Distributed Quality-of-Service Routing of Best Constrained Shortest Paths.
Abdelhamid MELLOUK, Said HOCEINI, Farid BAGUENINE, Mustapha CHEURFA


Presented By : Bijay Kumar Pathak
11/30/2012
Introduction

The Routing Problem

- Traditional routing protocols (RIP, OSPF, etc.) mainly use hop counts to select paths.
- This does not meet the requirements of many emerging communication applications.
- For example, live multimedia applications must make sure that
  - Packet delays are bounded.
  - Jitters (changes in packet delays) are well controlled.
Introduction

• The basic function of QoS routing is:
  - find a network path which satisfies the given constraints and
  - optimize the resource utilization

• QoS constraint include
  – Bandwidth
  – Delay
  – Data Loss rate
  – Queue length (available data space)
• QoS based routing to construct dynamic state dependent routing policies.
• The proposed algorithm used a reinforcement learning paradigm to optimize two QoS criteria:
  - cumulative cost path based on hop count
  - end-to-end delay
Introduction

- Algorithm contains two stages:
  1. Select N best candidate paths regarding the cost cumulative path from the source and destination nodes
  2. Distribute traffic among the N best path according to end-to-end delay criteria optimized by reinforcement learning
Introduction

• Packet distribution is based on a probabilistic module
• Probabilistic Module takes into account:
  – packet delivery time computed by Q learning process
  – latency in the waited queue
  – automatically compute the probability affected to each path.
Algorithm framework

Network topology changed

Data arrived from router y

Arrived reinforcement signal from router z

Search of N Best paths

1. Calculate the optimal Q-value corresponding to the N best path found
2. Send the packet to the x's best neighbor
3. Return the reinforcement signal to the router y

Update Q-values

Fig. N Best Path Q Routing Algorithm Framework
First Stage : Constructing N Best Path

- Circles corresponds to the events being able to occur
- Rectangles are the actions tracked by the router x.
- Router x reacts to three different events:
  - topology changes
  - the arrived packet of data
  - arrived reinforcement signal
- Label setting algorithm variant of Dijkstra’s algorithm is used to find shortest path
- All links cost is equal to 1
Second Stage : Q-learning algorithm to optimize the end-to-end delay

- Second step is to distribute the traffic on N candidate paths.
- Objective is to minimize the average packet delivery time
- Reinforcement signal is chosen corresponds to the estimated time to transfer a packet to its destination
- The value of the signal is chosen by a variant of Q-Routing algorithm
- Bellman-Ford asynchronous relaxation algorithm is used
- Each router x maintains in a Q-table a collection of values of $Q_x(d,s)$, for every destination d and for every interface s.
- Q value reflects a delay of delivering a packet for destination d via interface s.
Q-Learning

- Router x forwards the packet to the best next router y determined from Q-table.
- After receiving the packet, the router y provides x an estimate of its best Q value to reach the destination.
- The new information is added in the Q-values of the router x.
- The rule for updating the router x Q-values are:

\[
Q_x(d,s) = Q_x(d,s) + \eta \left( \max_k Q_y(d,k) + \zeta \right) - Q_x(d,s)
\]

Where \(\eta\) is called learning rate and \(\zeta\) represents the time spent by the packet in x’s queue and transmission time from x to s.
Reinforcement signal

• Reinforcement signal $T$ is defined as the minimum of the sum of the estimated $Q(x,s,d)$ time, and the waiting time in queue $q_s$ corresponding to router $s$.
• The value of $T$ is calculated by

$$T = \min_{s \in \text{neighbour of } x} \{q_s + Q(x,s,d)\}$$

where $Q(x,s,d)$, denote the estimated time by the router $x$ so that the packet reaches its destination $d$ through the router $s$. 
Adaptive Probabilistic path Selection in Multipath Routing

• Static Probability
  – Maximal $P_{\text{max}}$ is associated for the best path and divided the rest of probability $(1-P_{\text{max}})$ for the remaining N-1 paths
  – Uniform distributed random process is implemented in each router to force the router take the alternative routes find in N best path and not only the best one.

• For example, if we have $N=2$ (two paths), $P_1=0.8, P_2=0.2$, if the random number $\leq 0.8$, the router chooses the first Path otherwise the router takes the second one.

• This version of algorithm is named as KSPQR-VST in the paper.
Adaptive Probabilistic path Selection in Multipath Routing

• Dynamic Probability
  – Compute the probability affected to each path automatically
• For the router $x$, the set $\{1,\ldots,N\}$ of $N$ best paths found at time $t$, probability $P_{i}^{k}(t)$ for the $i^{th}$ path in the router $K$ at time $t$:

\[
P_{i}^{k}(t) = \frac{[1/D_{i}(t)]^{\alpha} [1/T_{i}^{k'}(t)]^{\beta}}{\sum_{i=1}^{K} [1/D_{i}(t)]^{\alpha} [1/T_{i}^{k'}(t)]^{\beta}}
\]

• $D_{i}(t)$- packet delivery time for path $I$ at time $t$.
• $T_{i}^{k'}(t)$- latency in queuing file associated to closet router $k'$
• This version of algorithm is named as KSPQR-VDY in the paper
Numerical Results

• Topology

• NSFnet
  – Traffic is sent receive by four end notes composed of 14 router And 21 bidirectional bonds

*Fig 2. Simplex NSFnet topology for simulation.*
Topology

- NTTnet
  - More complex
  - 55 interconnected routers and 162 bidirectional bonds

Fig 3. Complex NTTnet topology for simulation.
Traffic Model

- Request are assumed to arrive independently at each node, following Poisson distribution.
- For simplicity error management, flow and congestion control is not implemented.
- Behavior of algorithm is evaluated in isolation.
Comparative study

- Compare against two well known classical approach:
  - Shortest Path First (SPF)
  - Open Shortest Path First (OSPF)
Simulation with Low Load

Fig 4 (a). NSFnet with a low load.

Fig 4 (b). NNTnet with a low load.
Simulation with heavy load

Fig 5(a). NSFnet with a continuous heavy load

Fig 5(b). NNTnet with a continuous heavy load.
Conclusion

- N-best optimal path is computed with Dijkstra’s algorithm
- Learning algorithm is based on found N-best path in terms of cumulative link cost and optimization of the average delivery times on these links.
- Proves to be superior to classical algorithms
- Route efficiently in large networks even when critical aspects are allowed to vary dynamically.
Questions?