Link scheduling use rate-proportional scheduling algorithm

WFQ: weighted fair queuing

Assume that the traffic source is constrained by a a token bucket (σ, b) - b: token bucket size

- σ: long term rate

For a given ρ with n hops and the link capacity C_i the provable worst case end-to-end delay bound is given by

- $D(p, r, b) = \frac{b}{r} + n \cdot \frac{L_{max}}{r} + \sum_{i=1}^{n} \frac{L_{max}}{C_i} + \sum_{i=1}^{n} prop_i$

where: $r \geq \rho$ is the bandwith to be reserved for i

L_{max} is the maximum packet size

$prop_i$ propagation delay on link i

- delay jitter bound

$$J(p, r, b) = \frac{b}{r} + \frac{n \cdot L_{max}}{r}$$

- buffer space requirement at the h-th hop/link

$$B(p, b, h) = b + h \cdot L_{max}$$

Routing Problem I $\boxed{D\_r}$

Given a delay bound d, a leaky bucket (σ, b) and a bandwidth r $(r > \sigma)$ to reserve,

find a path with $r \le R_j$ $(\forall j \in P)$ such that

$D(P, r, b) \le d$ where R_j is the residual bandwidth of link j

- Problem D_r is solvable by both the Dijkstra's and Bellman-Ford shortest path algorithms
- If there are feasible paths, we can select one with the minimum delay

Algorithm:

1. Find a shortest path P using Dijkstra's algorithm (considering only those links with $R_i \geq \sigma$) using the length function

$$l(i) = \frac{L_{max}}{r} + \frac{L_{max}}{c_i} + prop_i$$

$$l(P) = \frac{b}{r} + \sum_{j \in P} l(j)$$

2. If $l(P) > d$ then there is <u>NO</u> feasible path

otherwise P is a path with the minimum delay

Routing Problem II $\boxed{D}$

Given a delay bound d and a leaky bucket (σ, b) find a path P, such that $D(P, r, b) \le d$ for some r with $\sigma \le r \le R_j$ $(\forall j \in P)$, where R_j is the link residual bandwidth

1. the bandwidth r to be reserved is unknown
2. For a given path P, the delay can be reduced by increasing r
3. the maximum reservable bandwidth on path P is $\min\{R_j \mid j \in P\}$
4. We use an iterative approach

   (a) for every $r = R_K$ of some link k in the net

$$l_r(i) = \frac{L_{max}}{r} + \frac{L_{max}}{C_i} + Prop_i$$

$$l_r(P) = \frac{b}{r} + \sum_{j \in P} l_r(j)$$

   (b) find a shortest path P_r such that there is no link j in P_r whose residual bandwidth $R_j < r$

(C) among all paths, find a path whose $l_r(p)$ is minimum $\le d$

Routing Problem III (D-J-r)

Given a delay bound d, a jitter bound j, a leaky bucket (σ, b), a bandwidth r $(r > \sigma)$ to reserve, find a path P with $r \le R_j$ $(\forall j \in P)$ $D(P, r, b) \le d$, $J(P, r, b) \le j$, where R_j is the residual bandwidth of link j

Algorithm :

- check jitter bound

$$\frac{b}{r} + n \cdot \frac{L_{max}}{r} \le j$$

$$\Rightarrow \quad n \le \left\lfloor \frac{r \cdot j - b}{L_{max}} \right\rfloor = N$$

thus for any path, as long as its hop count is no more than $N = \left\lfloor \frac{r \cdot j - b}{L_{max}} \right\rfloor$

it will meet the delay-jitter bound;

- Next, we find a path P with $l(P) \le d$ and no more than N hops

$$l(i) = \frac{L_{max}}{r} + \frac{L_{max}}{C_i} + prop_i$$

$$l(P) = \frac{b}{r} + \sum_{j \in P} l(j)$$

This problem can be solved by the Bellman-Ford algorithm

The Bellman-Ford algorithm finds a shortest path step-by-step with increasing hop count:

At the i-th step, a shortest path with at most i hops is found

The first feasible path found is the one with the minimum hop count.

Routing Problem IV (D-J)

Given a delay bound d, a delay jitter bound j and a leaky bucket (σ, b), find a path that $D(P, r, b) \le d$ and $J(p, r, b) \le j$ for some r with $\sigma \le r \le R_j$ $(\forall j \in P)$ where R_j is the residual bandwidth of link j

Algorithm 1. iterate the Bellman-Ford shortest-path algorithm over all possible values of the link residual bandwidth R_K

2. at each iteration of $r = R_k$ the length function l_r

$$l_r(i) = \frac{L_{max}}{r} + \frac{L_{max}}{C_i} + prop_i$$

$$l_r(p) = \frac{b}{r} + \sum_{j \in p} l_r(j)$$

3. Use hop count $N_r = \left\lfloor \frac{r \cdot J - b}{L_{max}} \right\rfloor$ to control the number of steps

4. Only those links with $R_i \geq r$ are considered in each step