Multipath Network Flows: Bounded Buffers and Jitter

Gruia Călinescu

Illinois Institute of Technology

Joint work with Tricha Anjali, Alexander Fortin, Sanjiv Kapoor, Nandakiran Kirubanandan, Sutep Tongngam

Multipath Routing

- increased throughput
- reduce congestion
- improved QoS/fault tolerance
- higher security
- **But** packet disordering and "jitter"

Requires buffers at receiver. Allowed by MPLS.

Multipath Routing - Optimization

Given: digraph $G = (V, E)$, capacity function $c(u, v) : E \rightarrow R^+$, length (delay) function $l(u, v) : E \rightarrow R^+$, source s, and sink t.

For flow path P from s to t, define the total length (delay) path, $L(P)$, as:

$$L(P) = \sum_{e=(u,v) \in P} l(u, v)$$

Generate a set of s-t paths $P_1, P_2, ..., P_k$ with flow values $f_1, f_2, ..., f_k$. Allowed by MPLS.

Multipath Routing - Constraints

$\sum_{i=1}^{k} f_i = \gamma$ (demand constraint)
$\sum_{i,e=(u,v)\in P_i} f_i \leq c(u,v);\ \forall e=(u,v) \in E$
$L(P_i) \leq L;\ \forall P_i$ (path length (delay) constraint)

Bounded Jitter MultiPath Flow (BJMPF):

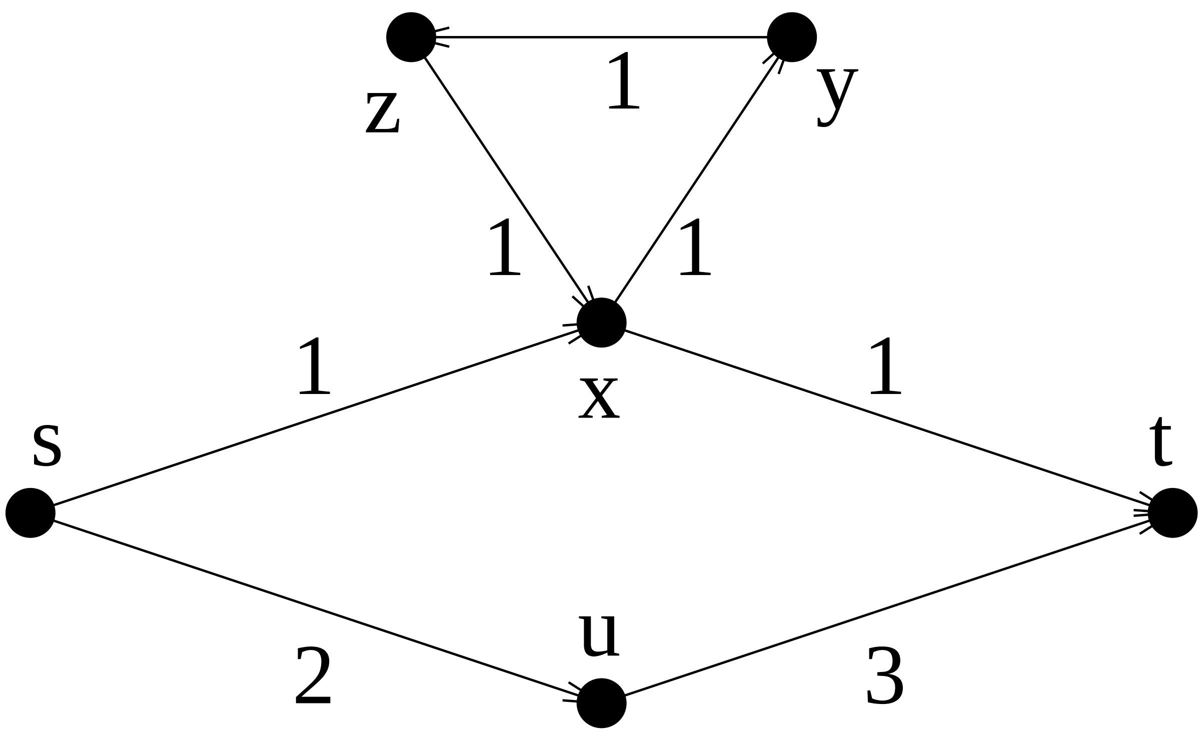
$$L(P_i) - L(P_j) \leq \delta;\ \forall P_i, P_j$$

Fixed Buffer MultiPath Flow (FBMPF):

$$\sum_i f_i * (L(P_1) - L(P_i)) \leq B.$$

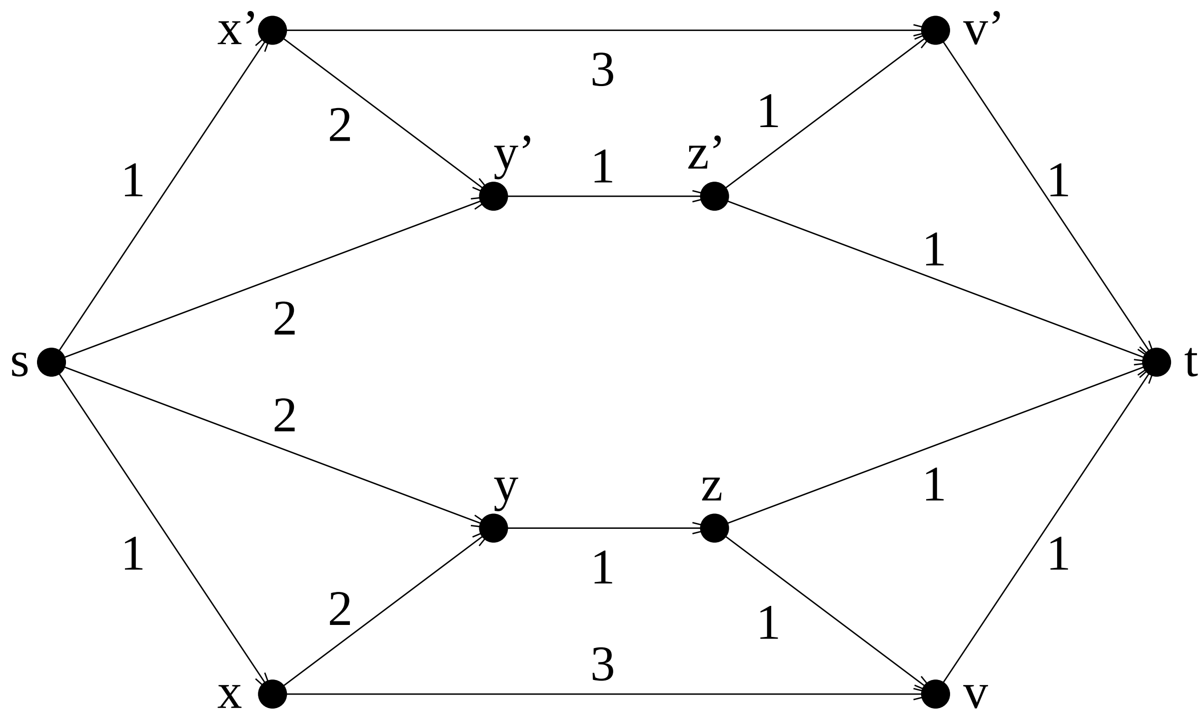
Here γ, L, and δ or B are given.

Cycles are good



Capacities are 1, delays are noted on edges. One can achieve jitter and/or buffer 0 with cycles only.

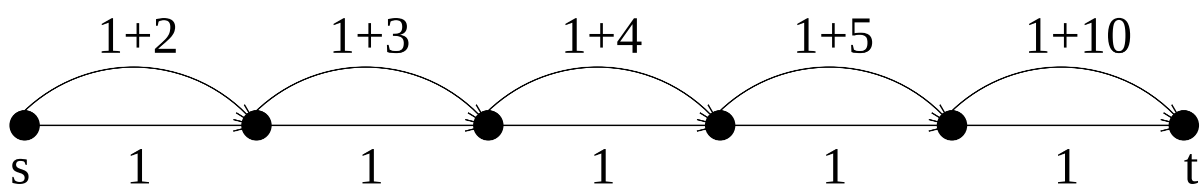
Example without integral solutions



Capacities are 1, delays are noted on edges. If $\gamma = 3$, and either $\delta = 0$ (for BJMPF) or $B = 0$ (for FBMPF), then the only solution uses each of six paths carrying 0.5 units of flow.

NP-Completeness

Reduction from Partition.



If the instance of Partition is $\{2, 3, 4, 5, 10\}$, then we get the graph above with $s = v_0$ and t=v_5. This instance can be partitioned into $A = \{3, 4, 5\}$ and $D = \{2, 10\}$, and the multipath solution uses the path which uses the first and last upper edges (with the others lower), and the path with first and last lower edges (with the others higher) - making both jitter and buffer size 0.

Bounded Jitter

By rounding: link delay $l : E \rightarrow \{1, 2, \ldots, k\}$ Algorithm polynomial in n, m, k, and U, where $U = \frac{\max_{e \in E} c(e)}{\min_{e \in E} c(e)}$.

We try all L in a range. Insist that all paths P_i used by the solution have length (delay) between $L - \delta$ and L. $\mathcal{P}(L, \delta)$ the sets of such paths. Now:

Given $N = (V, E, c, l, s, t)$ and $L, \delta, \gamma \in \mathbb{Z}_+$, find $q \in \mathbb{Z}_+$ and for each $i \in \{1, 2, \ldots, q\}$, a path $P_i \in \mathcal{P}(L, \delta)$ and $f_i > 0$ such that:

$$\sum_{i=1}^{q} f_i = \gamma$$

$$\sum_{i=1}^{q} f_i m(i, e) \leq c(e), \quad \forall e \in E$$

$m(i, e)$ is the number of times path P_i uses edge e.

Bounded Jitter - Packing LP

Linear program, exponentially many variables:

$$\text{maximize} \sum_{P \in \mathcal{P}(L,\delta)} f_P$$

$$\text{subject to} \sum_{P \in \mathcal{P}(L,\delta)} f_P \cdot m(P,e) \leq c(e) \quad \forall e \in E \quad (1)$$

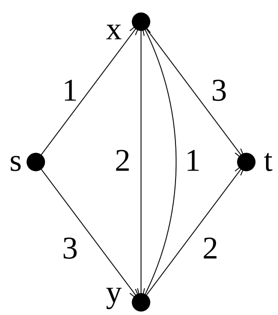
$$f_P \geq 0 \quad \forall P \in \mathcal{P}(L,\delta) \quad (2)$$

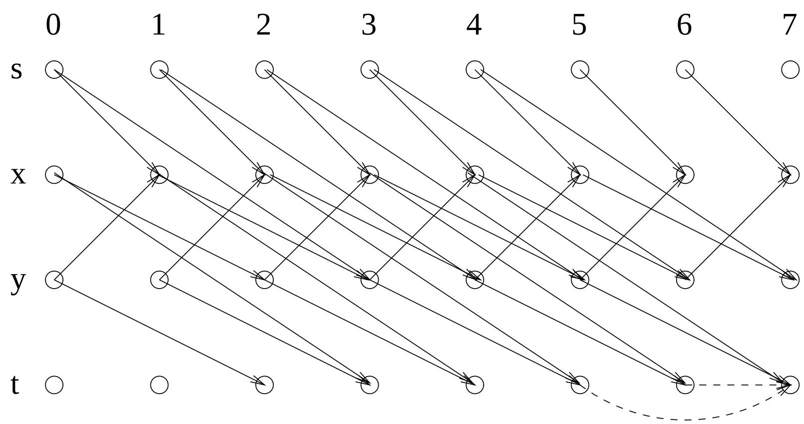
$m(P,e)$ is the number of times path P uses edge e. If the objective function is at least γ, the paths P with $f_P > 0$ give a feasible solution to BJMPF.

Resembles but not Max-Flow. Polynomial in $n, m, L, \log U$ solution: layered construction.

Instead of γ, get $\gamma(1-\epsilon)$ combinatorially using the Garg-Könemann algorithm.

GK asks for "min-size column"; for us it means finding, for a given vector $y(e)\ : e \in E$, a path $P \in \mathcal{P}(L,\delta)$ minimizing $y(P) \ := \ \sum_{e\in P} m(P,e)y(e)$. Shortest paths in the layered graph!





From $N = (V, E, s, t)$ (left, the numbers giving the length of each edge), the layered graph $\hat{N}$ is right with $L = 7$ and $\delta - 2$. The path s, x, y, x, t in N appears as s_0, x_1, y_3, x_4, t_7 in $\hat{N}$, and the path s, y, t in N appears as s_0, y_3, t_5, t_7 in $\hat{N}$.

Bounded Jitter: Range of L

Using volume considerations:

$$L \leq nk + k \frac{\sum_{e \in E} c(e)}{\min_{e \in E} c(e)}$$

We obtain an algorithm producing $(1-\epsilon)\gamma$ flow with total running time $O((1/\epsilon)^2 m^2 (kmU)^2) = O((1/\epsilon)^2 m^4 k^2 U^2)$.

Fixed Buffer

By rounding: link delay $l : E \to \{1, 2, \ldots, k\}$ Algorithm polynomial in n, m, k, and U.

We try all L in a range; this will be length of the longest used path. $\mathcal{P}(L)$ the set paths of length at most L. Now:

Given $N = (V, E, c, l, s, t)$ and $L, B, \gamma \in \mathbb{Z}_+$, find $q \in \mathbb{Z}_+$ and for each $i \in \{1, 2, \ldots, q\}$, a path $P_i \in \mathcal{P}(L)$ and $f_i > 0$ such that:

$$l(P_1) = L; \quad \sum_{i=1}^{q} f_i = \gamma$$

$$\sum_{i=1}^{q} f_i(L - L(P_i)) \leq B$$

$$\sum_{i=1}^{q} f_i m(i, e) \leq c(e) \ \ \forall e \in E$$

We solve the above problem by constructing a linear program which resembles the minimum cost flow problem on a layered network. Similar constructions appeared where given McCormick 1996, Banner-Orda 2005 for variations of max-flow.

For the range of L, using volume:

$$L \leq nk + k\frac{\sum_{e \in E} c(e)}{\gamma}$$

Algorithm: polynomial in k, m, U, where as before $U = \frac{\max_{e \in E} c(e)}{\min_{e \in E} c(e)}$.

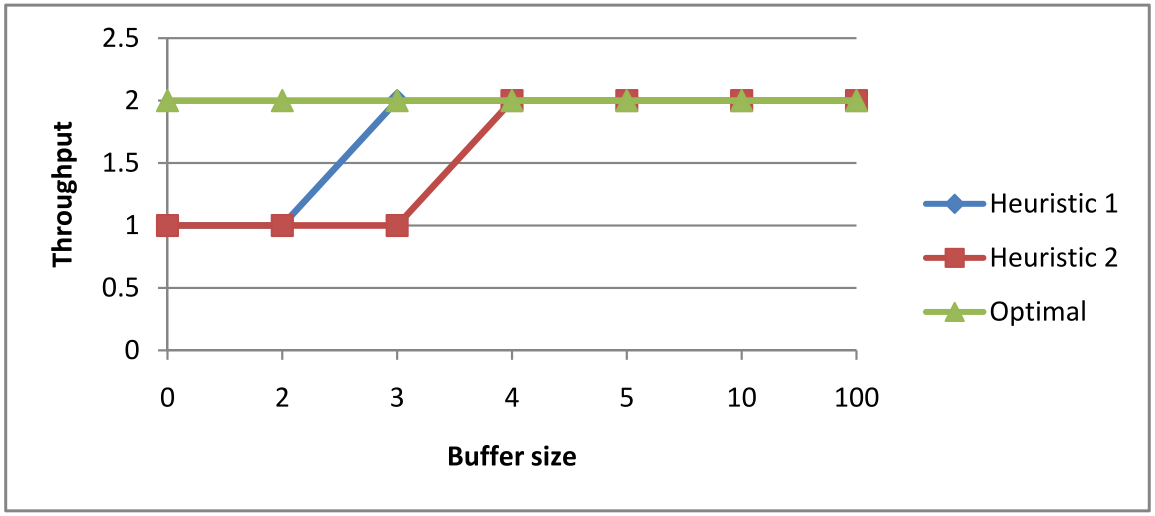
Heuristics

Our (greedy) heuristics aim to maximize the routed flow subject to either jitter or buffer constraints, or both.

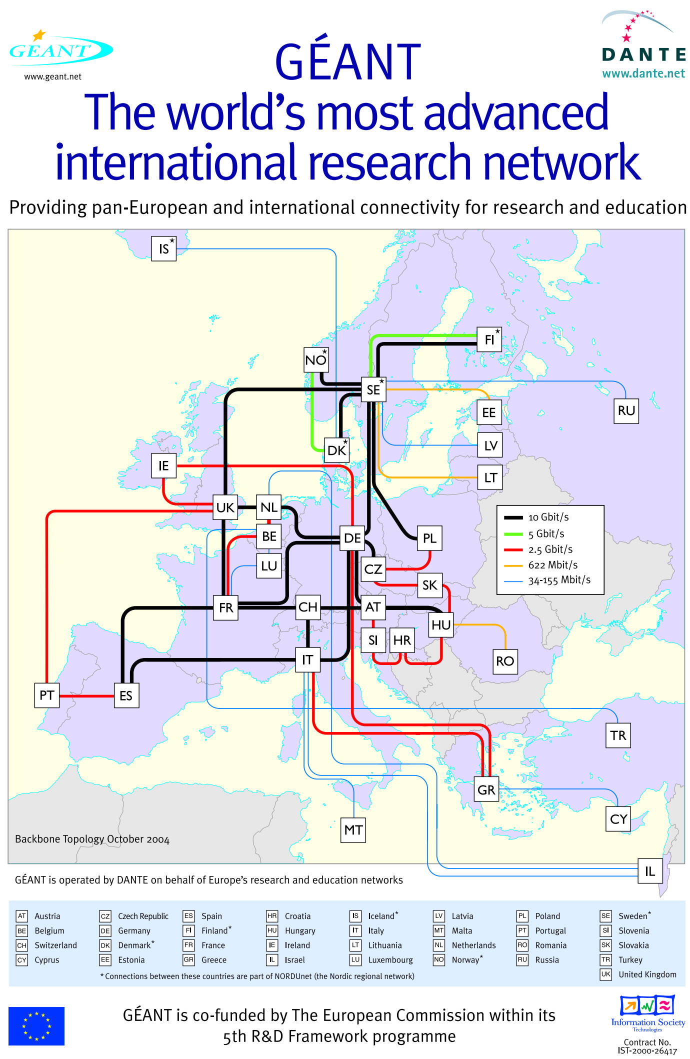
We use Shier's 1976 algorithm to find the k shortest paths (by length/delay) in the network. Repeated nodes are allowed in each of the paths.

First heuristic: sort the paths by length. Second: by (bottleneck) capacity.

Go through the paths in sorted order. Push the maximum flow on this path, subject to jitter and/or buffer constraints. For the first heuristic, if this flow is zero, we stop.

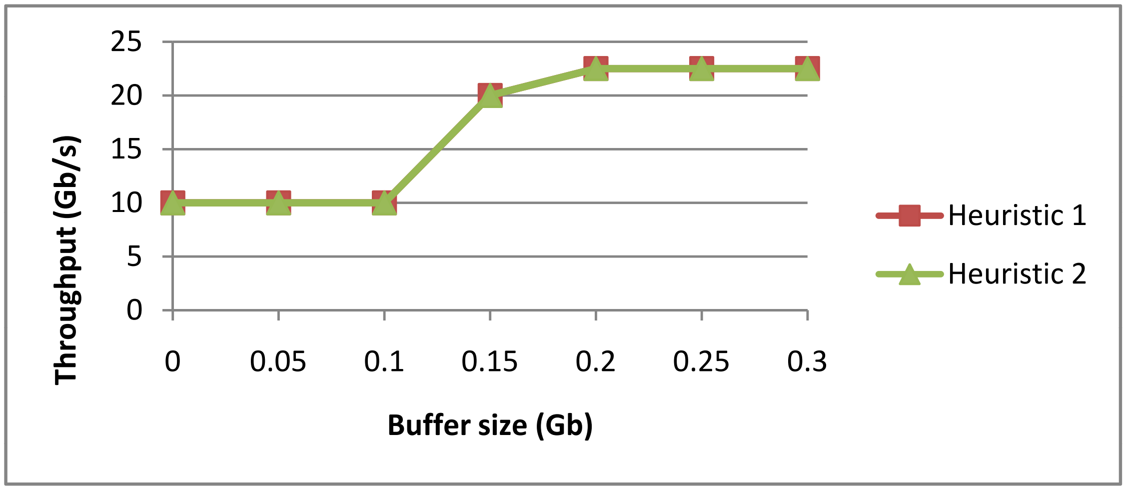


Test topology: 6 nodes. The first data point (buffer 0) corresponds to min-delay single path (current TCP/IP protocols). Optimal Algorithm: max flow with buffer 0 using cycles.



angleangle

Topology used for next page experiments.



Results in less than one second. In one hour, PseudoOptimum algorithm based on LPs achieves half the buffer size of the heuristic.

Conclusions

1. motivated the multipath routing problem with a buffer constraint at the destination
2. theoretical results to obtain optimal buffer size for a desired throughput
3. greedy heuristic and experiments: improves throughput over TCP/IP
4. theoretical results work (only) with fixed number of source-sink pairs

Extend the optimization metrics and algorithmic solutions to fluctuating demand and rapidly varying link delays: active research topic. Distributed Algorithms?