Measurement Based Routing Strategies on Overlay Architectures

Student: **Tuna Güven**

Faculty: **Bobby Bhattacharjee, Richard J. La,** and **Mark A. Shayman**

LTS Review

February 15^{th}, 2005

Outline

- **Measurement-Based Multi-path Unincast Routing**
  - Motivation and Problem Statement
  - Existing Approaches
  - Proposed Multi-path Routing Algorithm
    - Simultaneous Perturbation Stochastic Approximation (SPSA)
  - Simulation Results
- Measurement-Based Multi-path *Multicast* Routing
  - Motivation
  - Existing Approaches
  - Creation of multiple multicast paths
    - Digital Fountain Coding
  - Problem Formulation
  - Network Models
  - Proposed Multi-path Multicast Routing Algorithm
  - Simulation Results

Motivation

- Current Routing Algorithms
  - Single route for a source-destination pair
  - Unbalanced resource utilization
    - Create unnecessary bottlenecks and degrade network performance
    - Some parts of network underutilized
- Application-Layer Overlay Network
  - Overlay nodes - network devices located inside the network
    - Higher processing power and lower bandwidth
    - Used to create alternative paths
      - Source attaches an additional IP header with the address of an overlay node as the destination address
      - Overlay node strips the extra IP header and forwards the packet to the destination
    - Provides multiple routes for each source-destination pair
    - No need to modify the underlying routing protocols!

Problem Statement

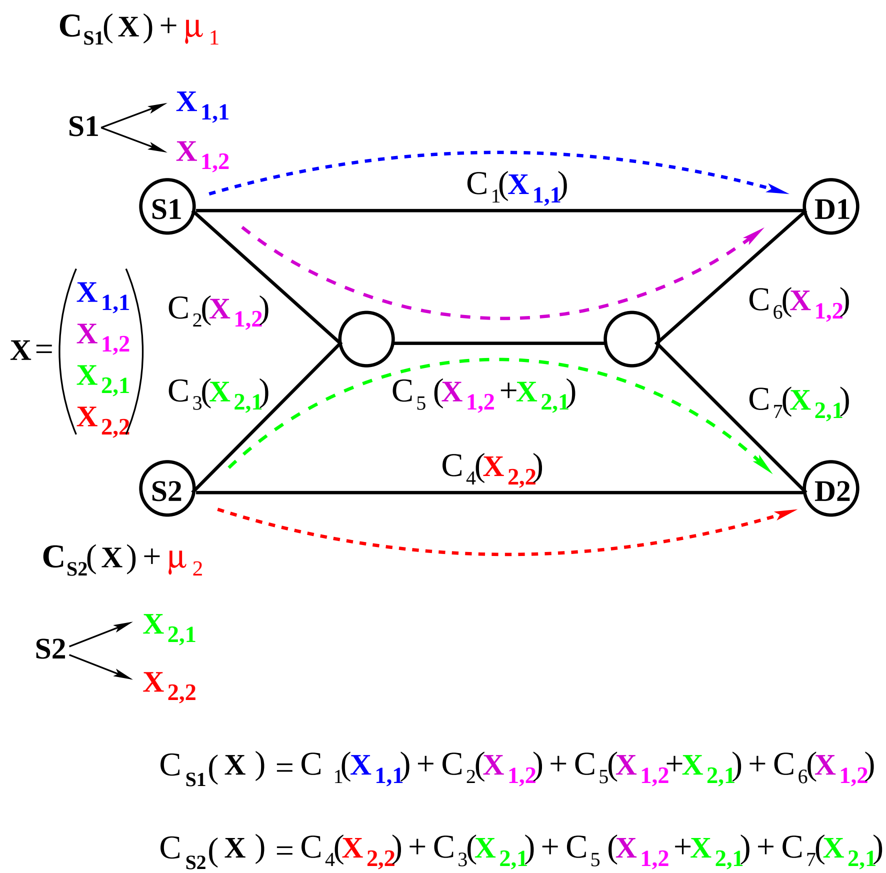
▶ Optimal Multi-path Routing:

$$\min_x C(x) = \min_x \sum_l C_l(x^l)$$

$$\text{s. t.} \sum_{p \in P_s} x_{sp} = r_s, \forall s \in S,$$

$$x_{sp} \geq \varepsilon, \forall p \in P_s, s \in S,$$

- $S = \{1, 2, \cdots, \mathrm{S}\}$ is the set of SD pairs
- $P_s \subseteq 2^L$ is the set of paths available to pair s
- x_{sp} is the amount of traffic routed on path $p \in P_s$
- $x = \{x_{sp}, p \in P_s, s \in S\}$
- $x^l = \sum_{s \in S} \sum_{l \in p:\, p \in P_s} x_{sp}$
- ε is an arbitrarily small positive constant
- $C_l(\cdot)$ is a convex and differentiable function



▶ **Goal:** Minimize $C(x)$ by distributing the load along alternative paths

- Distributed algorithm
- ***Noisy measurements***

Existing Algorithms

- Gradient projection algorithm:

$$x_s(k+1) = \Pi_\Theta\left[x_s(k) - a\nabla C_s(k)\right],$$

  - $x_s = (x_{sp},\ p \in P_s),\ \ a > 0$ is the step size,
  - $\nabla C_s(k) = (\partial C(x(k))/\partial x_{sp}, p \in P_s)$,

- J. N.Tsitsiklis, D.P. Bertsekas, "Distributed Asynchronous Optimal Routing in Data Networks," IEEE Trans. Automat. Control, 1986

- Key facts ignored in the existing solutions:
  - Cost measurements are noisy
  - Analytical cost function is not available (e.g., Network of G/G/1 queues)

- A. Elwalid, C. Jin, S. Low and I. Widjaja, "MATE: MPLS adaptive traffic engineering," IEEE Infocom, 2001
  - Gradient estimated using cost measurements in proposed algorithm
  - Analysis assumes known gradient

Approach - Stochastic Approximation (SA)

- A recursive procedure for finding roots of equation(s) using noisy measurements
- Replace $\nabla C_s(k)$ with its approximation $\hat{g}_s(k)$:

$$x_s(k+1) = \Pi_\Theta[x_s(k) - a_s(k)\hat{g}_s(k)].$$

- Alternative SA methods based on different gradient estimation approaches:
  - ***Finite Differences Stochastic Approximation (FDSA)***
  - ***Simultaneous Perturbation Stochastic Approximation (SPSA)***
- ***FDSA***: Each element of a p dimentional input vector is perturbed ***one at a time*** and corresponding measurements are obtained

$$\hat{g}_i(k) = \frac{y(x(k)+c(k)e_i) - y(x(k)-c(k)e_i)}{2c(k)},$$

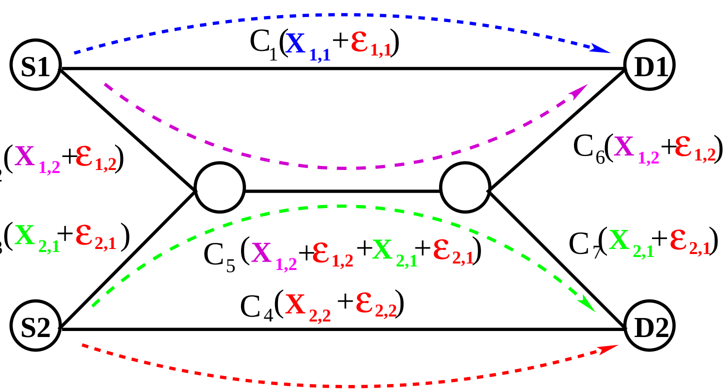
  - $y(\cdot)$ is the observed noisy cost measurement
  - $0 < c(k) < \infty$, $c(k) \to 0$ as $k \to \infty$
  - e_i denotes a unit vector with one in the i-th position and zeros elsewhere
- Requires ***2p*** measurements to get an estimate of the gradient
- ***Remark****:* Implementation presented in MATE relies on the FDSA idea

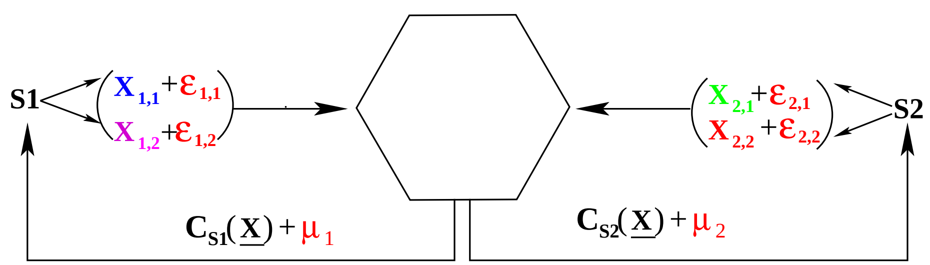
Simultaneous Perturbation Stochastic Approximation (SPSA)

- Elements of the input vector are ***randomly perturbed altogether*** to obtain ***two measurements***

$$\hat{g}_i(k) = \frac{y(x(k)+c(k)\Delta(k)) - y(x(k)-c(k)\Delta(k))}{2c(k)\Delta_i(k)}$$

  - $\Delta(k)$ is the vector of the random perturbations
    - Elements mutually independent with zero mean and uniformly bounded
    - Projected to a feasible space in our problem
  - Gradient estimate calculated using these two estimates





$$C_{S1}(\underline{X}) = C_1(X_{1,1}+\varepsilon_{1,1}) + C_2(X_{1,2}+\varepsilon_{1,2}) + C_5(X_{1,2}+\varepsilon_{1,2}+X_{2,1}+\varepsilon_{2,1}) + C_6(X_{1,2}+\varepsilon_{1,2})$$

$$C_{S2}(\underline{X}) = C_4(X_{2,2}+\varepsilon_{2,2}) + C_3(X_{2,1}+\varepsilon_{2,1}) + C_5(X_{1,2}+\varepsilon_{1,2}+X_{2,1}+\varepsilon_{2,1}) + C_7(X_{2,1}+\varepsilon_{2,1})$$

SA Overview: SPSA vs. FDSA

- Benefits of SPSA over FDSA:
  - It is shown that under reasonably general conditions, **SPSA and FDSA achieve same level of statistical accuracy for a given number of iterations although SPSA uses p times fewer measurements than FDSA**
  - J. Spall, "Multivariate stochastic approx. using simultaneous perturbation gradient approximation," IEEE Trans. Automat. Contr., 1992
- Promising potential for routing problem:
  - Fact: Measurements are costly and time-consuming
  - SPSA gives faster response to time-varying network conditions
  - With certain modifications, SPSA algorithm fits well to our routing problem

SPSA - Based Multi-path Routing

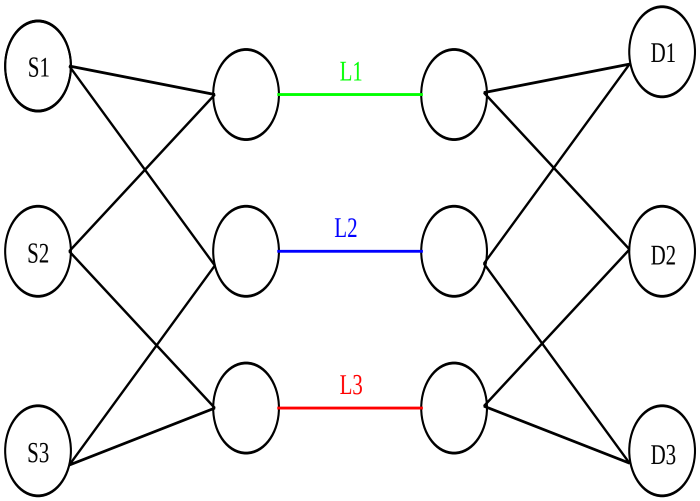
- Proposed Multi-path Routing Algorithm:
  - Each SD pair runs a copy of SPSA algorithm ***independently*** of each other

$$x_s(k+1) = \Pi_\Theta[x_s(k) - a_s(k)\hat{g}_s(k)]$$

$$\hat{g}_{s,i}(k) = \frac{|P_s|}{|P_s|-1} \frac{y_s(\Pi_\Theta[x(k)+c(k)\Delta(k)]) - y_s(x(k))}{c_s(k)\Delta_{s,i}(k)}$$

- Rate vector $x(k)$ converges to the **global optimum**.
- Advantages of the proposed algorithm:
  - Distributed and depends only on local state information
  - No analytical cost gradient function required
  - Measurements can be noisy
  - Significantly reduces measurement time and achieves faster convergence

Simulation Setup



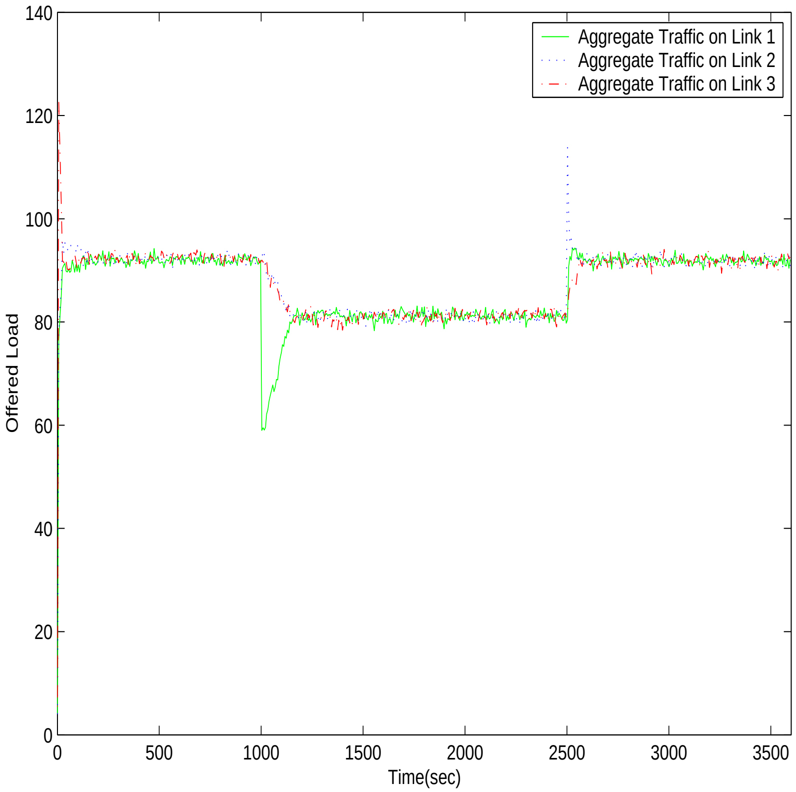
Network Topology

TABLE I
THE CROSS TRAFFIC DYNAMICS

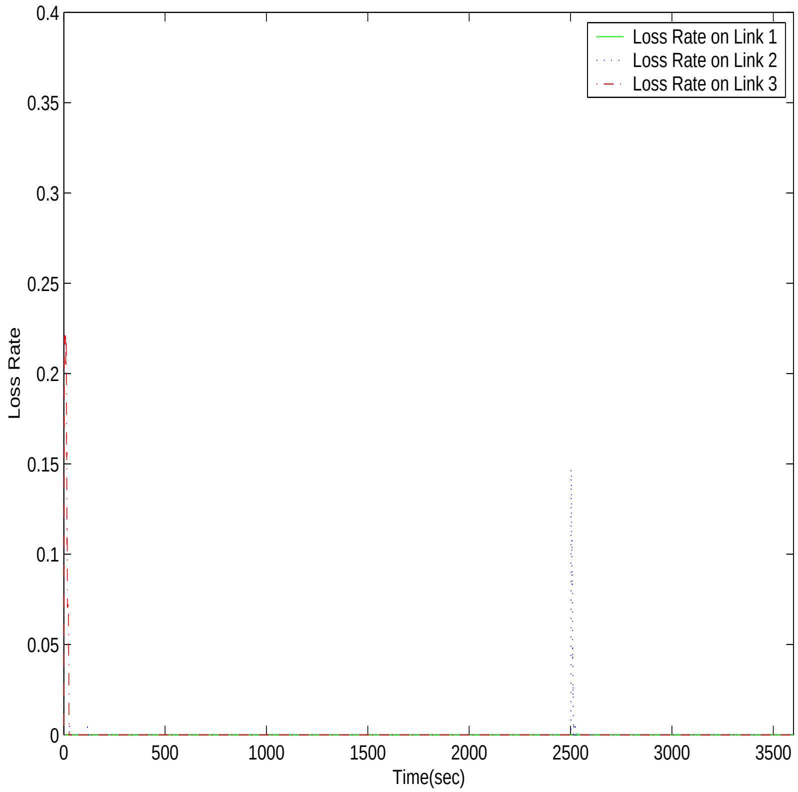
Link	Load Distribution in time (sec)		
	[0 – 1000)	[1000 – 2500)	[2500 – 3600)
$L1$	0.77	0.44	0.44
$L2$	0.33	0.33	0.67
$L3$	0.33	0.33	0.33

- Three SD pairs, each with two alternative paths
- Links capacity - 45 Mbps
- Source rates: 19.8 Mbps (= 0.44 of link capacities)
- Initial routes:
  - (S1→***L2***→D1), (S2→***L3***→D2), (S3→***L3***→D3).
- Lack of synchronization: offset

Simulation Results - (1)



Offered Load (%) (Offset = 50 msec)

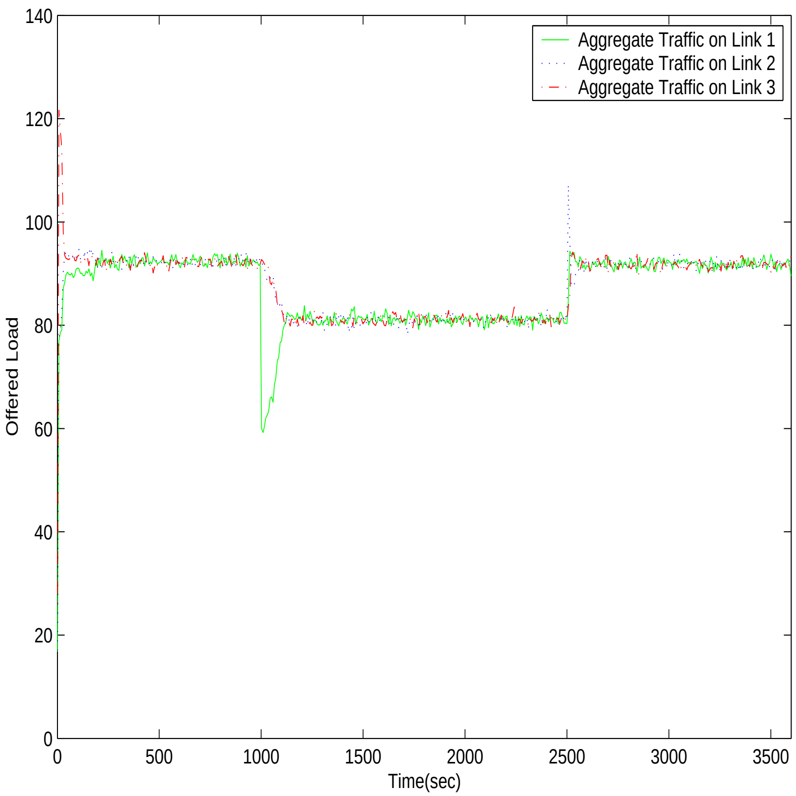


Packet Loss Rate (%)

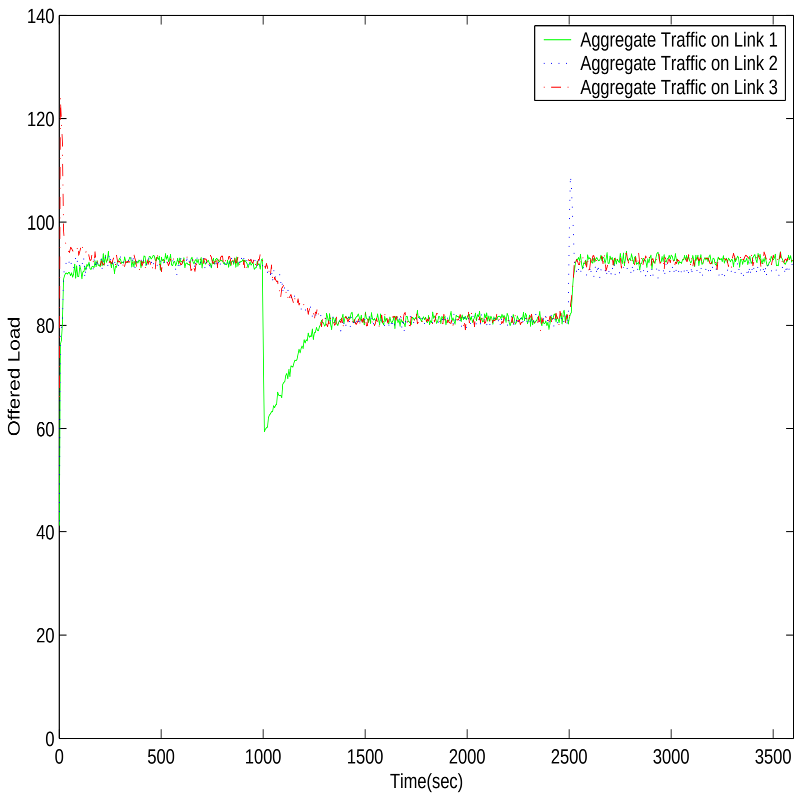
- ***Convergence Time:*** Approximately 500 secs for MATE and 200 secs for the proposed algorithm

Simulation Results - (1) Cont'd

► *Effect of Increasing Interference*



Offered Load (%)
(Offset = 200 msec)



Offered Load (%)
(Offset = 500 msec)

Outline

- Measurement-Based Optimal Multi-path Routing.
- **Measurement-Based Multi-path Multicast Routing:**
  - Motivation
  - Existing Approaches
  - Creation of multiple multicast paths
    - Digital Fountain Coding
  - Problem Formulation
  - Network Models
  - Proposed Multi-path Multicast Routing Algorithm
  - Simulation Results

Motivation

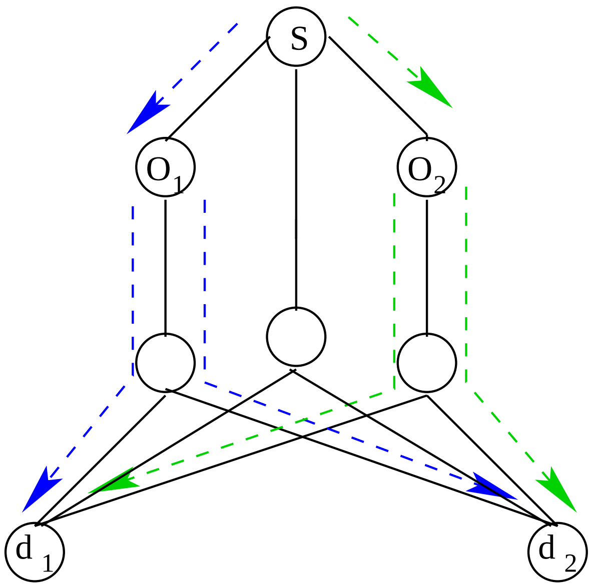
- Intra-domain multi-path multicast routing:
  - Demanding multicast applications with increasing bandwidth requirements
  - Load balancing over multiple paths for efficient network utilization
  - Highly connected ISP backbone topologies
    - N. Spring, et.al., “Measuring ISP topologies with Rocketfuel,” Sigcomm 2002
    - Availability of multiple paths
  - Extending ideas from multi-path unicast routing
  - **Goal:** load distribution using an application-layer overlay network
- Solution applicable for different network models

Existing Approaches

- Multi-tree Routing:
  - K. Park and Y. Shin, "Uncapacitated point-to-multipoint network flow problem," European Journal of Research, 2003
  - Limited to ***single*** multicast source case
  - Noise free measurements; analytic cost gradients are available
  - Cost function is ***strictly convex***, continuous and ***differentiable***
- Network Coding:
  - Y. Zhu, B. Li, J. Guo, "Multicast with Network Coding in Application-Layer overlay networks," IEEE JSAC vol 22, 2004
    - Limited to ***single*** multicast source case
    - Centralized approach
      - Linear codes are assigned to each link by the source node
      - Frequent updates are necessary every time a flow arrives/departs
  - A single packet loss is costlier than usual
    - Receiver requires the lost packet to decode a large block of data

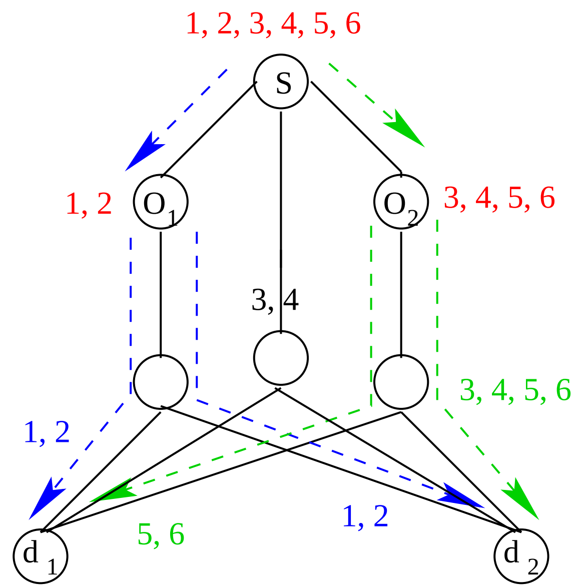
Creating Multiple Multicast Paths

- Application Layer Overlays:
  - Limited number of simple devices located inside the network (e.g., PCs with network processors)
  - Alternative paths are created between a source and a destination
    - Min-hop path from source to overlay and from overlay to destination (IP over IP)
    - Simplifying assumption: Consider only a single overlay node along each path
  - Not necessarily creates multi-trees



Bookkeeping Problem

- Problem with multiple paths in multicast:
  - How to map individual packets to paths for each destination to minimize number of packets sent?
  - Complex bookkeeping problem
- Can solve the problem ...
  - if it is possible to send ***distinct*** packets along each path
- Pre-coding using a erasure correcting code can solve the problem
- However, for efficient implementation the code rate ($R = K/N$) is required to be known before transmission
- Solution: ***Digital Fountain Coding***



$S \to d_1 = 2$ $\quad S \to d_2 = 0$

$S \to O_1 \to d_1 = 2$ $\quad S \to O_1 \to d_2 = 2$

$S \to O_2 \to d_1 = 2$ $\quad S \to O_2 \to d_2 = 4$

Digital Fountain Coding

- A special form of block coding with the following properties:
  - ***Rateless coding:***
    - Number of distinct encoded symbols generated is practically limitless
    - Number of encoded symbols to be generated can be determined on the fly.
  - Output symbols are generated by the XOR addition of ***randomly*** selected input symbols
  - Number of input symbols to be added is ***random*** as well
  - Decoder recovers the K input symbols from any M output symbols with a ***high probability***
    - e.g. ***Raptor Codes***: for $K = 64536$ and $M = 68026$, error probability is $1.71x10^{-14}$
  - Raptor Codes have asymptotically ***linear*** encoding and decoding times
  - Successful commercial implementation with encoding rates at several gigabits/sec by Digital Fountain Company
- Useful for multi-path multicast routing
  - Generate distinct packets - book-keeping unnecessary
  - Routing algorithms merely need to calculate the path rates

Problem Statement

- Optimal Multi-path Multicast Routing:

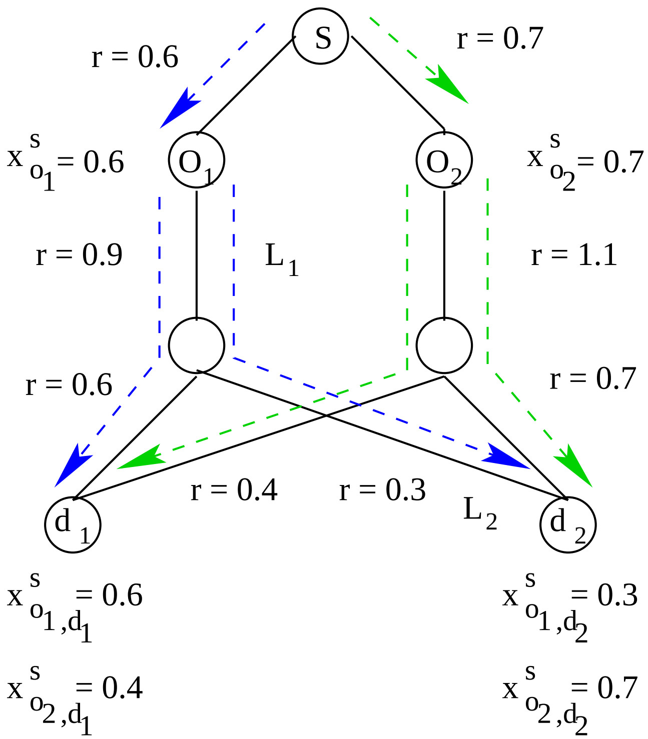
$$\min_x C(x) = \min_x \sum_l C_l(x^l)$$
$$\text{s.t. } \sum_{o \in O^s} x^s_{o,d} = r^s + \varepsilon^s, \forall s \in S, d \in D^s$$
$$x^s_{o,d} \geq \nu,\ \forall d \in D^s, o \in O^s, s \in S$$

- $S = \{1, 2, \cdots, \mathrm{S}\}$ - set of multicast sources
- D^s - set of destination nodes of the session s
- O^s - set of overlay nodes used to create paths between s and its destinations D^s
- $x^s_{o,d}$ - rate at which source s sends packets to destination d through overlay node o
- ε^s - required redundancy due to Digital Fountain Coding
- ν - an arbitrarily small positive constant
- Value of x^l depends on the adopted Network Model

Network Model- I

- Represents traditional IP networks without any multicasting capability

$$x^l = \sum_{s\in S}\left(\sum_{o\in O^s : l\in V_o^s} x_o^s + \sum_{o\in O^s}\left(\sum_{d\in D^s : l\in V_d^o} x_{o,d}^s\right)\right)$$



- $x_o^s = \max_{d\in D^s}\{x_{o,d}^s\}$ is the total rate at which overlay node o receives packets from source s
- $V_{n_2}^{n_1}$ is the set of links in the default path from node n_1 to node n_2

- ***Remark:*** As opposed to the unicast case, $C^l(x^l)$ is not differentiable with respect to input variables $x_{o,d}^s$

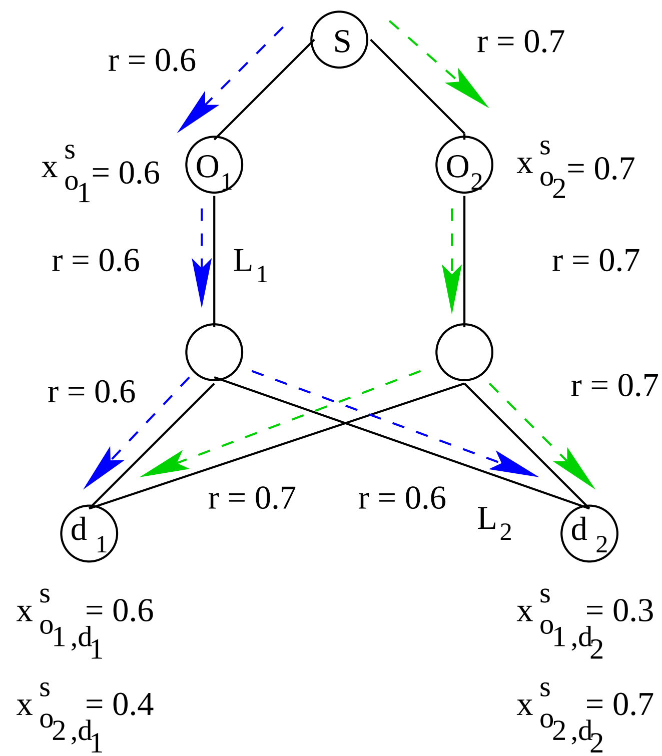
Network Model-II

- Represents a network model with IP Multicast capability (e.g., DVMRP)

$$x^l = \sum_{s \in S} \left(\sum_{o \in O^s : l \in V_o^s} x_o^s + \sum_{o \in O^s : l \in T_o^s} x_o^s \right)$$

  - $x_o^s = \max_{d \in D^s} \{x_{o,d}^s\}$ is the total rate at which overlay node o receives packets from source s
  - $V_{n_2}^{n_1}$ is the set of links in the default path from node n_1 to node n_2, established by the underlying routing protocol (e.g., OSPF)
  - T_o^s is set of links in the multicast tree rooted at overlay node o and serving nodes in D^s
  - Observation:

$$\begin{aligned} x_{o,d}^{s\star} &= x_{o,d'}^{s\star} \quad \forall d, d' \in D^s \\ x_o^{s\star} &= x_{o,d}^{s\star} \quad \forall d \in D^s, o \in O^s, s \in S. \end{aligned}$$

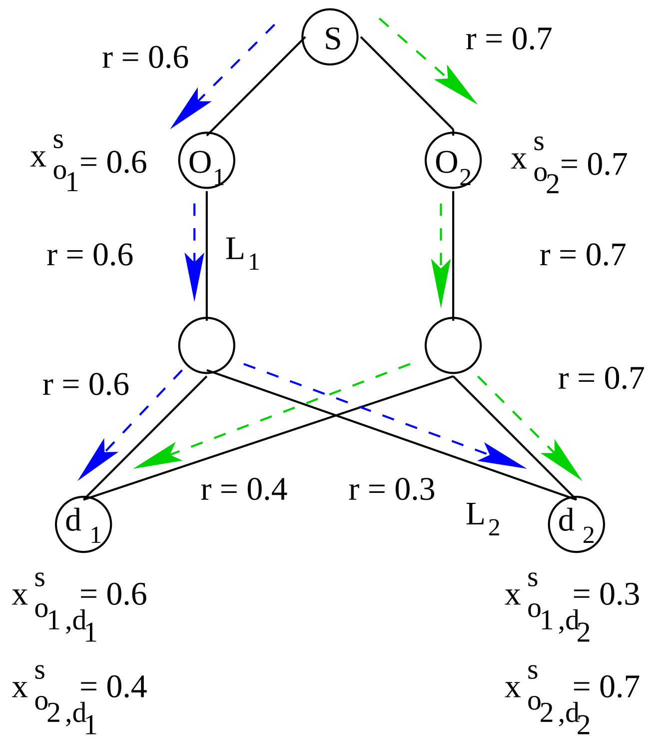


- Hence, the rate allocation problem can be reduced to find $x := (x_o^s, s \in S, o \in O^s)$.

Network Model-III

- Represents a network model with ***smart routers*** in addition to IP multicast
  - Capable of forwarding packets onto each branch at a different rate

$$x^l = \sum_{s\in S}\left(\sum_{o\in O^s:l\in V_o^s} x_o^s + \sum_{o\in O^s}\max_{d\in D^s:l\in \hat{V}_d^o} x_{o,d}^s\right)$$



  - $V_{n_2}^{n_1} \subset L$ is the set of links in the default path from node n_1 to node n_2
  - $\hat{V}_d^o$ denotes the set of links along the path from overlay node o to destination d in the multicast tree
    - May be different from the path provided by the underlying routing protocol

SPSA - Based Multi-path Multicast Routing

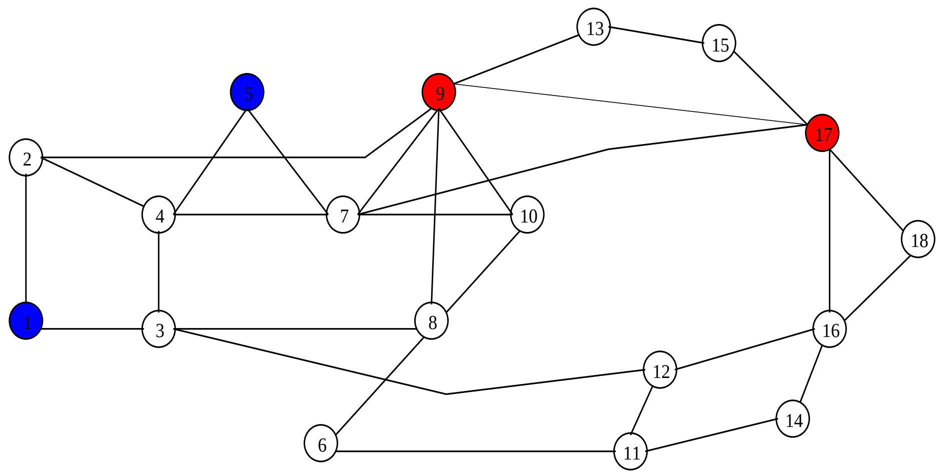
- Each multicast source runs SPSA ***independently*** to minimize the cost along its paths.

$$x_s(k+1) = \Pi_{\Theta_s}[x_s(k) - a_s(k)\hat{g}_s(k)]$$
$$\hat{g}_{s,i}(k) = \frac{|O^s|}{|O^s|-1} \frac{y_s(\Pi_\Theta[x(k)+c(k)\Delta(k)]) - y_s(x(k))}{c_s(k)\Delta_{s,i}(k)}$$

- Main differences from the unicast case:
  - Cost function no longer differentiable
    - Convex Analysis (i.e., subgradients) instead of Taylor Series expansion
- The overall system converges to the ***global optimum***
- Merits of the optimal routing algorithm:
  - Distributed, and depends only on local state information
  - Does not rely on analytical cost gradient function
  - Measurements can be noisy
- Same algorithm can be run under all network models
  - Benefits of additional multicasting functionality can be analyzed

Simulation Results - (1)

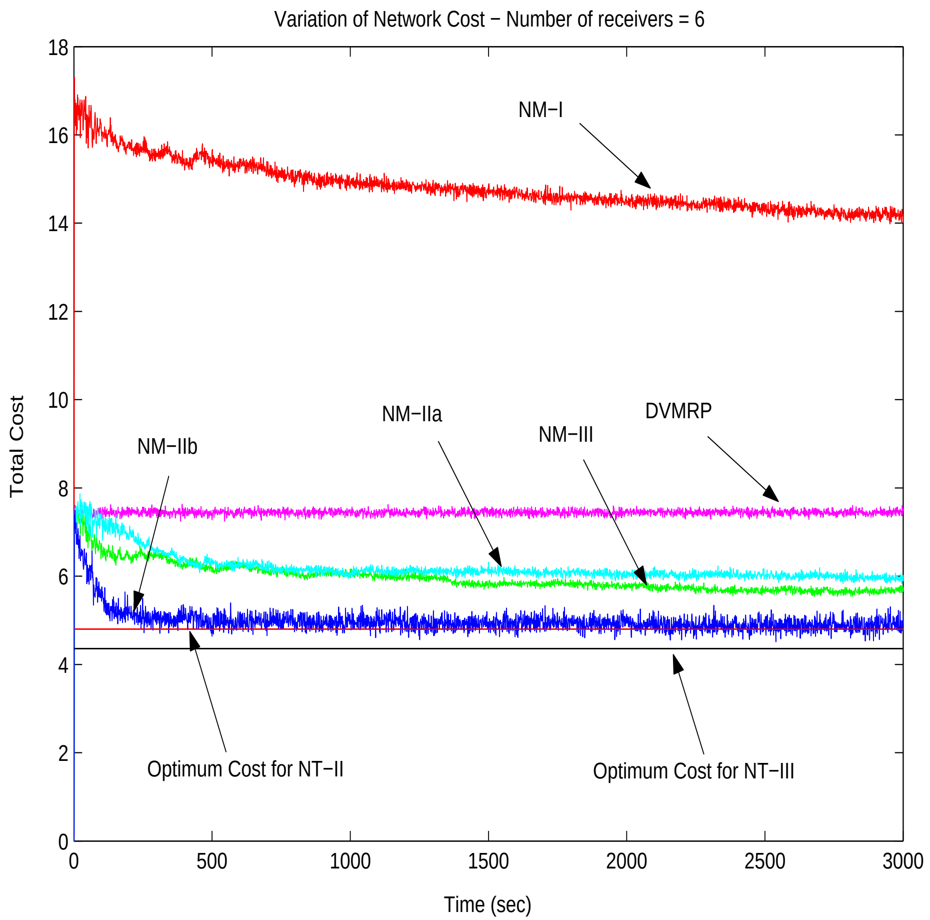
- ISP topology analysis - 1
  - MCI backbone topology



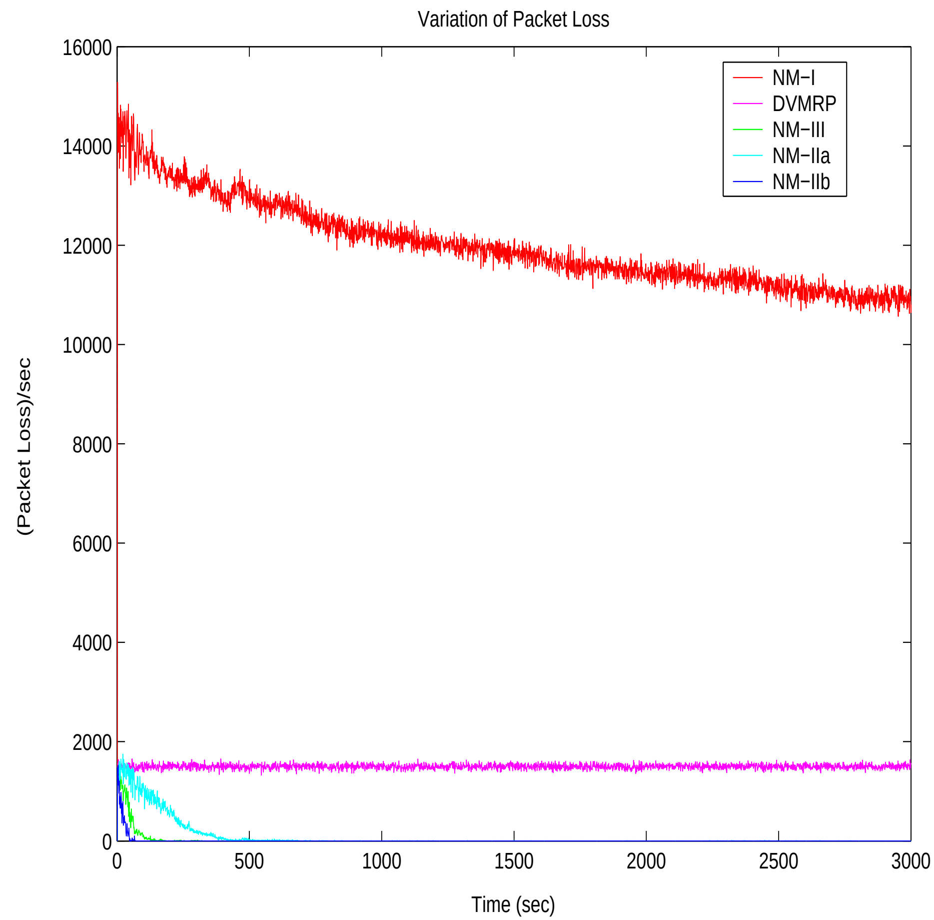
  - Link bandwidth: 20 Mbps
  - Nodes 1 and 5 are multicast sources
  - Each source creates 11.5 Mbps Poisson traffic
  - Nodes 9 and 17 are overlay nodes
  - Link cost : $(x^l/c^l)^2$, where x^l is the link rate and c^l is the link capacity
- Performance of the proposed algorithm under different network models
- Comparison with DVMRP

Simulation Results - (1) Cont'd

- Number of receivers = 6



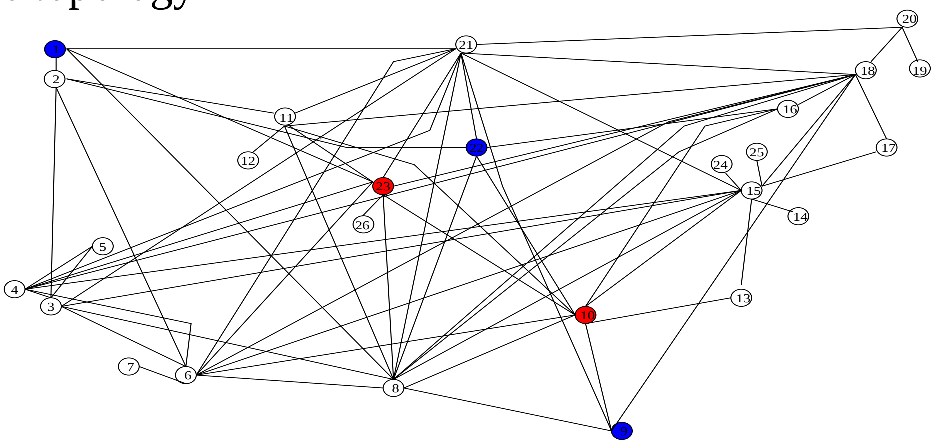
Network Cost



Packet Loss

Simulation Results - (2)

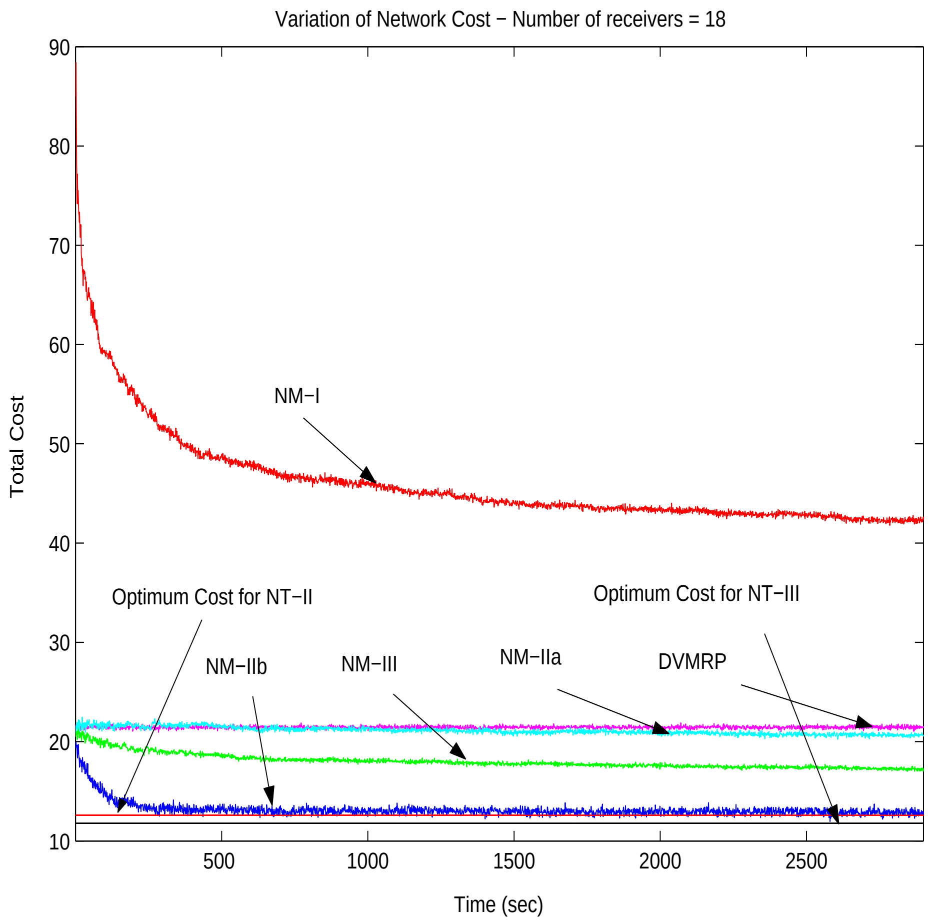
- ISP topology analysis - 2
  - Sprint backbone topology



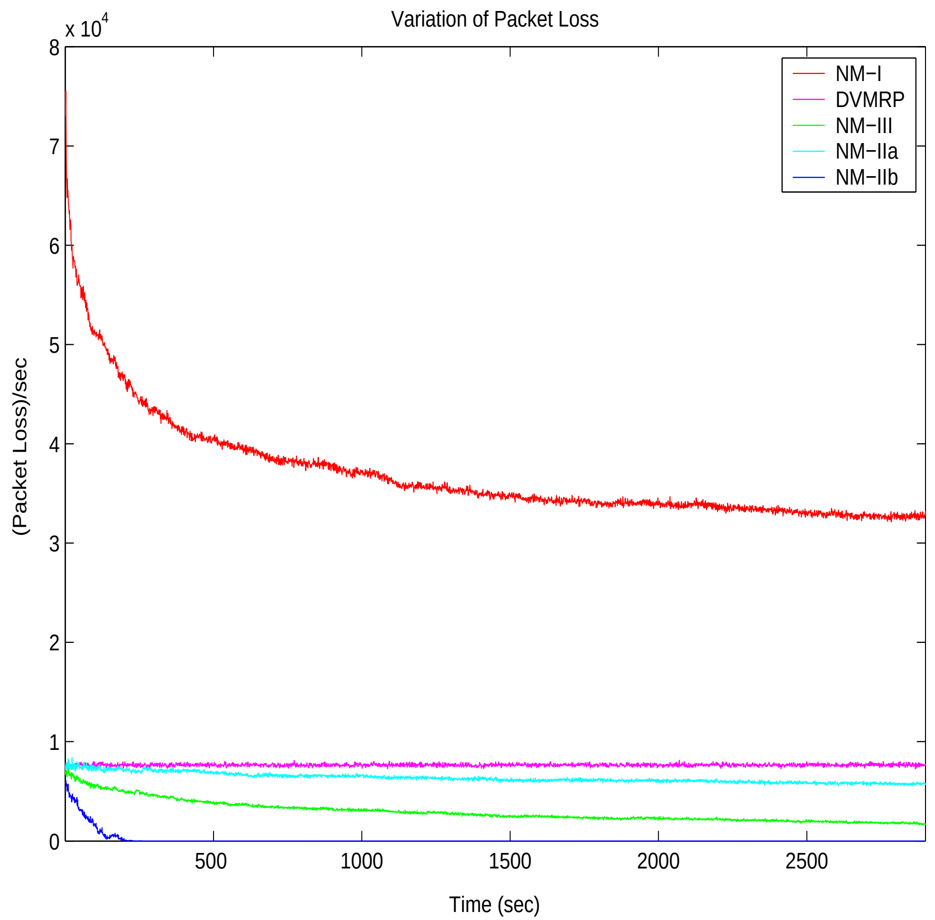
  - Higher node connectivity compared to MCI topology (3.167 vs 5.077)
  - Link bandwidth: 20 Mbps
  - Nodes 1, 9 and 22 are multicast sources
  - Each source creates 10 Mbps Poisson traffic
  - Nodes 10 and 23 are overlay nodes
  - Each source has 18 receivers
- Performance of the proposed algorithm under different network models
- Comparison with DVMRP

Simulation Results - (2) Cont'd

- Number of receivers = 18



Network Cost

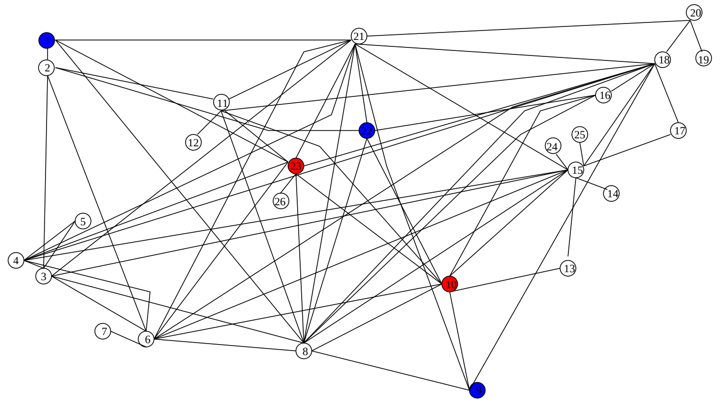


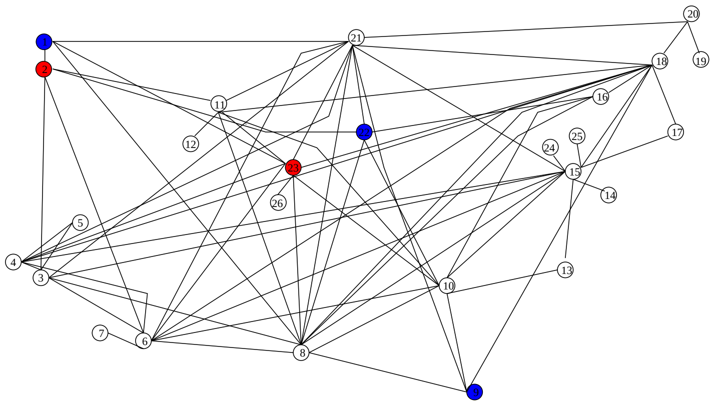
Packet Loss

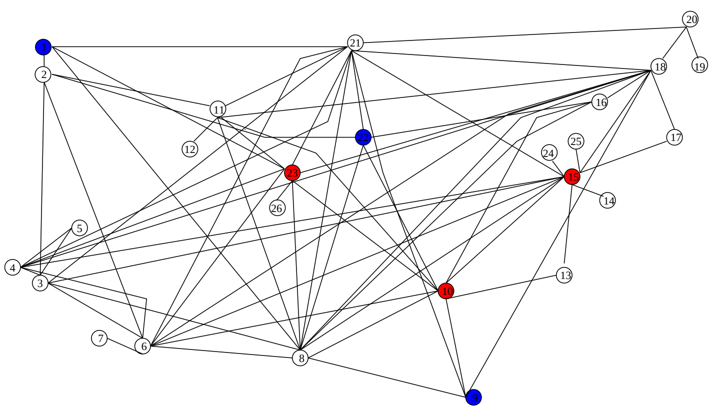
Future Work: Overlay Topology Control

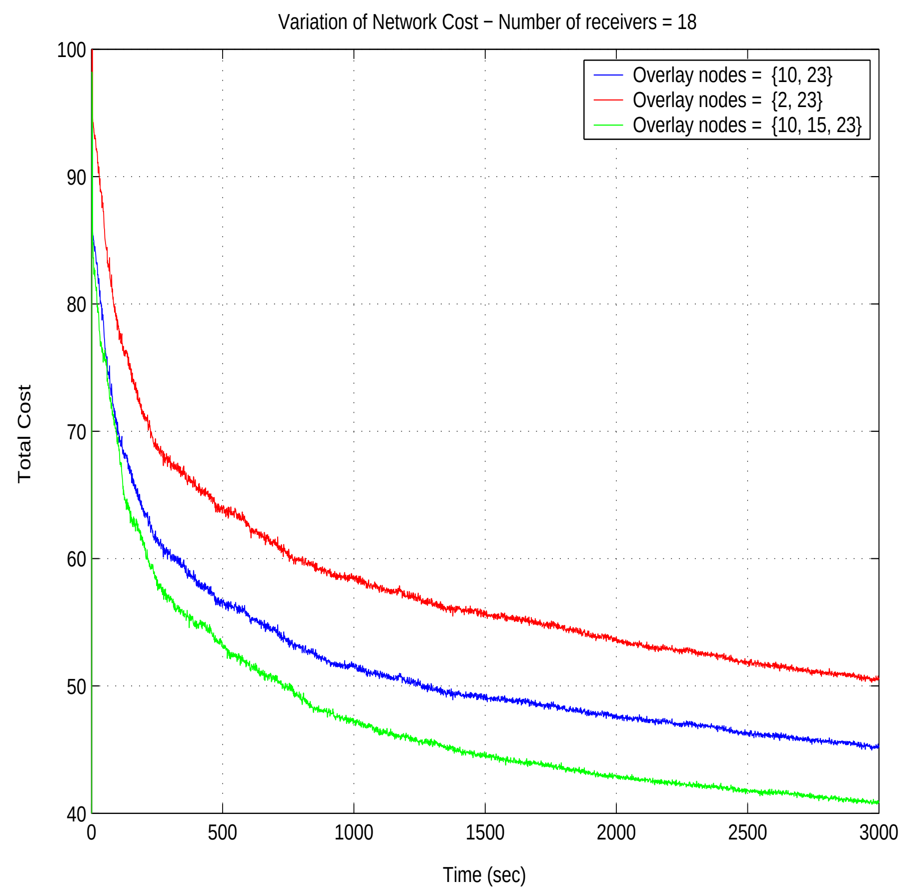
- We have assumed the paths between source destination pairs are given
  - Number, location, and connectivity of overlay nodes was assumed to be given and fixed
- Significant effects on the overall performance of the routing algorithms
- Each overlay node comes with additional cost:
  - Want to maximize network performance with minimum number of overlay nodes
- Simple simulation study reflecting the effect of overlay selection on performance:
  - Experiment done under Network Model-I under Sprint backbone topology

Overlay Topology Control









Overlay Topology Control

- Connectivity of overlay nodes may have significant effects as well
  - Relax the assumption of having only one overlay node along each path
- ***Goal:***
  - Establish an overlay topology control architecture in conjunction with the existing multipath routing algorithms
  - Optimization methods such as Simulated Annealing or Genetic Algorithms may be used for this combinatorial problem
  - Alternative: Optimal paths can be discovered first by ignoring the overlay architecture and then they can be approximated by limited number of overlays