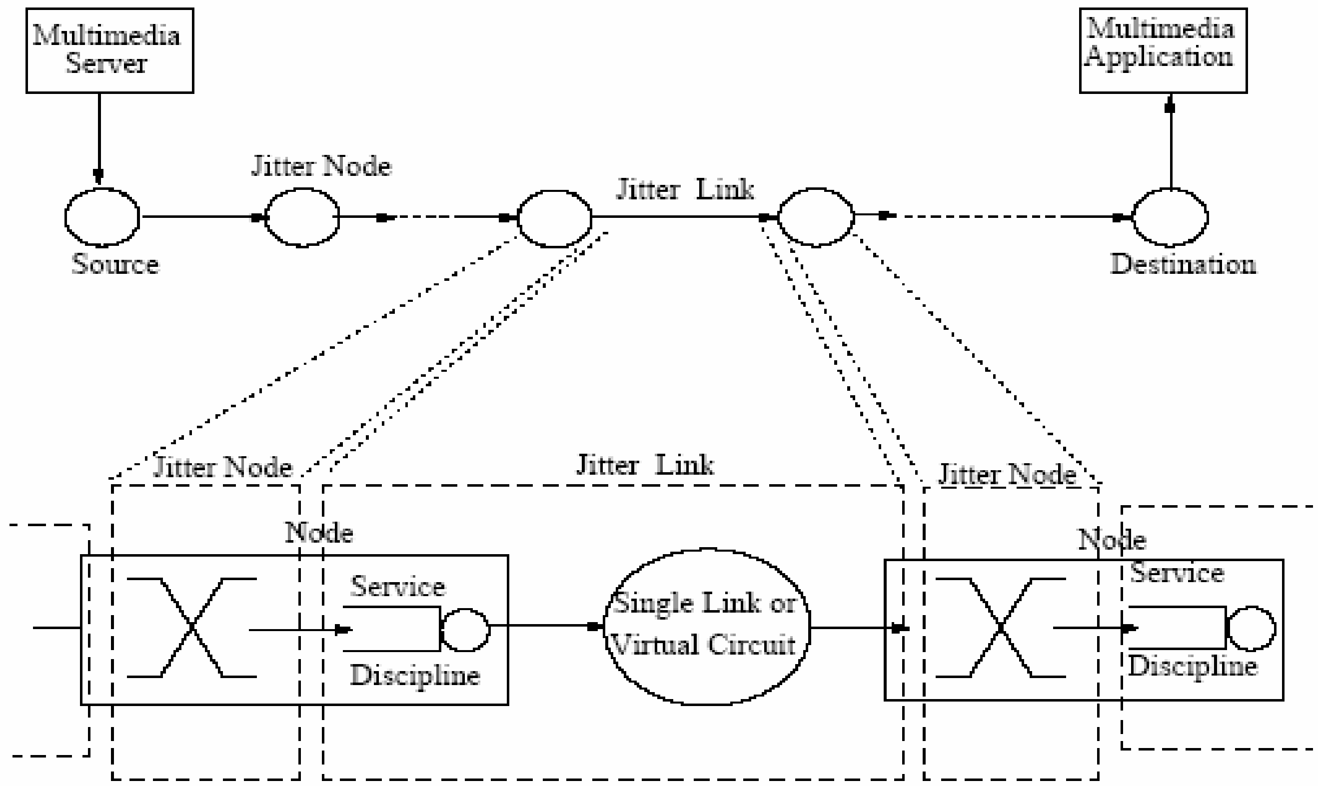

Fig. 1. The network model.

Some Notation

- Jitter link (u,v) is characterized by following:
  - Maximum propagation delay D_{uv}
  - Minimum propagation delay d_{uv}
  - Maximum transmission capacity C_{uv}
  - Reliability Q_{uv}
  - Jitter J_{uv}

- Resource used:
  - b(u,v) : buffer allocated in the service discipline of node u
  - r(u,v) : bandwidth allocated along jitter link (u,v)

- Upper bounds:
  - B(u,v) : total buffer space available for a data stream at node u
  - C_{uv} : physical link capacity
  - $C_{uv}^{max} <= C_{uv}$: maximum bandwidth available for allocation along (u,v)

QoS parameters as functions of resources

$$\begin{aligned}
0 < J_{uv} = &\quad f_{J_{uv}}(b(u,v), r(u,v)) &< \infty \\
0 \leqslant Q_{uv} = &\quad f_{Q_{uv}}(b(u,v), r(u,v)) &\leqslant 1 \\
0 < D_{uv} = &\quad f_{D_{uv}}(b(u,v), r(u,v)) &< \infty
\end{aligned}$$

$$\begin{aligned}
&J_p(p) = \\
&\quad f_J(b(v_0,v_1), r(v_0,v_1), \ldots, b(v_{n-1},v_n), r(v_{n-1},v_n)) \\
&Q_p(p) = \\
&\quad f_Q(b(v_0,v_1), r(v_0,v_1), \ldots, b(v_{n-1},v_n), r(v_{n-1},v_n)) \\
&D_p(p) = \\
&\quad f_D(b(v_0,v_1), r(v_0,v_1), \ldots, b(v_{n-1},v_n), r(v_{n-1},v_n)) \\
&r_p(p) = \\
&\quad f_R(b(v_0,v_1), r(v_0,v_1), \ldots, b(v_{n-1},v_n), r(v_{n-1},v_n))
\end{aligned}$$

Path satisfying QoS requirements

- A path p is said to satisfy the QoS requirements of the stream if :

$$J_p(p) \leqslant J_{\text{req}},$$
$$Q_p(p) \geqslant Q_{\text{req}},$$
$$D_p(p) \leqslant D_{\text{req}}, \text{ and}$$
$$r_p(p) \geqslant R_{\text{req}}.$$

- Log-reliability :

$$Z_{uv} = -\ln(Q_{uv}),$$

$$Z_p(p) = -\ln(Q_p(p)).$$

Path Resource Problem and Feasible path

For a given path path p_j, the *path-resource problem* is defined as follows:

$$\begin{aligned} Z_p^*(p_j) \quad &= \min\Big\{Z_p(p_j)\Big\}, \\ \text{Subject to:} \quad & J_p(p_j) \leqslant J_{\text{req}}, \\ & 0 \leqslant b(u,v) \leqslant B(u,v) \quad \forall (u,v) \in p_j, \\ & R_{\text{req}} \leqslant r(u,v) \leqslant C_{uv}^{\max} \quad \forall (u,v) \in p_j. \end{aligned} \tag{7}$$

A path p is said to be *feasible* for the QoS requirements specified by $(R_{\text{req}}, J_{\text{req}})$ if the feasible set in the path-resource problem defined in equation (7) for the path p is non-empty.

QoS Routing with Resource Allocation Problem

Let $P(s,d)$ be the set of all paths between the nodes s and d. The problem of *QoS routing with resource allocation* is defined as follows:

$$\zeta^*(s,d) = \min_{p_j \in P(s,d)} \left\{ Z_p^*(p_j) \right\}. \quad (8)$$

QoS Routing Algorithm

Pre-Process(G)
▶ Initialization Loop
For each $x \in \text{adj}(s)$ $z^1(s,x) := \text{Path-Opt}(G,x,s)$
For each $x \in \text{adj}^{-1}(s)$ $\pi^1(x) := x$
For each $x \notin \text{adj}(s)$ $z^1(s,x) := \infty$
▶ Main Loop
for $k := 1 \to |V| - 2$
  for each node $x \in V - \{s\}$
    $z^{k+1}(s,x) := z^k(s,x)$
    $\pi^{k+1}(x) := \pi^k(x)$
    for each node $w \in \text{adj}^{-1}(x)$
      $\pi_{old} := \pi^{k+1}(x)$
      $\pi^{k+1}(x) := w$
      $temp := \text{Path-Opt}(G,x,s)$
      if $temp < z^{k+1}(s,x)$
        $z^{k+1}(s,x) := temp$
     else
        $\pi^{k+1}(x) := \pi_{old}$

Time Complexity : $O(K|V||E| + I)$

Application to Rate-based Service Disciplines

– Generalized Processor Sharing (GPS)

– Packet by Packet GPS (PGPS)

– Worst-case Fair Weighted Fair Queuing (WF^2Q)

– Self-clocked Fair Queuing (SCFQ)

GPS

$$Q_p^{(i)*}(p(n)) = \max\left\{\min\left\{\frac{\min\limits_{1\leqslant j\leqslant n}\{b^{(i)}(v_{j-1},v_j)\}}{\sigma^{(i)}},1\right\}\right\},$$

Subject to:

$$J_p^{(i)}(p(n)) = \frac{\min\left\{\sum_{j=1}^{n}\{b^{(i)}(v_{j-1},v_j)\},\sigma^{(i)}\right\}}{r_p^{(i)}(p(n))} \leqslant J_{\text{req}},$$

$$0 < b^{(i)}(v_{j-1},v_j) \leqslant \mathbf{B}_(v_{j-1},v_j), \quad j \in \{1,\ldots,n\},$$

$$\rho^{(i)} \leqslant r_p^{(i)}(p(n)) \leqslant \min_{1\leqslant j\leqslant n} C_{v_{j-1},v_j}^{\max}.$$

PGPS and WF^{2}Q

$$Q_p^{(i)*}(p(n)) = \max\left\{\min\left\{\min_{1\leqslant j\leqslant n}\left\{\frac{b^{(i)}(v_{j-1},v_j)}{\sigma^{(i)}+jL}\right\},1\right\}\right\},$$

Subject to:

$$J_p^{(i)}(p(n)) = \frac{q^{(i)*}(n)}{r_p^{(i)}(p(n))} \leqslant J_{\text{req}},$$

$$0 < b^{(i)}(v_{j-1},v_j) \leqslant B^{(i)}(v_{j-1},v_j), \quad j \in \{1,\ldots,n\},$$

$$\rho^{(i)} \leqslant r_p^{(i)}(p(n)) \leqslant \min_{1\leqslant j\leqslant n} C_{v_{j-1},v_j}^{\max}.$$

Self-Clocked Fair Queuing (SCFQ)

$$Q_p^{(i)*}(p(n)) = \max\left\{ \min\left\{ \min_{1 \leqslant j \leqslant n} \left\{ \frac{\{b^{(i)}(v_{j-1}, v_j)}{\sigma^{(i)} + jL} \right\}, 1 \right\} \right\};$$

Subject to:

$$J_p^{(i)}(p(n)) = \frac{q^{(i)*}(n)}{r_p^{(i)}(p(n))} + \sum_{j=1}^{n} \frac{(K_{v_{j-1}v_j} - 1)L}{C_{v_{j-1}v_j}} \leqslant J_{\text{req}},$$

$$0 < b^{(i)}(v_{j-1}, v_j) \leqslant B^{(i)}(v_{j-1}, v_j), \quad j \in \{1, \ldots, n\},$$

$$\rho^{(i)} \leqslant r_p^{(i)}(p(n)) \leqslant \min_{1 \leqslant j \leqslant n} C_{v_{j-1}, v_j}^{\max}.$$

Extension to Multicast Routing (W. Sheikh et al, ICC 2005, 2207)

- Finding the optimal multicast tree
- Steiner Arborescence problem
- NP-Complete
- Presented a 2-approximate solution
- Proved the bound