

One-to-Many TCP

Scalability and End-to-End Reliability

Zhen LIU

IBM T. J. Watson Research Center

Summary

- Non Scalability of Reliable Native IP Multicast
- Multicast Overlays
- Reliability of Multicast Overlays by Back-Pressure
- Scalability of Overlay-Multicast
- The Infinite Memory Case
- joint work with F. Baccelli (INRIA-ENS), A. Chaintreau (INRIA-ENS), S. Sahu (IBM Research) and A. Riabov (Columbia), INFOCOM 04, MTNS 04.

Reliable Group Communication over the Internet

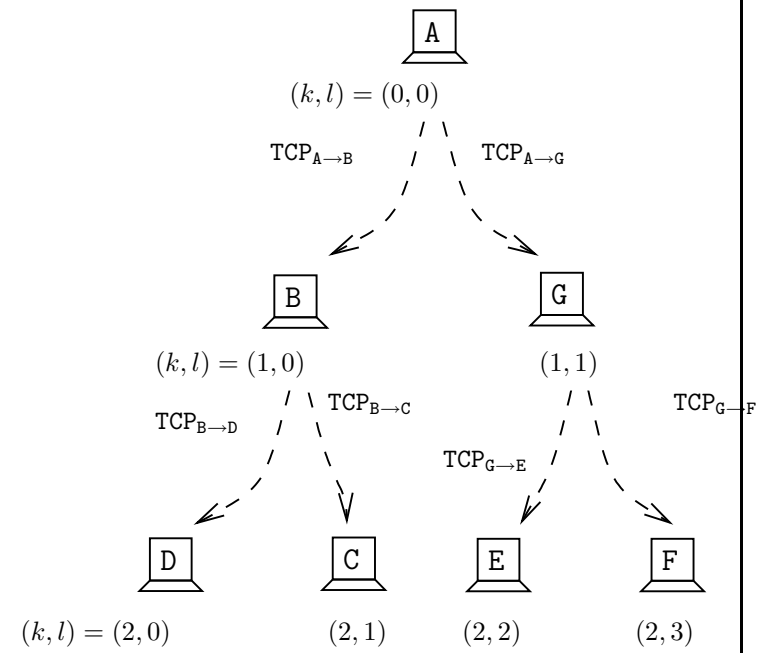
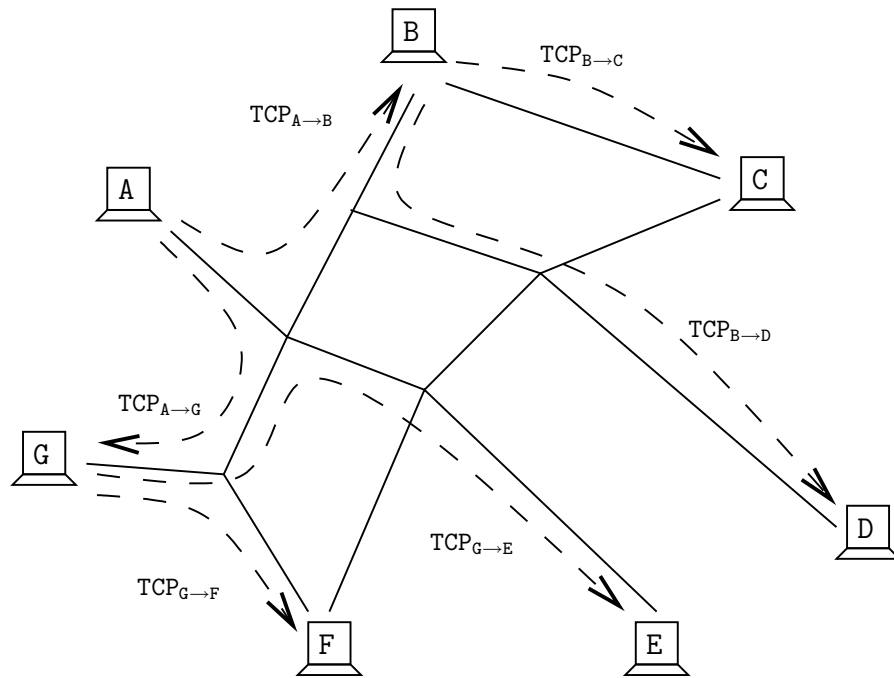
- **Group communication**: the same large data has to be transported in an efficient and reliable way from a source to a large set of users
- **Multicast tree**: the broadcasting is made via a tree where each node duplicates the packets it receives from its mother node and sends them to all its daughter nodes
 - Native IP multicast
 - Overlay based multicast (or application-level multicast)

Native IP Multicast

- **Native reliable IP multicast**: the nodes of the tree are Internet routers;
- IP-supported multicast has **deployment obstacles**: new functions (routing, replication) are needed in routers.
- Reliable IP-supported multicast has scalability problems: in the presence of random fluctuations, when a window congestion control mechanism is used for ensuring reliability, the **group throughput tends to 0 when the group size gets large**
 - S. Bhattacharyya, D. Towsley, J. Kurose INFOCOM 1999
The Loss Path Multiplicity Problem in Multicast Congestion Control
 - A. Chaintreau, F. Baccelli, C. Diot INFOCOM 2001
Impact of Network Delay Variation on Multicast Session Performance with TCP-like Congestion Control

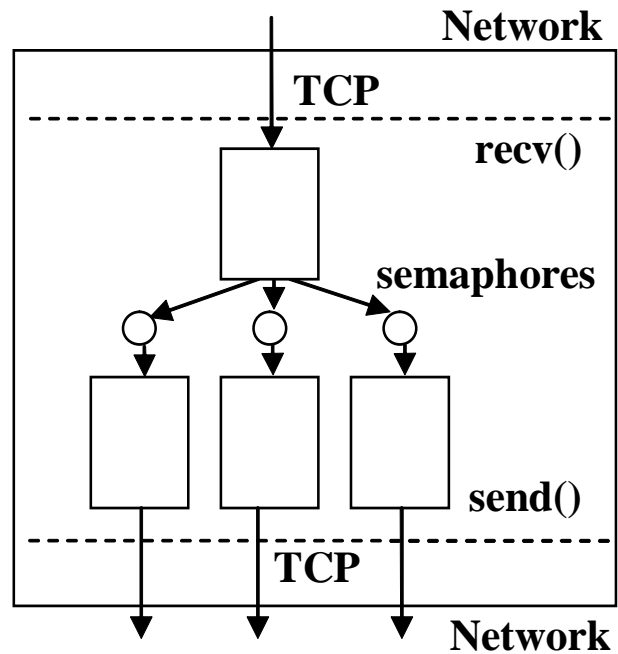
Multicast Overlays

- **IP overlays**: application-layer structures that are built upon the existing transport protocols of the Internet
- **Multicast Overlay**
 - the nodes of the tree are end-systems
 - the edges of the tree are point-to-point TCP connections
 - the end-systems are in charge of multicast routing and replication: after receiving data from its mother node, a node replicates the data on each of its outgoing links and forwards it in sequence to each of its daughter nodes in the overlay tree.



Zoom on End-Systems

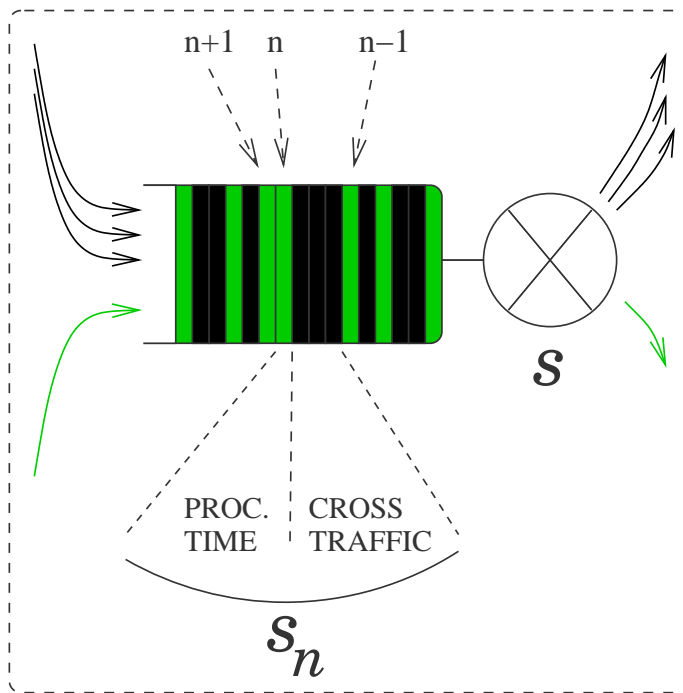
- End-system input and output buffer interaction during overlay multicast.



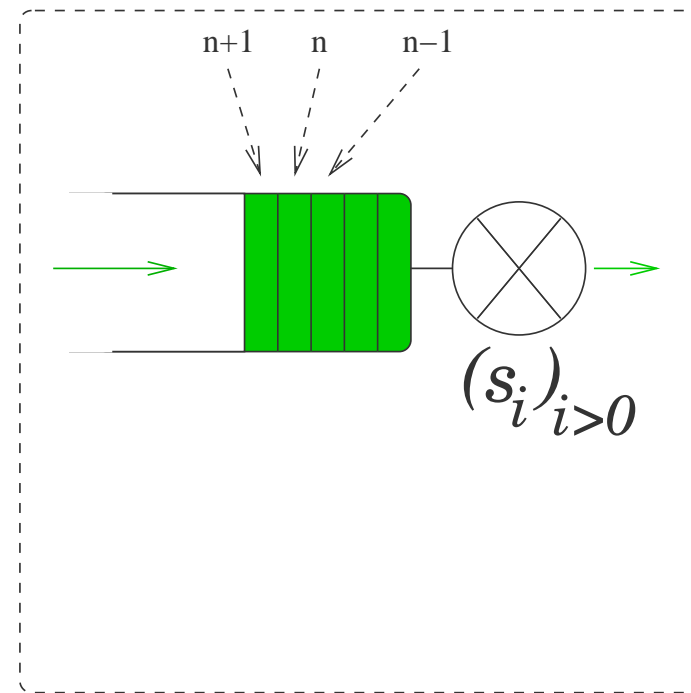
Zoom on a Point to Point TCP Connection

- Each TCP connection
 - has a route that consists of a sequence of **Internet routers in series**
 - is affected by **random fluctuations** (slow downs) and random packet losses/marking (AQM, RED) due to its competition with (Internet) **cross traffic**
 - reacts to slow downs using the TCP **adaptive window congestion control mechanism**
 - recovers from packet losses using the **retransmission** mechanism of TCP.

TCP Connection: Random Fluctuations in Routers



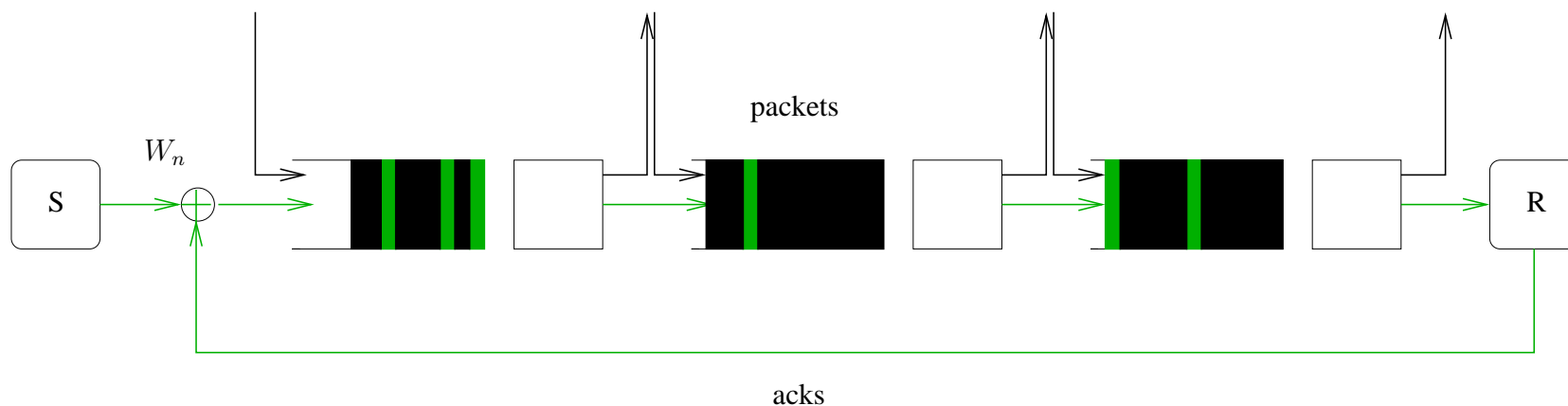
A Router



Our Model

TCP Connection: End to End Adaptive Window Flow Control

- Each received packet is acknowledged by the receiver back to the source via an ack. that contains the sequence number of the packet.
- **Window flow Control:** if the window size is W , the source can only send packet $n + W$ after packet n has been acknowledged.



TCP Connection: End to End Adaptive Window Flow Control (*continued*)

■ TCP window size adaptation

$$w_0 = 1, \quad w_{n+1} = f(w_n, F(n)),$$

$F(n)$: feedback signal on the state of congestion, function of losses (TD) or marking (MK)

■ General principle of TCP Reno's congestion avoidance (CA) phase: AIMD dynamics

$$f(w_n, \text{OK}) = w_n + \frac{1}{w_n}, \quad f(w_n, \text{TD or MK}) = \left\lfloor \frac{w_n}{2} \right\rfloor$$

■ Maximal window advertised by the receiver depending on the size of its input buffer.

TCP Connection with Losses: Retransmissions

- If packet m is lost whereas packets $m + 1, m + 2 \dots$ are received, these packets trigger **duplicate acks**
- When packet $m + 3$ is received, its duplicate ack (TD) triggers the **Fast Retransmit Fast Recovery** procedure
 - the sender sends a **duplicate of packet m** ,
 - **the sender halves its window** and inflates it of 3 units, which usually blocks the sending of new packets
 - each time a new packet $m + 4, m + 5, \dots$ is received, a new duplicate ack is sent back to the source, and each such duplicate ack inflates the window: **constant number of in flight packets**
 - when the duplicate of packet m arrives, all packets from m to $m + W_m$ are simultaneously acknowledged and the window then starts **a new CA phase from $W_m/2$** .

Reliability of Multicast Overlays via Back-Pressure

- Three different types of packet losses in the overlay network:
 1. Losses in the path in-between the nodes:
recovered by the TCP acknowledgment and retransmit mechanisms
 2. Losses due to input buffer overflow:
will not occur thanks to the back-pressure mechanism of TCP
 3. Losses due to output buffer overflow:
will not occur thanks to the overlay back-pressure: a packet will be removed from the input buffer only when it is copied to all of the output buffers. The copy process is blocked when one output buffer is full.
- These two back-pressure mechanisms guarantee that there will be no loss at the overlay nodes even if they have finite-size buffers.

Main Conclusion on Reliability of Multicast Overlays

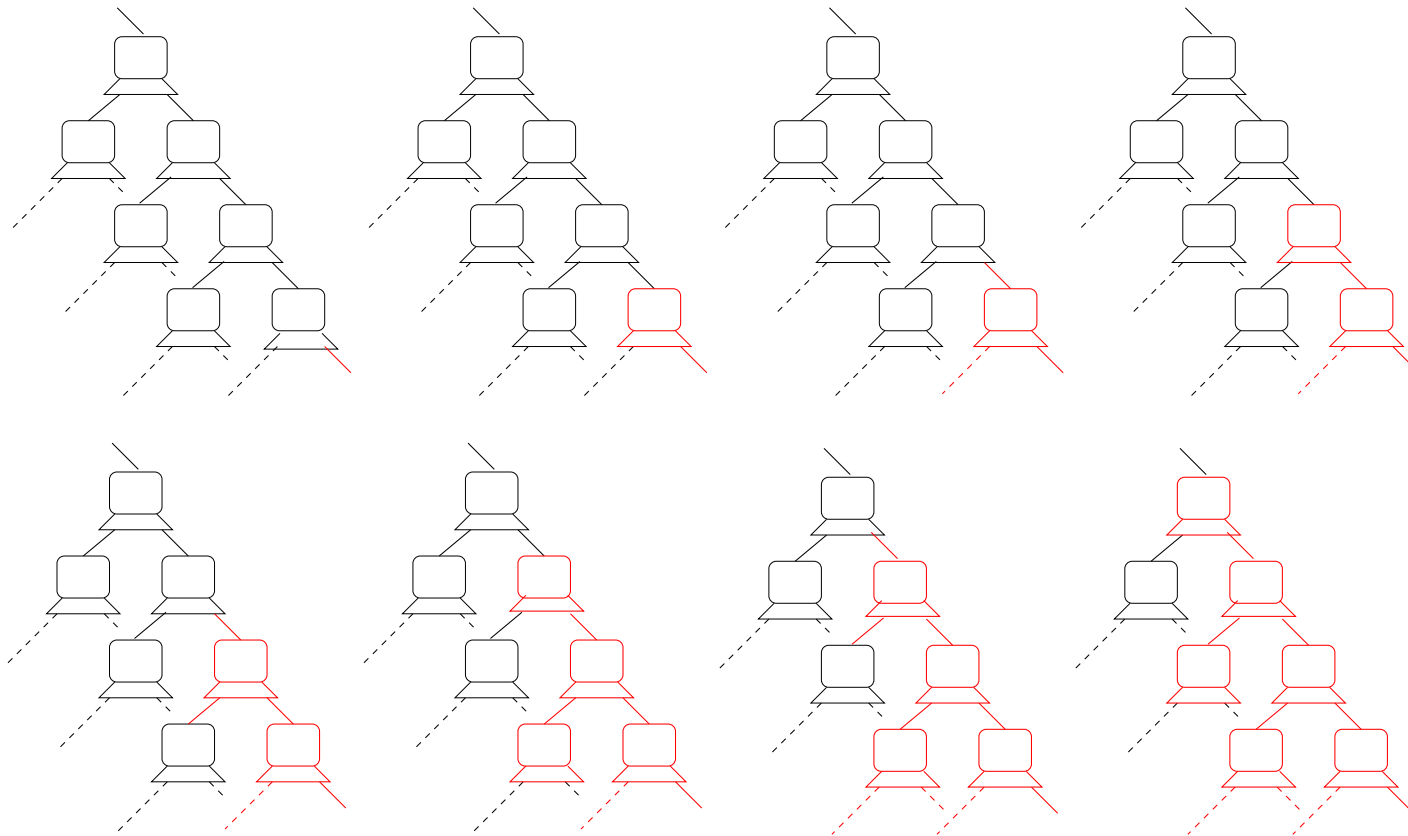
- Backpressured multicast overlays offer a **reliable point to multipoint transport mechanism that adapts to congestion in the network** in the same spirit as TCP does point to point.

Main Question about Scalability of Multicast Overlays

- There are two key new **blocking phenomena** proper to multicast overlays:
 1. In any multicast overlay, the mechanisms ensuring **in-sequence relaying** at each end-system that follows a TCP connection with losses
 2. In any back-pressured multicast overlay, the **overflow prevention** mechanism in each end-system with finite memory.
- How much will these chained blocking mechanisms **reduce the throughput of the group communication** when the group size grows large?

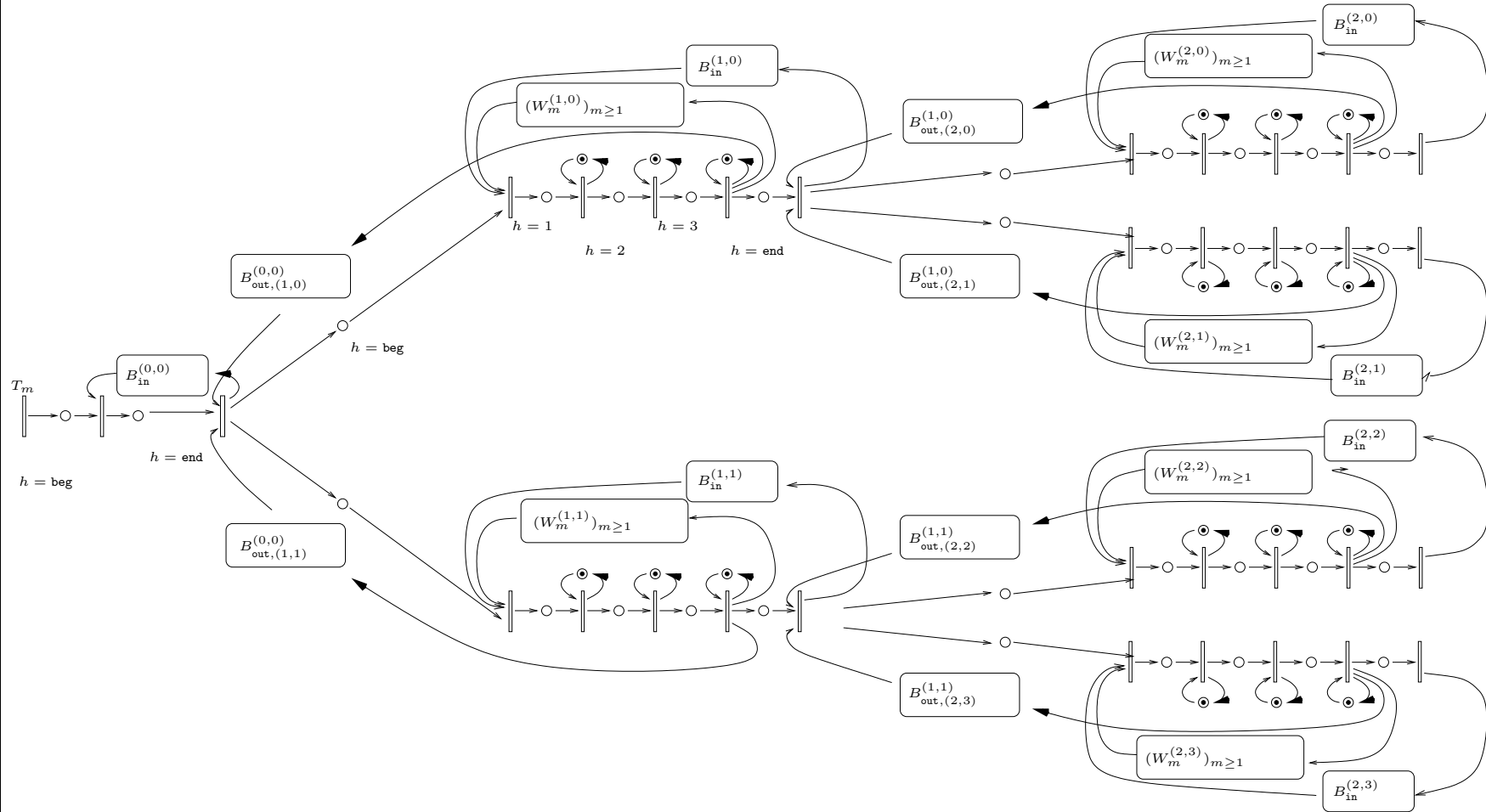
Urvoy-Keller and Biersack NGC, October 2002

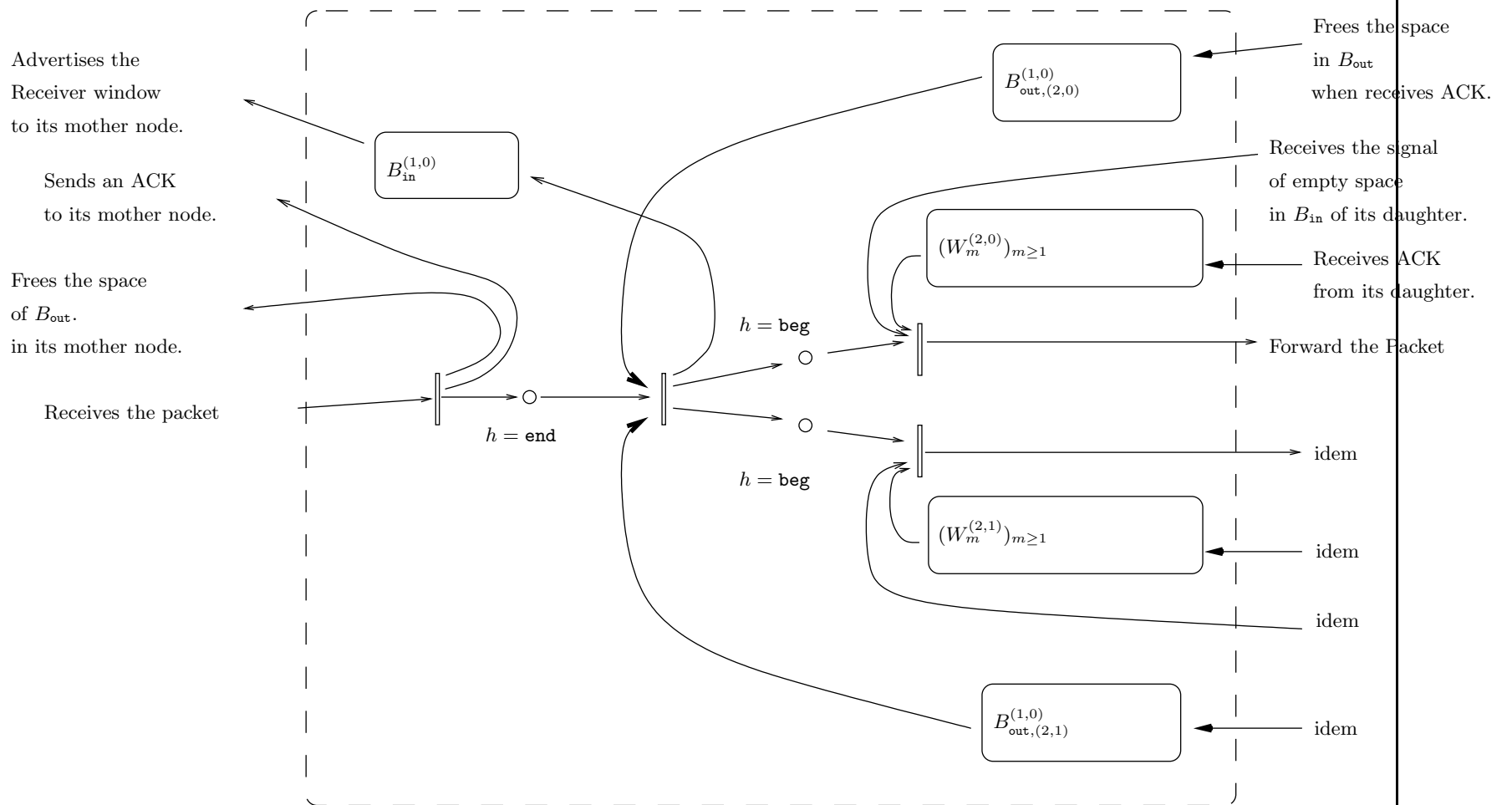
Backward and Forward Propagation of Fluctuations due to Back-Pressure



Petri Net-like Model

- Description of an end-system in the marking case
 - $W_m^{(k,l)}$ the window sequence in TCP connection k, l
 - $H_{(k,l)}$ # routers in TCP connection k, l





Evolution Equations for Fast Simulation

- $x_m^{(k,l,h)}$ time when (k, l, h) completes the transmission of packet m
- Equation for root (with $\mathbf{d}(k, l)$ the daughter nodes of k, l):

$$x_m^{(0,0,\text{beg})} = T_m \vee x_{m-B_{\text{in}}^{(0,0)}}^{(0,0,\text{end})}$$

$$x_m^{(0,0,\text{end})} = x_m^{(0,0,\text{beg})} \vee \left(\bigvee_{l \in \mathbf{d}(0,0)} x_{m-B_{\text{out},(1,l)}^{(0,0)}}^{(1,l,H(1,l))} \right)$$

- $T_m \equiv 0$: saturated input case.

Evolution Equations for Fast Simulation (continued)

- **Internal nodes:** for $k \geq 1$, $l \geq 0$, (with $\mathbf{m}(k, l)$ the mother node of k, l):

$$x_m^{(k,l,\text{beg})} = x_m^{(k-1,\mathbf{m}(k,l),\text{end})} \vee x_{m-B_{\text{in}}^{(k,l)}}^{(k,l,\text{end})} \vee x_{m-W_m^{(k,l)}}^{(k,l,H(k,l))}$$

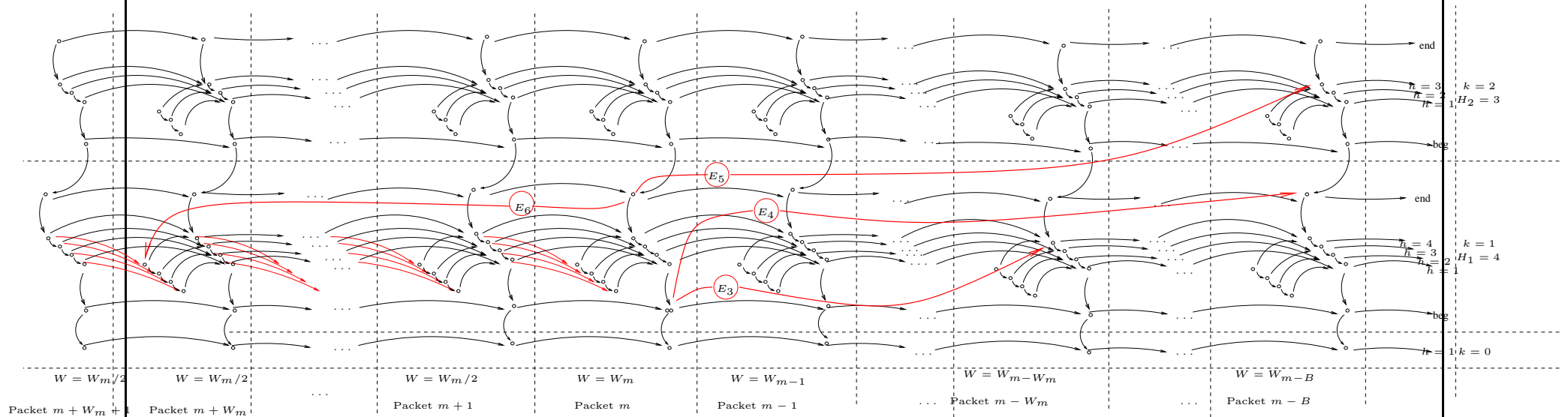
$$x_m^{(k,l,1)} = \left(x_m^{(k,l,\text{beg})} \vee x_{m-1}^{(k,l,1)} \right) + s_m^{(k,l,1)}$$

$$x_m^{(k,l,H(k,l))} = \left(x_m^{(k,l,H(k,l)-1)} \vee x_{m-1}^{(k,l,H(k,l))} \right) + s_m^{(k,l,H(k,l))}$$

$$x_m^{(k,l,\text{end})} = \left(x_m^{(k,l,H(k,l))} \vee \left(\bigvee_{l' \in \mathbf{d}(k,l)} x_{m-B_{\text{out},(k+1,l')}^{(k+1,l',H(k+1,l'))}} \right) \right)$$

Interpretation as Longest Path in a Random Graph

- Case of two TCP connections in series with losses and resequencing



- E_3 : window control; E_4, E_5 : backpressure; E_6 : resequencing.

- In the saturated root case, $x_m^{(k,l,\text{end})}$ is the weight of the **maximum weight path** from (k, l, end, m) to $(-1, 0, 1, 0)$ in this random graph.

Topology – Statistical Assumptions

■ Homogeneous model

- The tree has a fixed degree D (math + first mile effect)
- All TCP connections are structurally and statistically equivalent
- All back-pressure parameters are the same everywhere in the tree.
- The packet marking or loss process is independent and identically distributed (i.i.d.) in each TCP connection: W_m 's are Markov Chains.
- Aggregated service times are independent and identically distributed in all routers, with law σ with finite mean.

Topology – Statistical Assumptions (*continued*)

■ Non-homogeneous model

- The fan out degree in the tree is bounded from above by D
- The number of hops of each route is bounded from above by H
- The packet losses/markings are i.i.d. with loss/marketing probability bounded from above by $p < 1$
- The input and output buffers are bounded from below and from above by constants
- The aggregated service times are independent and upper bounded by a random variable σ with finite mean.

■ Both cases guarantee a positive throughput for each TCP connection when saturated

Main Result

Theorem Consider an overlay multicast tree with infinite height $k = 0, 1, 2, \dots$. If the random variable σ is light tailed, i.e. there exists a real number $\tau > 0$ such that $\mathbb{E}[e^{t\sigma}] \leq A(t) < +\infty$ for all $0 \leq t \leq \tau$, then under both the homogeneous and the non-homogeneous assumptions,

$$\theta^{-1} = \limsup_{m \rightarrow \infty} \frac{x_m^{(k,l,\text{end})}}{m} \leq \text{Const}(H, D) < \infty \text{ a.s. .}$$

uniformly in (k, l) , both for the marking and the loss-resequencing cases. In the light tailed i.i.d. case, the group communication is strictly positive even in an infinite tree.

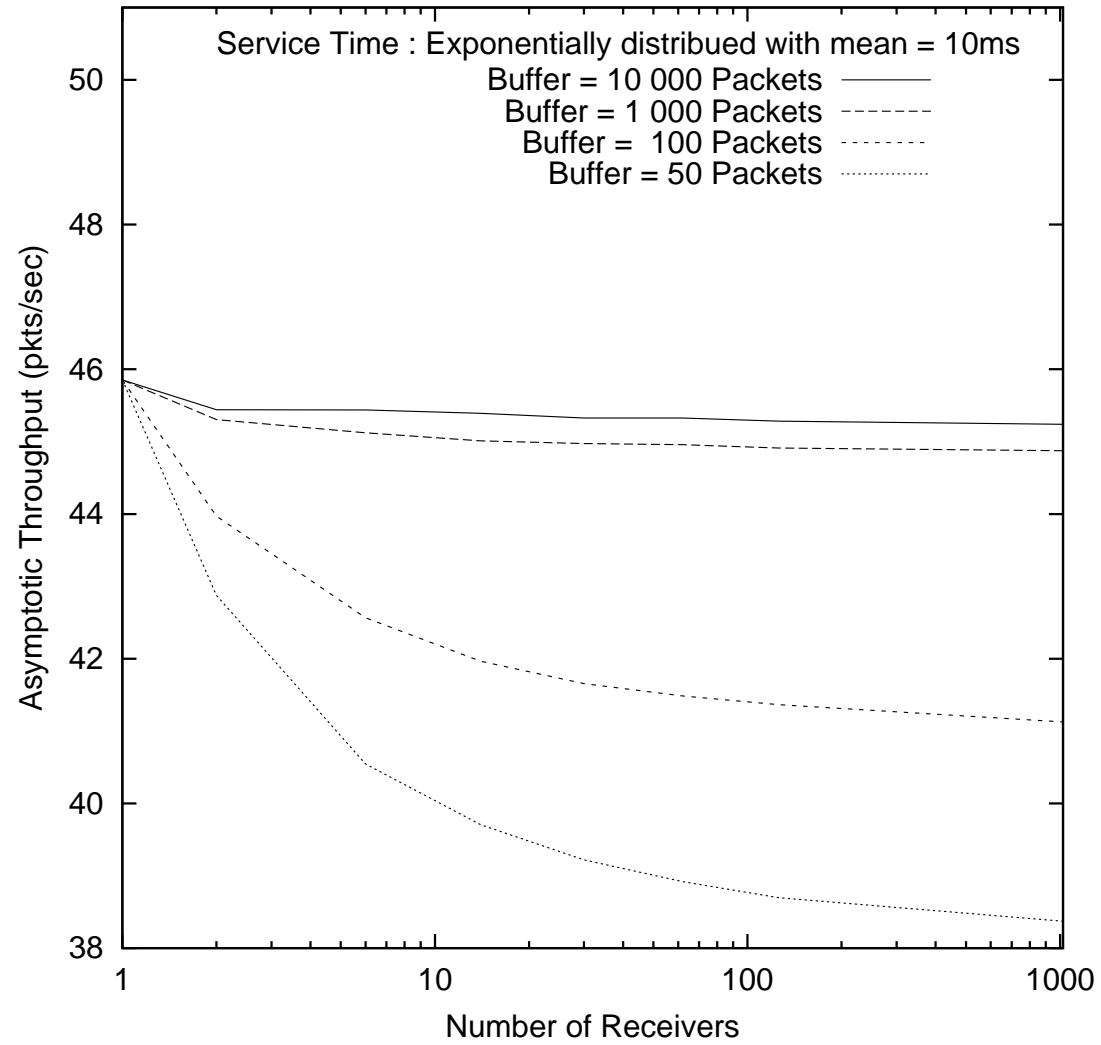
Extensions

- Heavy tailed case with moment condition in the case of an infinite series of TCP connections
- The tree case with heavy tails is an open question

Simulations: Buffer Size

- Based on the (max plus) equations
- Can handle up to 1000 nodes
- Focus on asymptotic group throughput
- Default option:
 - $p = 0.01$, 10 routers in series, exponential and Pareto random aggregated service times with mean equal to 10ms
 - $B=50,100,1000$ and 10 000 Pkts
 - transfer of 2GB of data

TCP Reno, Overlay : 10 Routers, $W_{max} = 40$, Pack Loss Proba = 0.01



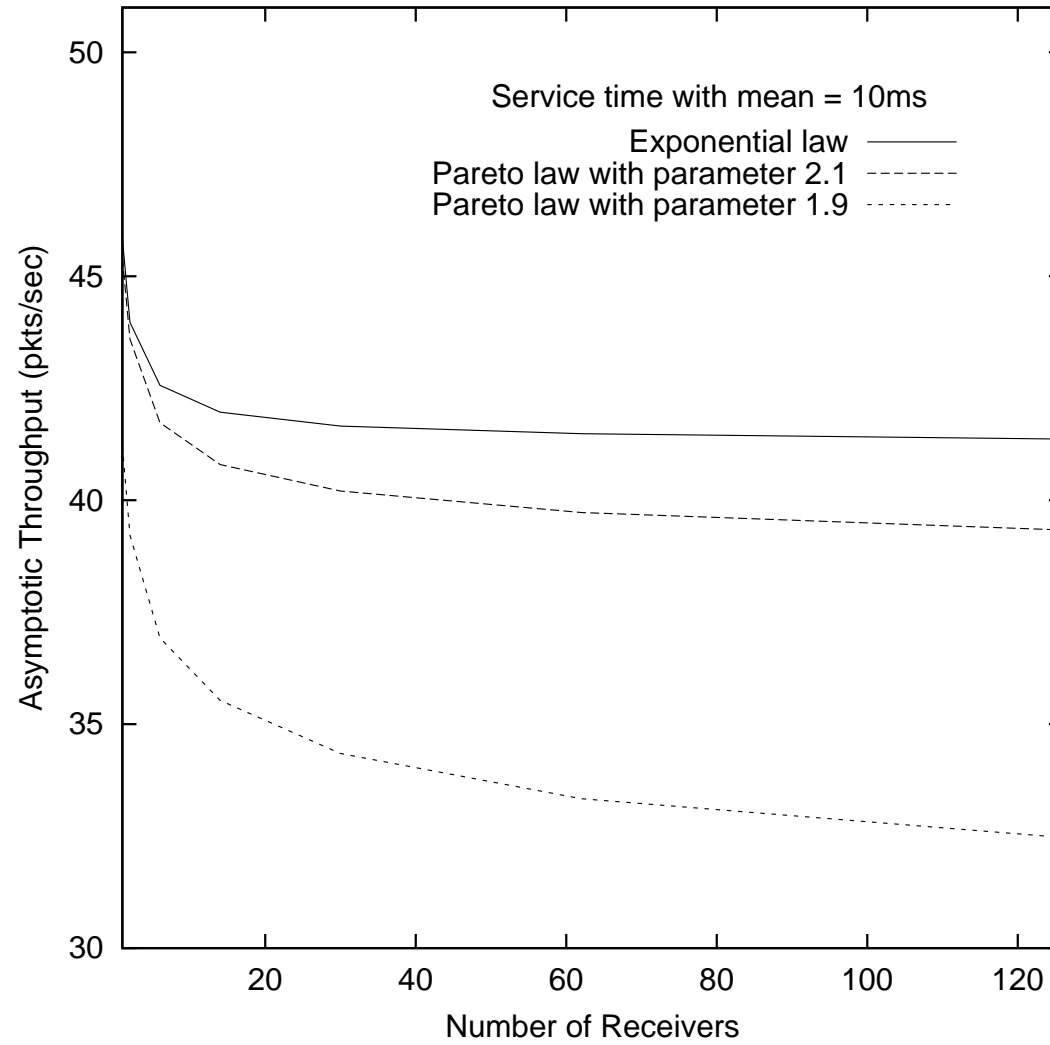
Simulation: Ratio of Asymptotic Throughput / One-Connection Throughput

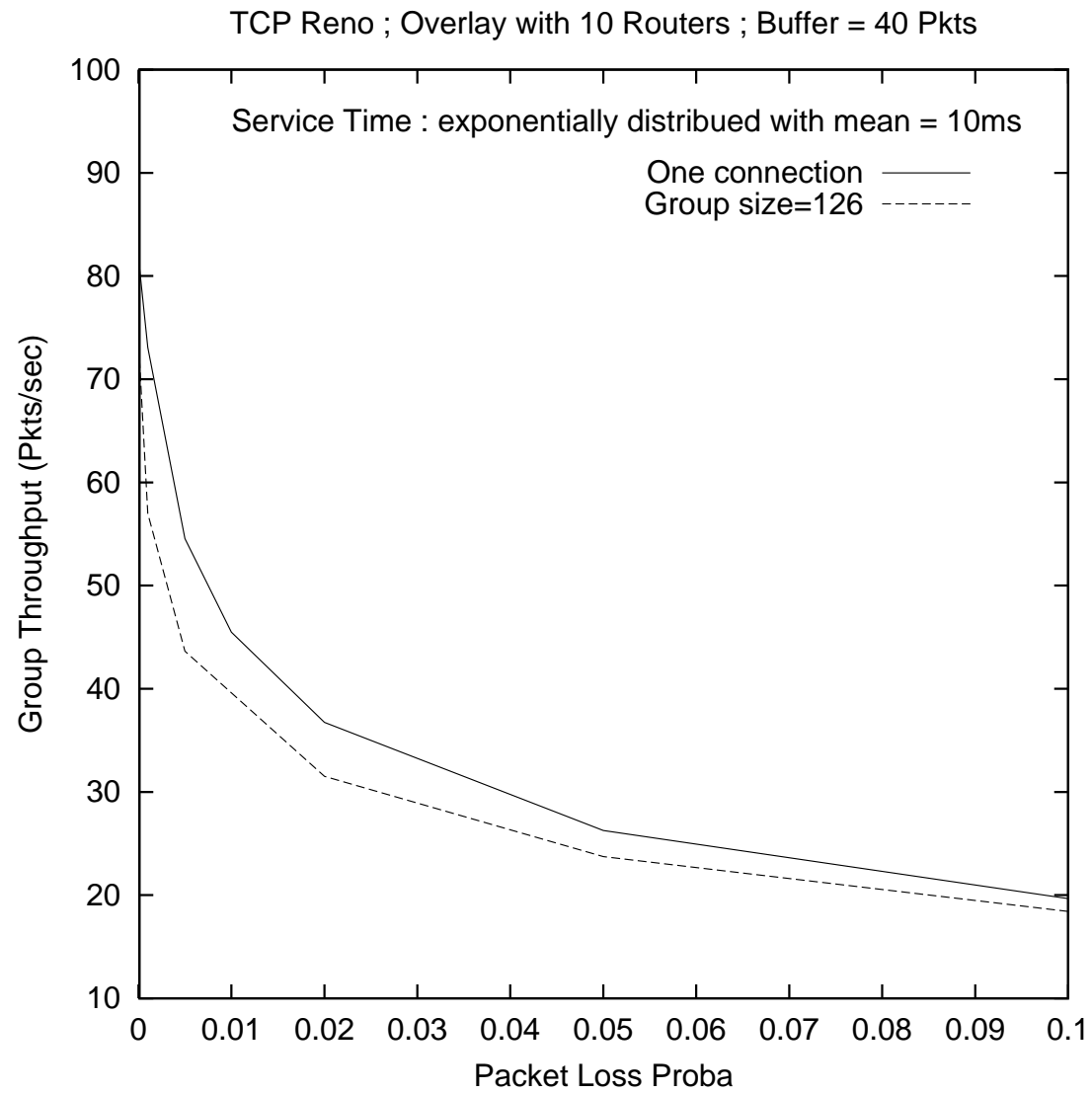
Buffer (Pkts)	10,000	1,000	100	50
TCP RENO	.99	.98	.90	.83
TCP ECN	.99	.99	.92	.87

Simulation: Influence of Fluctuations

- Law of cross traffic
- Packet loss probability

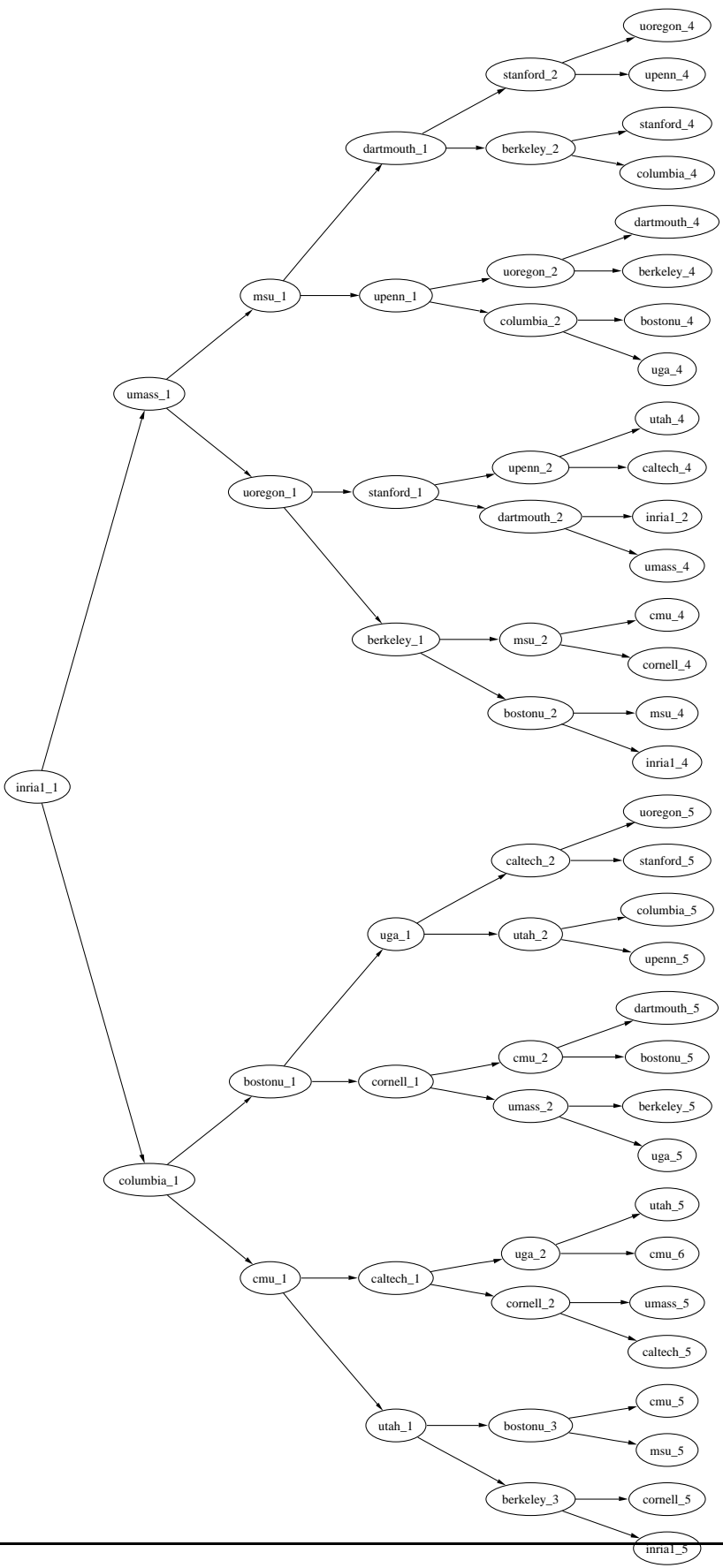
TCP Reno, Overlay : 10 Routers, $W_{\max} = 40$, Loss Proba = 0.01, Buffer=100Pkts





Planet-Lab Experiments

■ Multicast tree consisting of 64 PlanetLab nodes



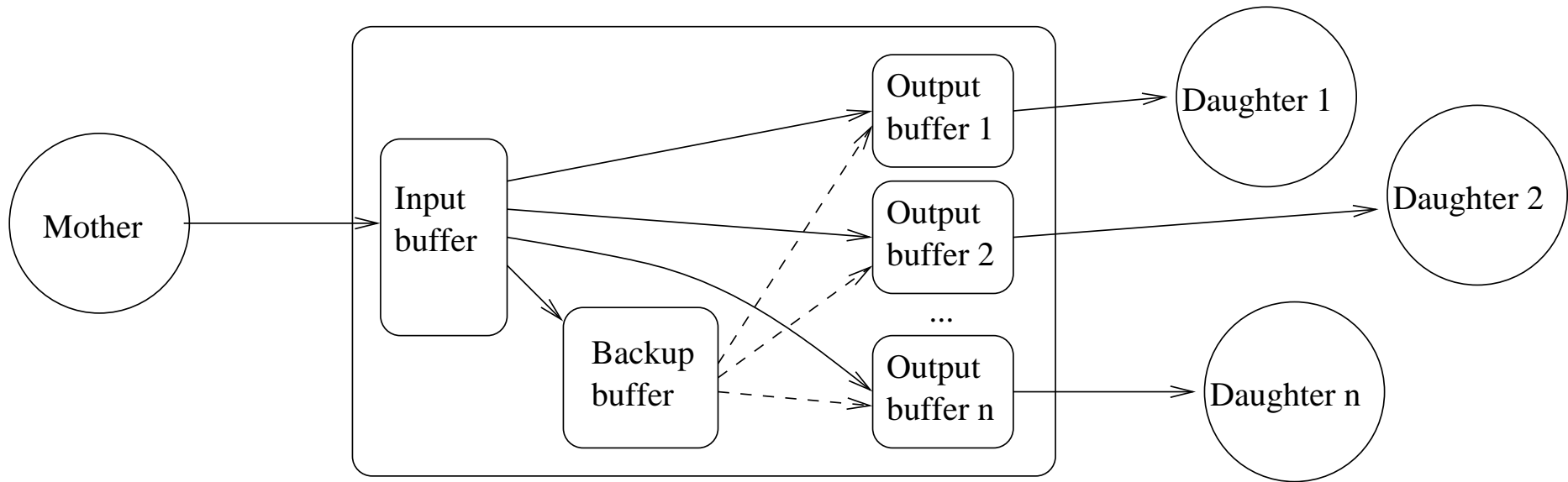
Planet-Lab Experiments

- Size of each packet is 200 bytes

Group size:	15	31	63
Buffer=50 Pkts	95	86	88
Buffer=100 Pkts	82	88	77
Buffer=1000 Pkts	87	95	93

- The absolute numbers are different but again the group throughput changes very little with the group size.

Handling Node Failures: backup buffer



End-to-end reliability

Theorem 2: An overlay multicast system with backup buffer of size $(m \cdot (B_{OUT}^{\max} + B_{IN}^{\max}) + B_{OUT}^{\max})$ is end-to-end reliable with tolerance to m failures.

TABLE III

END-TO-END RELIABILITY EXPERIMENTS IN PLANET-LAB

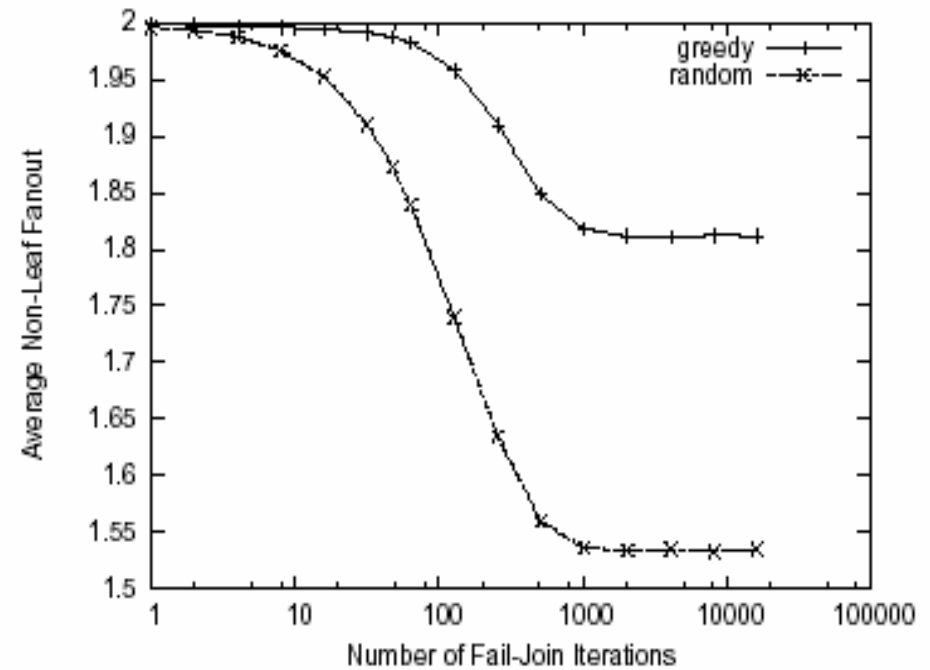
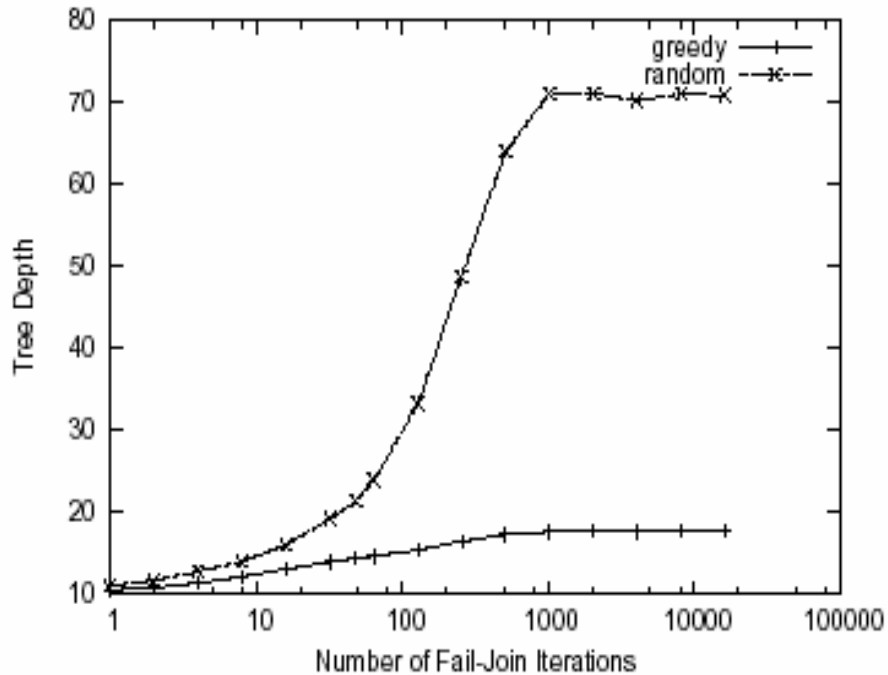
	min	average	max
Throughput (Pkts/sec)	49.05	55.24	57.65
# of Retransmitted Packets	34	80.5	122
Reconnection time (D)	0.12	3.53	5.2
Reconnection time (M)	0.27	3.81	5.37

Handling Leave/Join/Node Failures

Algorithm GREEDY_RECONNECT

1. Suppose node (k, l) fails. Let \mathcal{S} be the set of orphaned subtrees, rooted at daughters of (k, l) . Let \mathcal{A} be the set of active nodes in subtree of $(k - 1, m(k, l))$, but not in the subtree of (k, l) .
2. Choose a node $(k + 1, l') \in \mathcal{S}$ that has subtree of largest depth.
3. Choose a node $(p, q) \in \mathcal{A}$ that is closest to the source.
4. Connect $(k + 1, l')$ to (p, q) .
5. Update $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(k - 1, l')\}$ and add active nodes from subtree of $(k + 1, l')$ to \mathcal{A} .
6. If \mathcal{S} is not empty, go to Step 2.

Solution Features: balanced tree with bounded degree



The Infinite Buffer Case: Scalability of Throughput

- The infinite buffer case can be studied on a **line** rather than a tree without loss of generality.
 - θ_k : local saturated throughput of TCP connection k .
 - The throughput in the line at node K is given by the minimum of the saturated local throughputs : $\Theta_{1,K}^\lambda = \min(\lambda, \theta_1, \dots, \theta_K)$

The throughput is positive for groups of any size, if the local saturated throughputs are all lower bounded by θ .

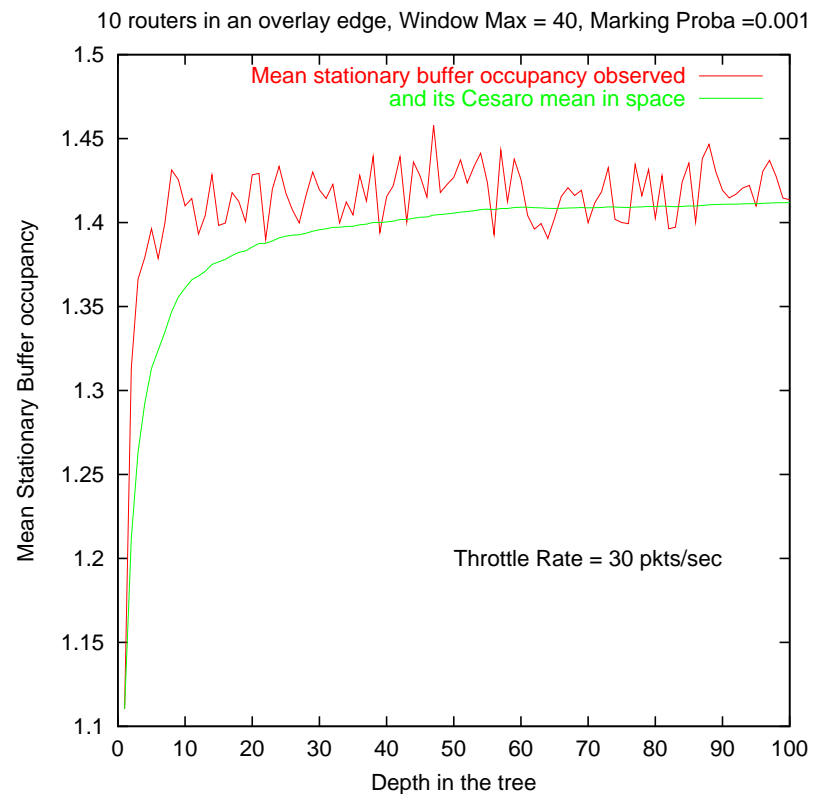
Infinite Buffer Case: Need for a Source Rate Control

- In order to avoid buffer occupancy explosion, it is necessary to **control the rate** at which the source sends data.
- For $\lambda < \theta$, for all K , the latency $L_m^{(K)}$ of packet m from end-system K to $K + 1$ converges to a stationary law when m tends to ∞
- **Question** How does this stationary law evolve when K is growing?
- We focus on
 - the marking case
 - the homogeneous case: $H_k = H$, $p_k = p$, $s_m^{(k,h)} \stackrel{d}{=} s$

Mean Buffer Occupancy : Empirical Study

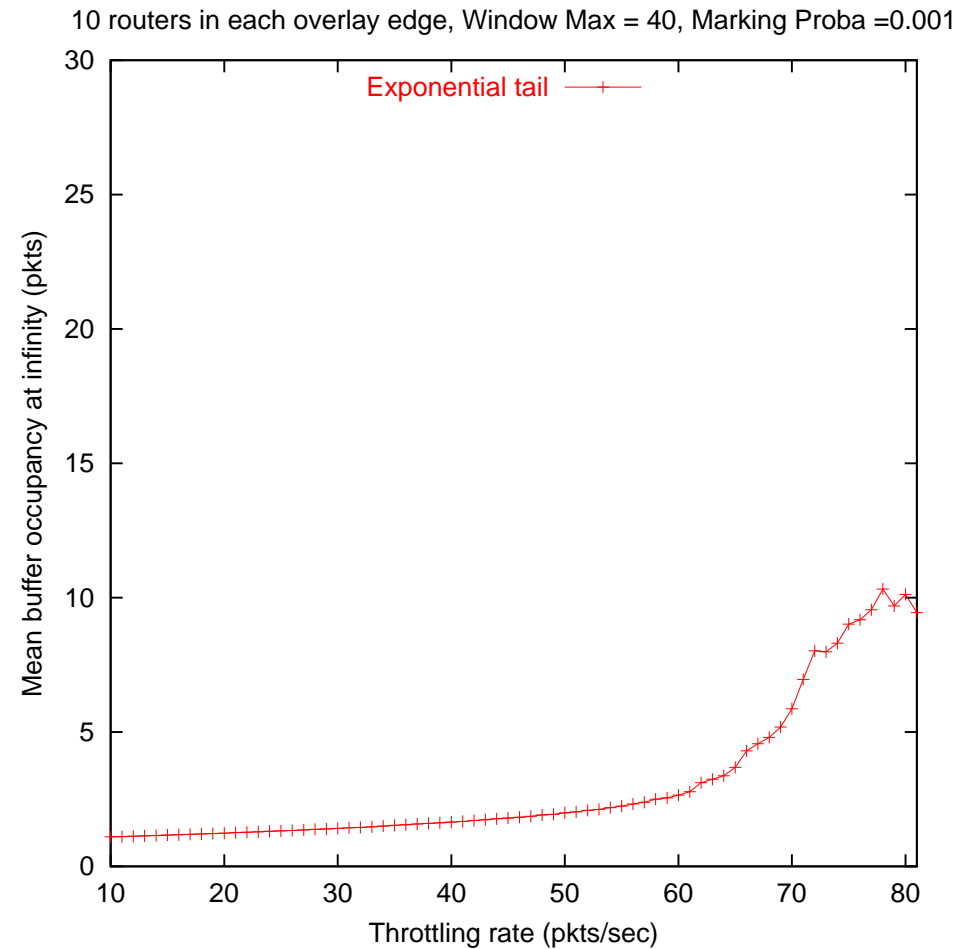
– Numerical computation of the (max,plus) equation.

- Stationary mean buffer occupancy w.r.t. the distance from the source.
- Convergence when the distance tends to infinity.



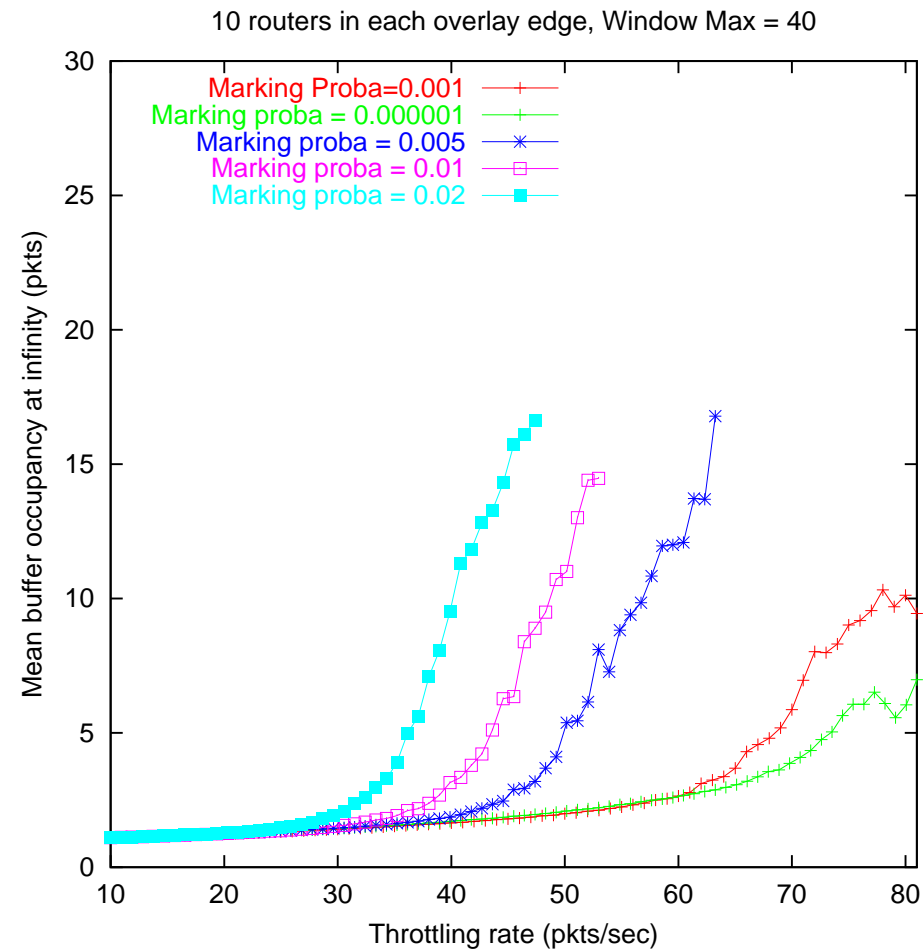
Mean Buffer Occupancy : Empirical Study (cont'd)

- “Infinite time” - “Infinite space” limit mean buffer occupancy, w.r.t. throttling rate.



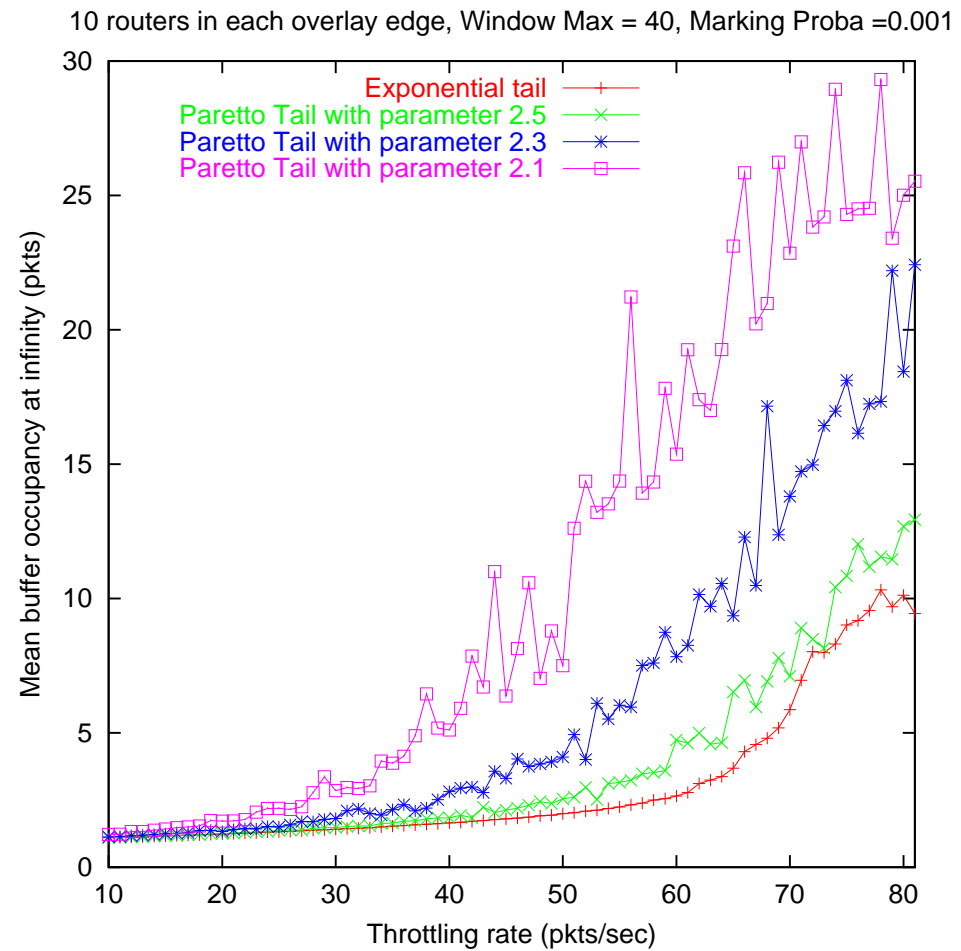
Mean Buffer Occupancy : Empirical Study (cont'd)

- Several packet marking probability cases.



Mean Buffer Occupancy : Empirical Study (cont'd)

- Several cross traffic distribution cases.



Mean Buffer Occupancy : Condition on law s

- For $\mathbb{E}[s^2] = \infty$, the stationary law of buffer occupancy (as well as latency) never admit a finite mean.
- For $\mathbb{E}[s^2] < \infty$, the mean buffer occupancy in end-system K and the expected latency from the source to end-system K is finite ... but nothing can be said for $K \rightarrow \infty$.
- From now, we assume : $\int_0^{+\infty} P(s \geq u)^{1/2} du < \infty$.

(It is implied by $\mathbb{E}[s^2 (\log(s))^{2+a}] < \infty$, for $a > 0$ and by $\mathbb{E}[s^b] < \infty$, for $b > 2$).

Mean Buffer Occupancy : Mathematical Justification

- Define $\gamma(u) = \lim_{k \rightarrow \infty} \frac{x^{(k,l,end)}_{[xk]}}{k}$.
- The Legendre transform of γ gives the limit with K (in Cesaro sense) of the stationary latency $L^{(K)}$ of level K when the source rate is throttled to λ :

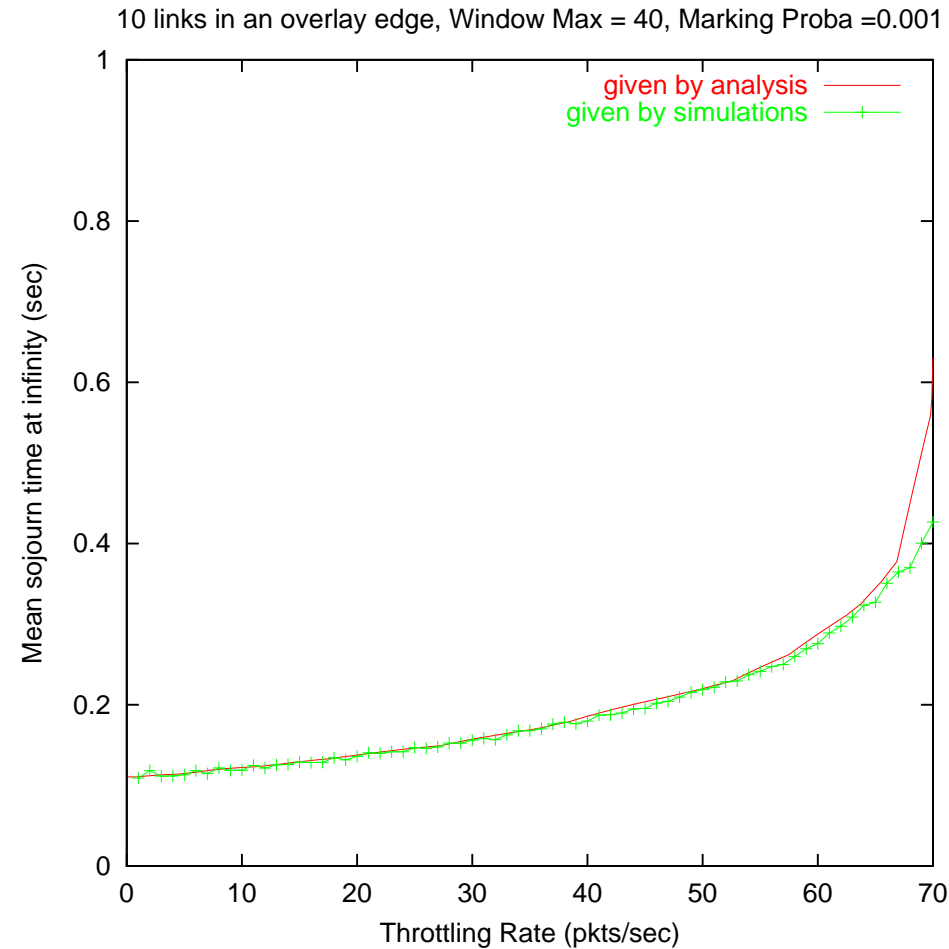
$$\lim_{K \rightarrow \infty} \frac{L^{(K)}}{K} = d(\lambda) = \sup_{x > 0} \left(\gamma(x) - \frac{1}{\lambda}x \right) .$$

- Multiplied by λ , one gets a bound on the limit with K (in Cesaro sense) of the mean stationary buffer B_K used in an end-system with level K .

$$\lim_{K \rightarrow \infty} \frac{B^{(k)}}{K} \leq \lambda \cdot d(\lambda)$$

Mean Buffer Occupancy : Numerical Validation

- Two methods are used to evaluate $d(\lambda)$.
- Legendre transform gives an efficient way to estimate $d(\lambda)$.



Optimal Tree Construction

- Complete graph $G = (V, E)$
 - Nodes = end-systems; numbered from 1 to n , where node 1 is the root.
 - Edges: each pair of nodes $i, j \in V$ is connected via a route in the Internet with local saturation throughput θ_{ij}
- Question (infinite buffer - rate control case): Find a tree from the root with maximum group throughput, where group throughput is the minimum of all path throughputs in the tree.

Optimal Tree Construction (<i>continued</i>)
--

- Model I: Access Link not the Bottleneck
- Algorithm to construct a tree with optimal group throughput:
 - Sort all $n(n - 1)/2$ edges in increasing (local maximal) throughput order
 - Discard edges starting with those with the smallest throughput until the set of remaining edges on the n nodes makes a connected graph;
 - Build a spanning tree rooted in the source using the remaining edges of the sorted list.

The resulting spanning tree is optimal.

Optimal Tree Construction (*continued*)

- Model II: Accounting for Bottleneck at Access Link
- The constraints on node throughput for each node i can be treated as degree constraints
- The decision problem under this setting is a generalization of the minimum degree spanning tree
- The problem is provably NP-hard.
- Approximation algorithm with polynomial running time also proposed.

Conclusions

- Reliable multicast overlays can be deployed on top of the current TCP/IP
- Throughput scales provided all point to point connections that are used within the overlay offer minimal quality guarantees
- This conclusion holds true even in the case of moderate input and output buffers
- Need of a rate control in case of infinite buffers for guaranteeing scalable latency.
- General methodology (based on links between large overlay networks and statistical physics) applicable to other overlay networks.