Western

UNIVERSITY · CANADA

Multihoming: Scheduling, Modelling, and Congestion Window Management

Abdallah Shami

T. Daniel Wallace

Simon Fraser University

July 2013

Western Engineering

Department of Electrical and Computer Engineering

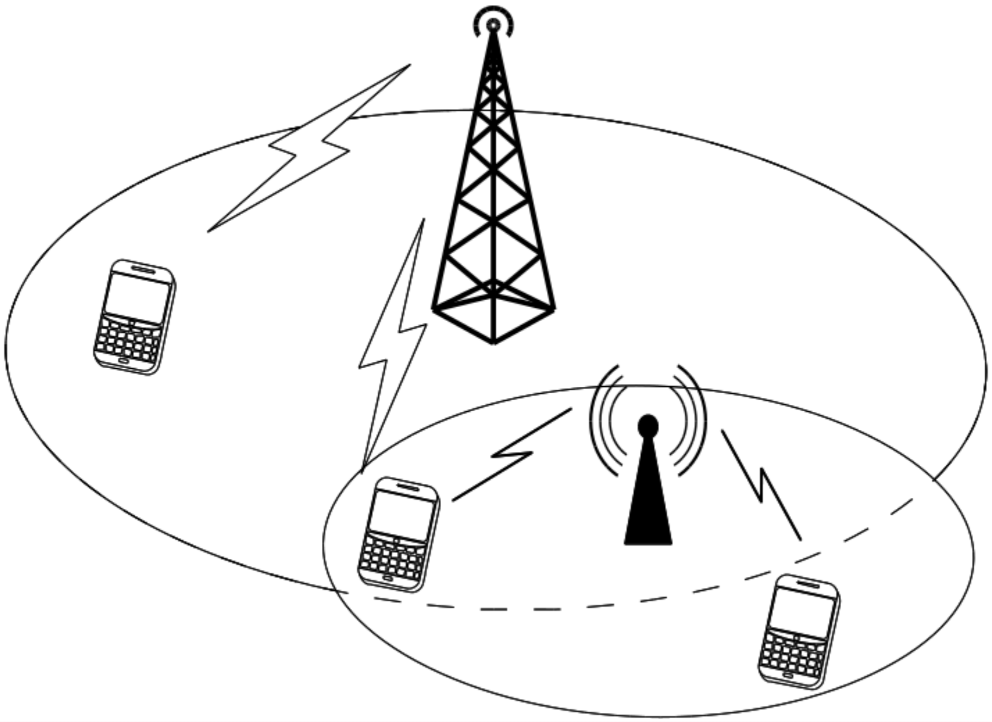
Talk Outline

- Motivation/Introduction
- A Review of Multihoming Issues using SCTP*
- CMT: Scheduling
- CMT: Modelling
- CMT: Congestion Window Management
- Conclusion

*T. D. Wallace, and A. Shami, “A review of multihoming issues using the stream control transmission protocol,” IEEE Communications Surveys & Tutorials, vol. 14, issue 2, pp. 565-578, 2012.

Multihoming

- Computing devices with multiple network interfaces.
  - e.g., the BlackBerry and iPhone include 802.11 (WLAN) and GSM/UMTS (cellular network) technologies.
  - All laptops have WiFi and Ethernet.



Concurrent Multipath Transfer (CMT)

- Goal: Take advantage of multiple network paths, between end-points, to increase application throughput.

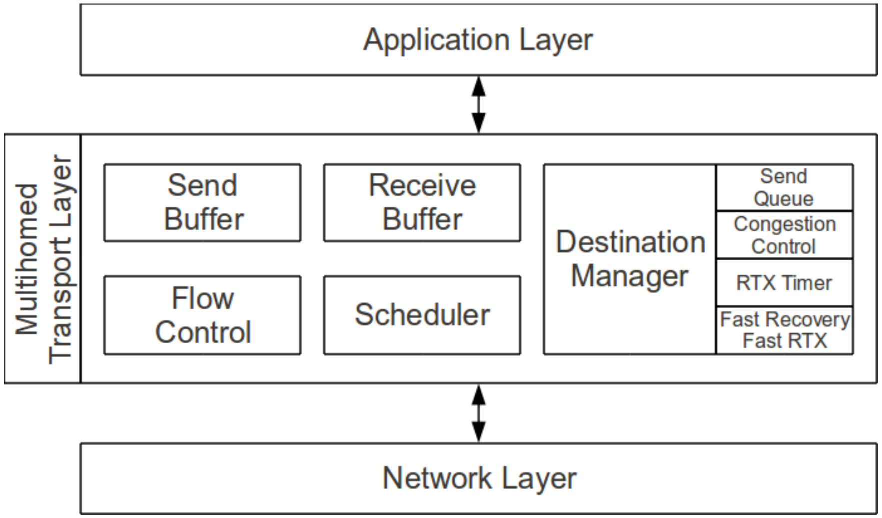
- Architecturally: works from the transport layer in the OSI model.

- Congestion control is managed on a per destination basis, but flow control is handled at the session layer (i.e., only one receive buffer).

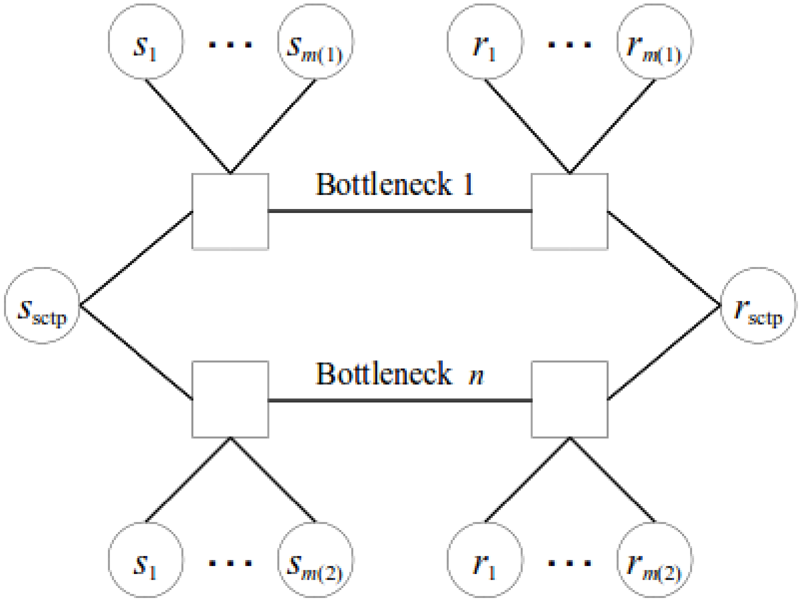
Stream Control Transmission Protocol

- An Internet Engineering Task Force (IETF) project since 2000 (RFC 4960).
- Implemented in many operating systems (e.g., Linux, Mac OS, FreeBSD, Solaris, Windows).
- TCP Similarities:
  - Reliable transport protocol.
  - Ordered delivery.
  - Implements congestion control and flow control.
- Main difference: SCTP supports multihoming.
  - Allows application data to be transmitted to multiple IP addresses.
  - Most useful for vertical handoffs and network faults.
  - Provides the basics to implement CMT.

System Description



Transport Layer Architecture



Multihomed Network Topology

General Terms & Concepts

- Congestion Window (CWND)
  - A variable that controls the amount of data that can be in flight.
- Send Buffer (SBUF)
  - The amount of memory allocated to accept data from the application layer before it's send into the network.
- Receive Buffer (RBUF)
  - The amount of memory allocated to accept data from the network before passing it to the application layer.
- Receive Window (RWND)
  - The available space in the RBUF.
- Throughput
  - The rate that data arrives at the receiver.

Multihoming: Problems, Issues, and Challenges

- Handover Management
  - Preemptive Path Selection
  - Fault Tolerant Path Selection
  - Post Handover Synchronization
- Concurrent Multipath Transfer
- Cross Layer Activities
  - Bandwidth estimation
  - Wireless error notifications
  - Network intelligence

Talk Outline

- Motivation/Introduction
- A Review of Multihoming Issues using SCTP*
- **CMT: Scheduling**
- CMT: Modelling
- CMT: Congestion Window Management
- Conclusion

CMT: Scheduling

- Scheduling & Transmission Basics
- Current Scheduling Approaches for CMT
- On-demand Scheduling
- Performance Results
- Summary & Contributions

Scheduling & Transmission Basics

- Data arrives from the application, fragmented into packets, then waits in the SBUF for a transmission opportunity.

- Transmission opportunities occur when:
  - 1) CWND must be greater than the number of packets it has in flight.
  - 2) RWND is greater than zero.

- Cumulative packet is transmitted.
  - The packet with lowest sequence number in the SBUF that has yet to be sent.

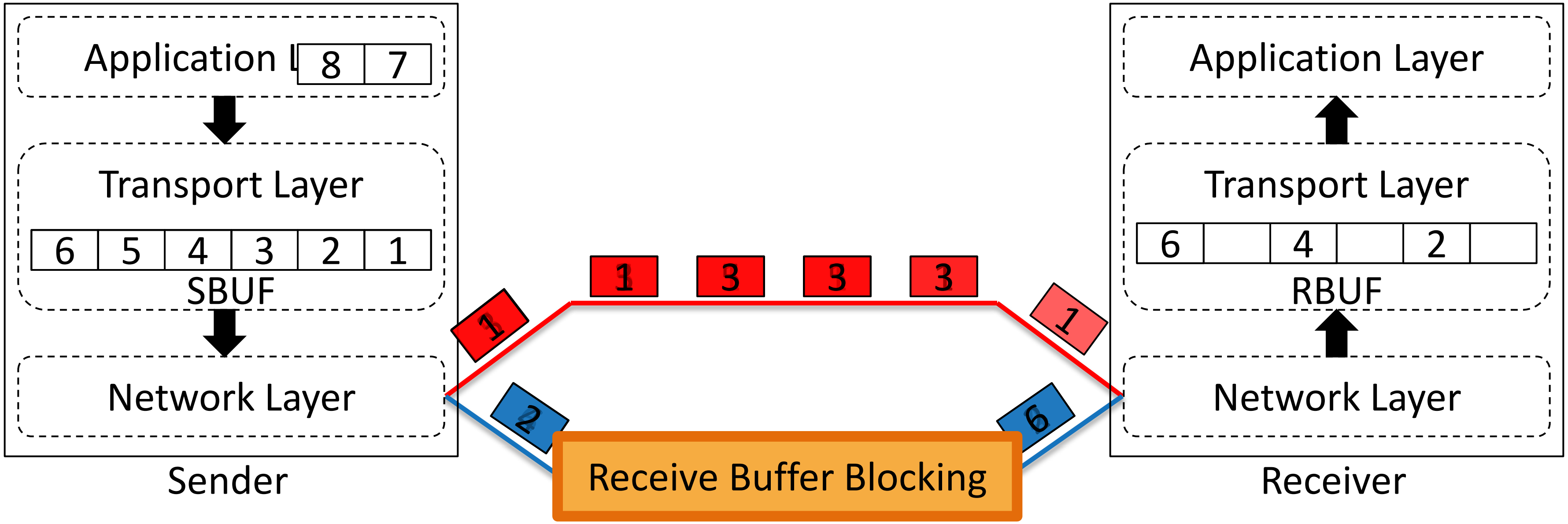
Current Scheduling for CMT

- Naïve Round Robin Scheduling*
  - No intelligence is used during the scheduling process.
  - When only one destination has a transmission opportunity, uses the basic scheduling and transmission technique.
  - When multiple destinations have transmission opportunities, packets are transmitted to each destination in a round robin fashion.

*J. Iyengar, P. Amer, and R. Stewart, “Concurrent multipath transfer using sctp multihoming over independent end-to-end paths,” IEEE/ACM Trans. Netw., vol. 14, no. 5, pp. 951–964, 2006

Current Scheduling for CMT

- Naïve Round Robin Scheduling
  - Assume bandwidth is the same on either path, but the delay on the red path is twice as long as the blue path.
  - Both paths always have transmission opportunities.



Current Scheduling for CMT

- Bandwidth Aware Scheduler (BAS)*
  - Attempts ordered packet delivery using bandwidth estimates.
  - Scheduling decisions are made before transmission opportunities
  - When packets arrive at the SBUF, they are assigned to a destination's send queue
  - Assignments are based on a *reception index*.

$$R(d) = \frac{L(p) + O(d) + S(d)}{B(d)}$$

$p \equiv$ a new packet

$d \equiv$ a destination address

$L(p) \equiv$ returns the size of packet p

$O(d) \equiv$ returns the number of packets (or bytes) in flight to destination d

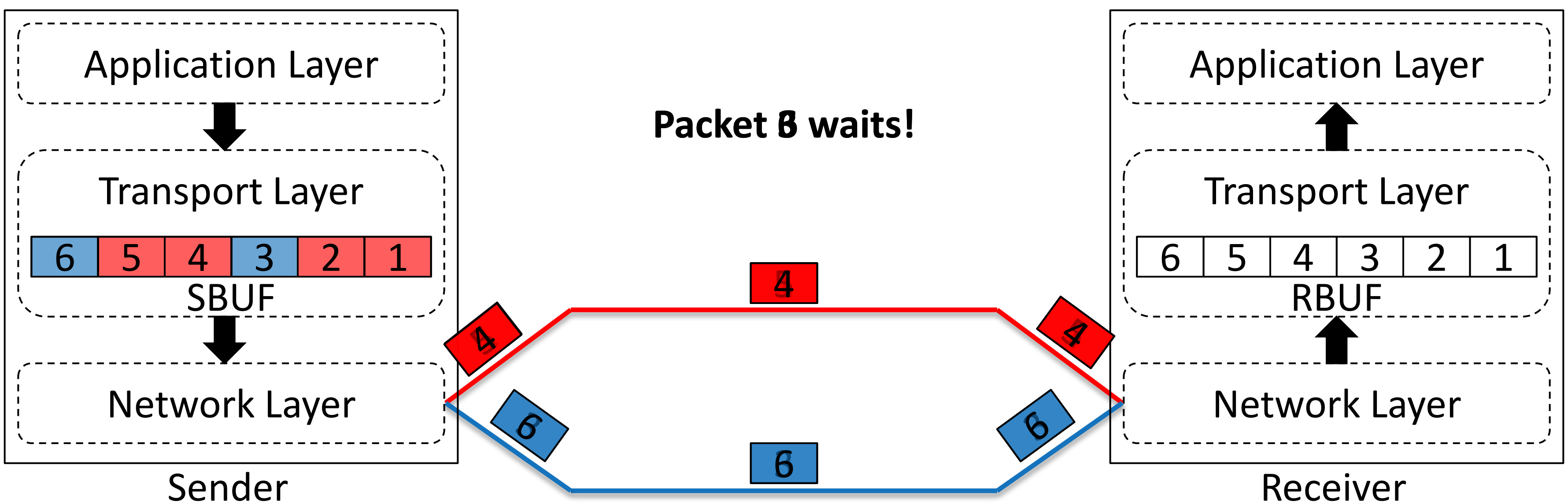
$S(d) \equiv$ returns the number of packets in the send queue for destination d

$B(d) \equiv$ returns a bandwidth estimate for destination d

* M. Fiore, C. Casetti, and G. Galante, "Concurrent multipath communication for real-time traffic," Comput. Commun., vol. 30, no. 17, pp. 3307–3320, 2007.

Current Scheduling for CMT

- Bandwidth Aware Scheduler
  - Assume bandwidth on the red path is twice the speed of the blue's, but both experience the same propagation delay.
  - Initially, CWND on the red path is only 1 packet.



On-demand Scheduler

- Goal: Find the cumulative packet (in SBUF) that cannot be delivered to any other destination sooner.
  - ODS waits for a transmission opportunity before it makes its scheduling decision.
  - Uses bandwidth and propagation delay to manufacture a packet's estimated time of acknowledgement (ETA).
  - Recursively simulates the transmission and acknowledgement of packets in the SBUF.

On-demand Scheduler

- Search Algorithm

Copy send buffer.

(1) Search Initialization:

- let d^{NS} be the destination looking for a packet at the start of a new search;
- let t^{NS} be the starting time of a new search;
- let Q be a copy of the send buffer at the start of a new search;
- let D be a copy of all destination state at the start of a new search;
- let D' be a the set of destinations with earlier ETAs at the start of a new search;
- goto (2);

Find outstanding packet with lowest ETA and simulate ACK.

(2) Simulating Acknowledgements:

if all packets in Q have been transmitted or acknowledged **then**
    exit search with failure;
else
    let a be the packet in Q with the earliest ETA such that $a.sent$ = TRUE and $a.acked$ = FALSE;
    set $t = a.eta$;
    if $t < t^{\mathrm{NS}}$ **then**
        set $t = t^{\mathrm{NS}}$;
    end if
    set $d = a.dest$;
    increment $d.pba$;
    update $d.cwnd$;
    remove all cumulative packets from Q;
    goto (3);
end if

Calc ETAs for each destination.

(3) Calculating ETAs:

let p be the packet in Q with the lowest sequence number such that $p.sent$ = FALSE;
compute $A(p, d^{\mathrm{NS}}, t^{\mathrm{NS}})$;
for each destination d in D' **do**
    compute $A(p, d, t)$;
end for
if d^{NS} has the lowest ETA at time t **then**
    return with s
else
    remove any destinations from D' with later ETAs than d^{NS};
    goto (4);
end if

(4) Simulating Transmissions:

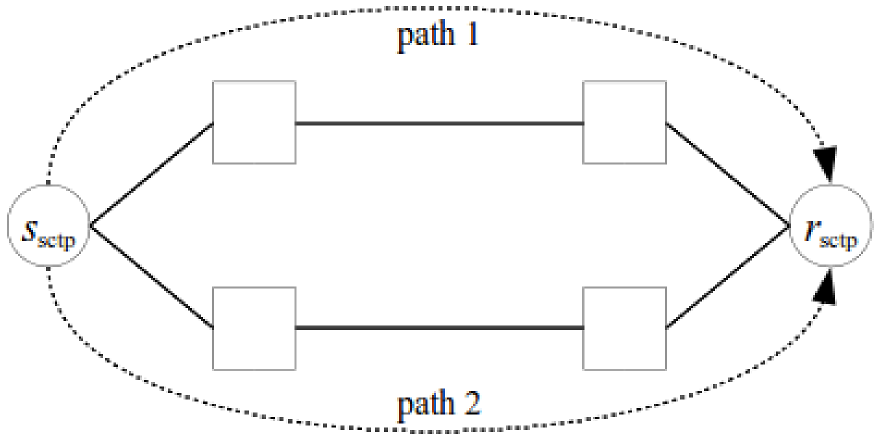
let D_t^{Tx} be the set of destinations at time t with a transmission opportunity;
if $D_t^{\mathrm{Tx}} \neq \{\emptyset\}$ **then**
    if $d_{\min} \neq d_{\min}^{\mathrm{Tx}}$ **then**
        start a new search for $d_{\min}^{\mathrm{Tx}}$ using Q, D, and D' at time t;
        goto (1);
    end if
    set $p.dest = d_{\min}^{\mathrm{Tx}}$;
    set $p.eta = A(p, d_{\min}^{\mathrm{Tx}}, t)$;
    set $p.sent$ = TRUE;
end if
goto (2);

Simulate transmission.

Start a new search.

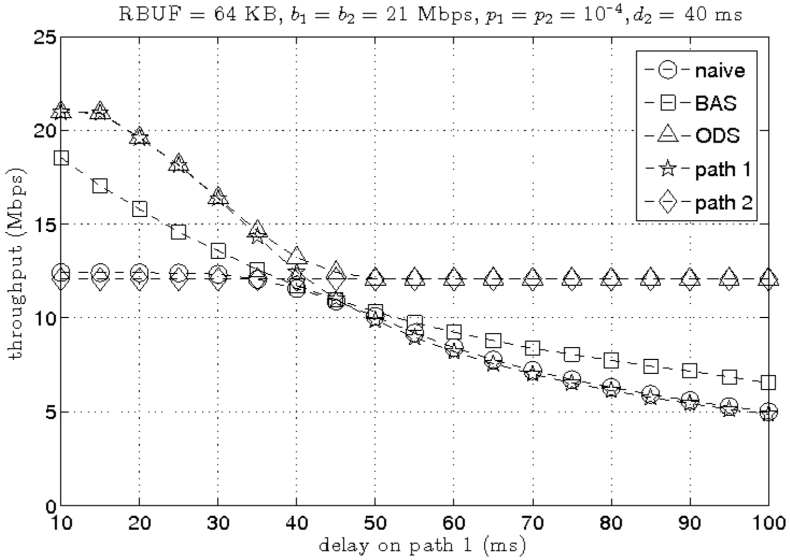
Performance Results

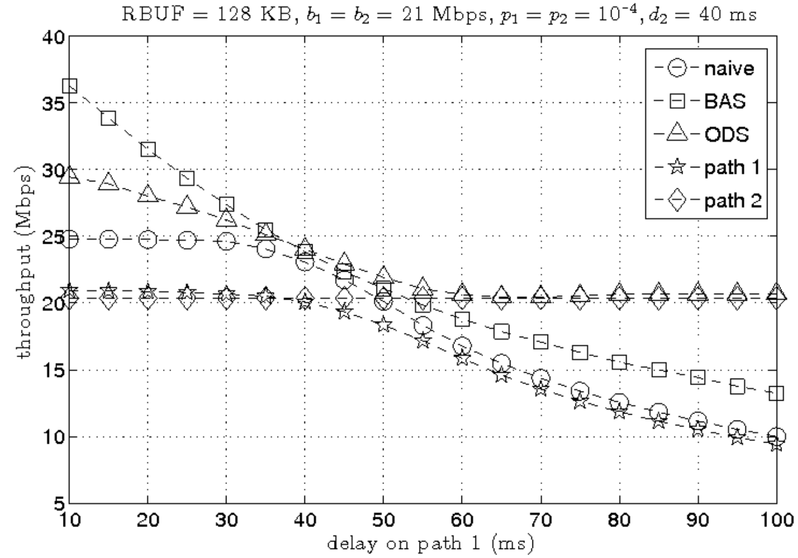
- Network Topology

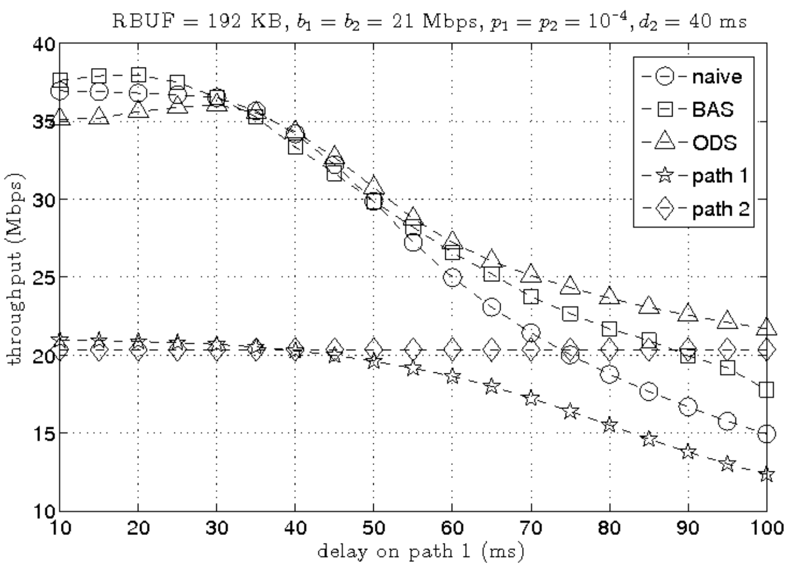


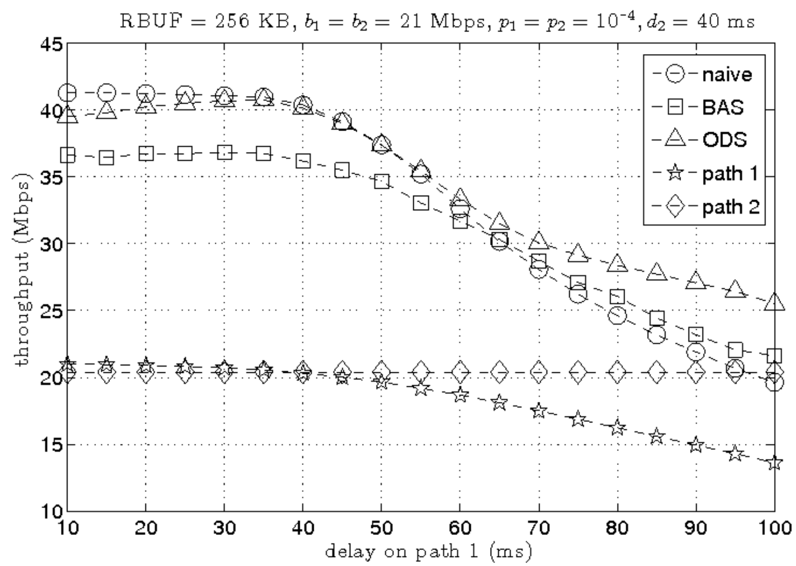
- Simulation
  - Implemented each scheduling algorithm in ns-2.
  - Created a variety of network scenarios to evaluate CMT: delay-based disparity, bandwidth-based disparity, loss-based disparity, different RBUF sizes.
  - Simulated a 1 GB file transfer.

Performance Results









Summary

- Developed a new scheduling algorithm for CMT called On-demand Scheduling (ODS).

- Compared each scheduling algorithm.
  - ODS will often improve performance, especially when the system is constrained by a limited RBUF.
  - BAS is only suitable when the RBUF is very large and preferable when there is a minor disparity in path delays.
  - Naïve scheduling is best when there is minimal difference in delays.

- Evaluated ODS under different network scenarios:
  - delay-based disparity (significant improvement)*
  - bandwidth-based disparity (some improvement)
  - loss-based disparity (still an open problem)

*will revisit later

CMT: Modelling

- Modelling Framework
- Markov Model
- Renewal Model
- Performance Results
- Summary & Contributions

Modelling Framework

- Goal:
  - Given a multihomed system, approximate the throughput of a long-term session employing CMT.
- Parameters:
  - bandwidth (per path)
  - delay (per path)
  - probability of packet loss (per path)
  - RBUF size
- Approach:
  - Model independent SCTP sessions using techniques from TCP literature, then aggregate throughput predictions.
- Assumptions:
  - Perfect scheduling.
  - The CWND must be less than or equal to the RBUF.

Markov Model

- Discrete-time Markov Chain (DTMC)
  - SCTP is represented as a set of discrete transmission rounds.
  - Each round is a state in the Markov chain.
  - During each round some number of packets are transmitted, where some or all of those packets might be lost.
- States:
  - (ω,ξ,τ)
  - ω = size of the CWND during a round
  - ξ = number of packets transmitted during a round
  - τ = slow-start threshold during a round
- Operating Modes:
  - Congestion Avoidance (CA)
  - Exponential Backoff (EB)
  - Slow-start (SS)

Markov Model

- States

$$\mathcal{C} = \left\{(\omega, \xi, \tau) : 2 \le \omega \le \omega_{\max}, 1 \le \xi \le \omega, \omega \in \mathrm{Z}, \xi \in \mathrm{Z}\right\}$$

$$\mathcal{S} = \left\{(\omega, \xi, \tau) : \omega = \xi = 2^i, 1 \le i \le \log_2(\tau), \tau \in \mathcal{T}, \tau > 1, i \in \mathrm{Z}\right\}$$

$$\mathcal{E} = \left\{(0, \xi, \tau) : 2 \le \xi \le M, \tau = 1) \cup (\xi = 1, \tau \in \mathcal{T})\right\}$$

- Steady-state probability

$$\pi = \pi \cdot \mathbf{Q}$$

- Throughput

$$\eta = \frac{E[\xi] - E[\gamma]}{E[\delta]} \quad \begin{cases} E[\xi] = \sum_{i \in \{\mathcal{C}, \mathcal{S}\}} \pi(i)\xi(i) + \sum_{i \in \mathcal{E}} \pi(i) \\ E[\gamma] = \sum_{i \in \{\mathcal{C}, \mathcal{S}\}} \pi(i) \sum_{j}^{\xi(i)} jP(j, \xi(j)) + \sum_{i \in \mathcal{E}} \pi(i) P(1,1) \\ E[\delta] = \sum_{i \in \{\mathcal{C}, \mathcal{S}, \mathcal{E}\}} \sum_{i' \in \{\mathcal{C}, \mathcal{S}, \mathcal{E}\}} \pi(i) \cdot D(i, i') \end{cases}$$

Renewal Model

- Renewal Theory
  - A stochastic process continually restarts at regular intervals.
  - Formulate a closed-form expression to represent an SCTP session.

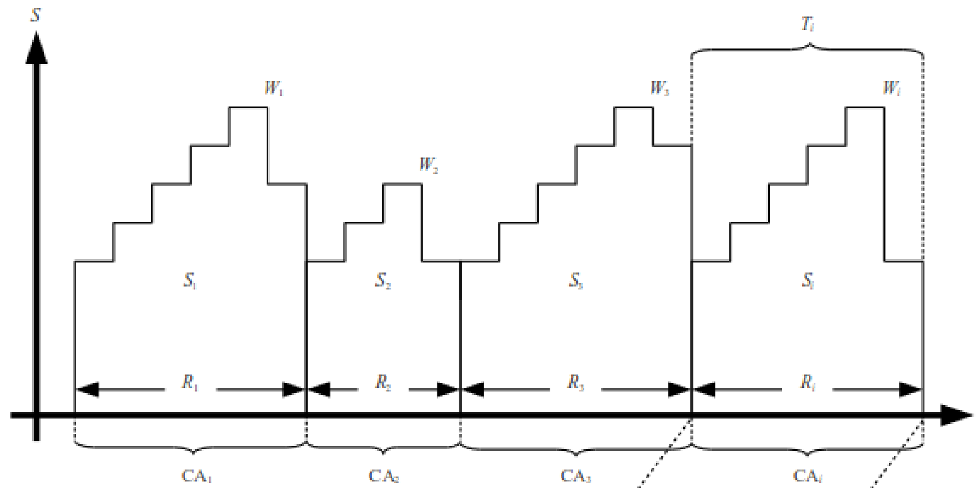
- Throughput is approximated by an average interval.
  - t = length of time of the average interval
  - S_t = number of packets transmitted during the average interval
  - L_t= number of packets lost during the average interval

- Operating Modes:
  - Congestion Avoidance (CA)
  - Exponential Backoff (EB)
  - Slow-start (SS)

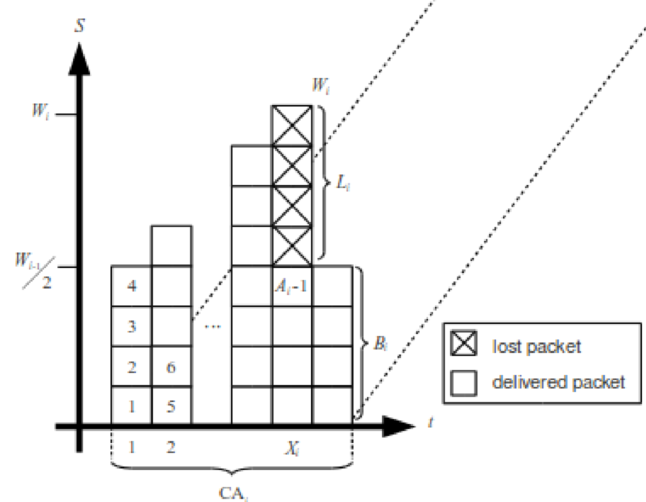
Renewal Model

- Congestion Avoidance (infinite RBUF, no timeouts)

$$\eta = \frac{E[S]-E[L]}{E[T]} \quad \left\{ \begin{aligned} E[S] &= \frac{1-p}{p} + E[W] \\ E[L] &= \frac{E[W]}{2} \\ E[T] &= (E[X]+1)\cdot \mathrm{RTT} \end{aligned} \right.$$

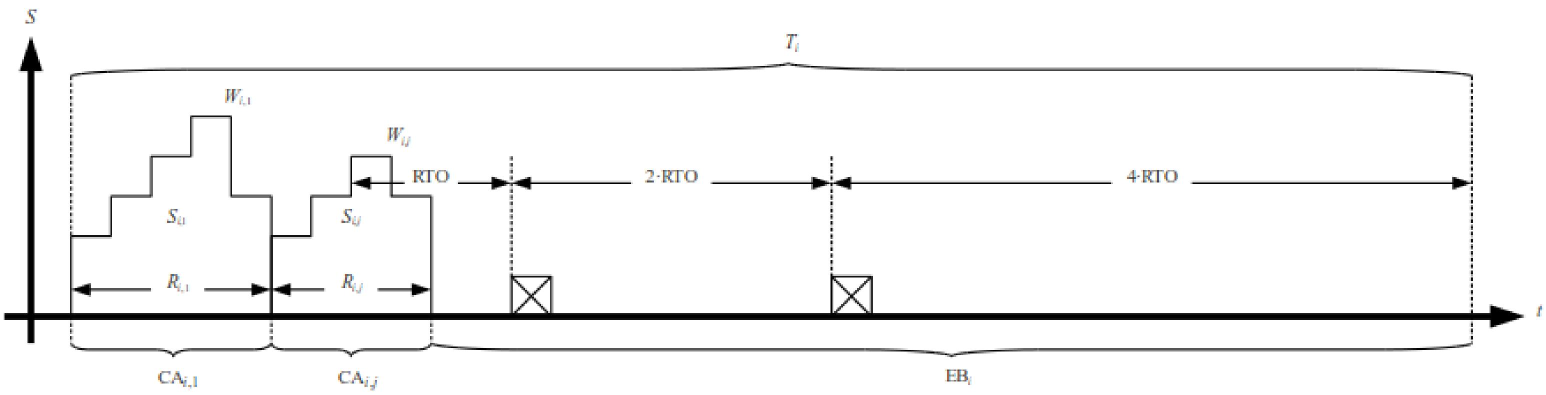


$$\left. \begin{aligned} E[W] &= \sqrt{\frac{8(1-p)}{3p} + \frac{1}{9}} - \frac{1}{3} \\ E[X] &= \sqrt{\frac{2(1-p)}{3p} + \frac{1}{36}} + \frac{5}{6} \end{aligned} \right\} \quad \boxed{\eta = \frac{2(1-p) + E[W]\cdot p}{2p\cdot(E[X]+1)\cdot \mathrm{RTT}}}$$



Renewal Model

- Exponential Backoff

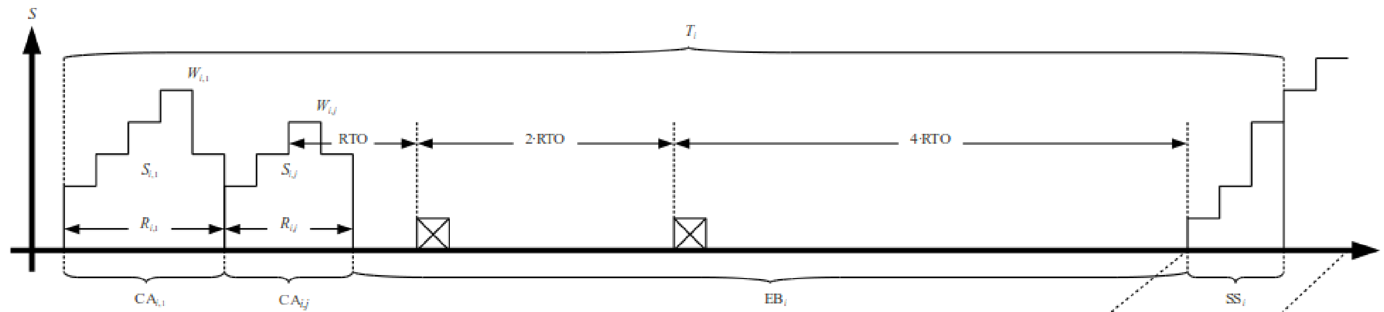


$$\eta = \frac{E[S^{\mathrm{CA}}] - E[L^{\mathrm{CA}}] + P^{\mathrm{TO}}}{E[T^{\mathrm{CA}}] + P^{\mathrm{TO}} \cdot E[T^{\mathrm{EB}}]}$$

$$\begin{cases} P^{\mathrm{TO}} = \dfrac{1-(1-p)^8 - (1-(1-p)^4)(1-p)^W}{1-(1-p)^W} \\[2ex] E[T^{\mathrm{EB}}] = \dfrac{(1-p)(1-2^M p^M)}{1-2p}\mathrm{RTO} + \dfrac{p^M}{1-p}\mathrm{RTO}_{\max} \end{cases}$$

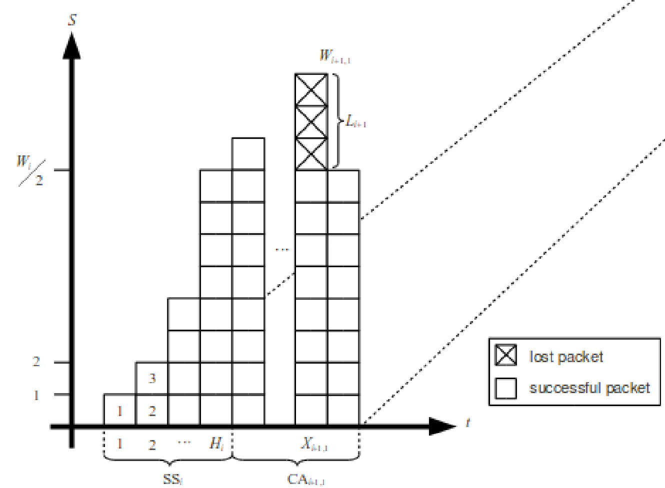
Renewal Model

- Slow-start

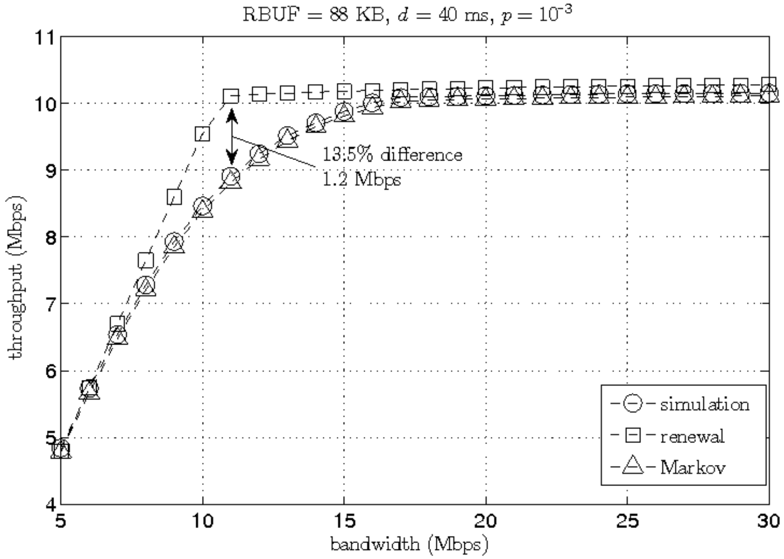


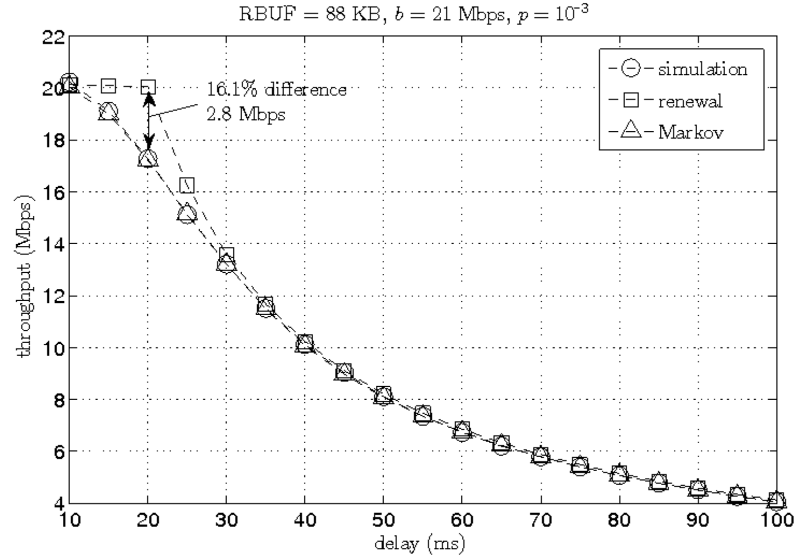
$$\eta = \frac{E[S^{\mathrm{CA}}] - E[L^{\mathrm{CA}}] + P^{\mathrm{TO}} \cdot E[S^{\mathrm{SS}}]}{E[T^{\mathrm{CA}}] + P^{\mathrm{TO}}(E[T^{\mathrm{EB}}] + E[T^{\mathrm{SS}}])}$$

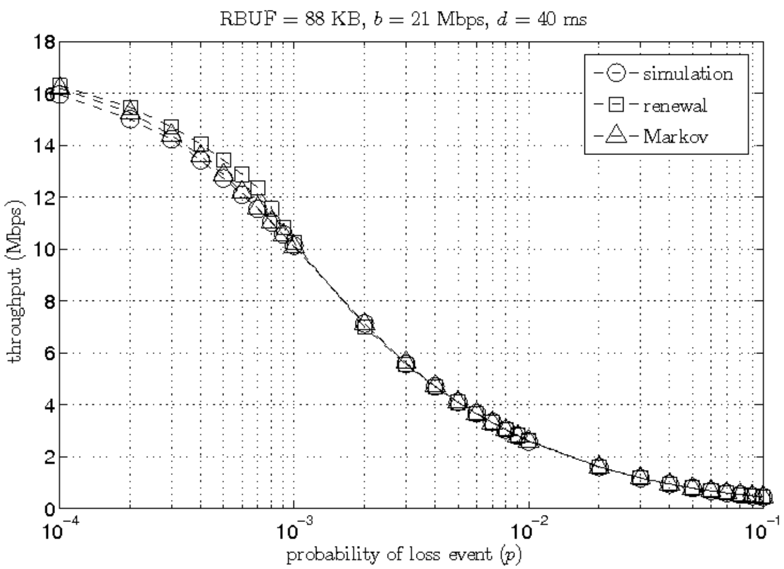
$$\begin{cases} E[S^{\mathrm{SS}}] = E[W] - 1 \\ E[T^{\mathrm{SS}}] = \max(1, \log_2 E[W]) \cdot \mathrm{RTT} \end{cases}$$

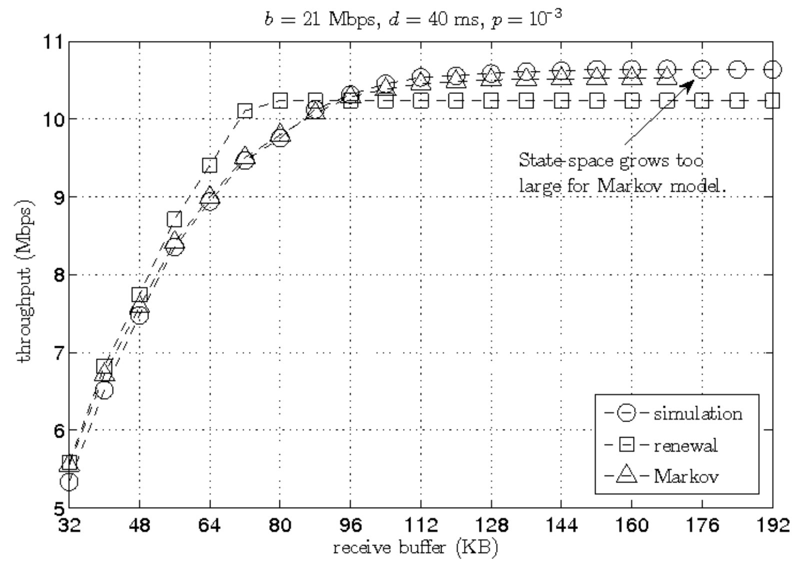


Performance Results









Summary

- Created a tractable framework to model the throughput of CMT.
- Developed two different models based on well-known techniques:
  - Discrete-time Markov Chain & Renewal Theory
- Compared the performance of both models with simulated results.
  - Markov model is more accurate but suffers from issues of scalability.
  - Markov model uses Gaussian Elimination to solve an unbounded matrix.
  - Renewal theory is more cost effective, but approximations are not always accurate.
  - Renewal theory approximates throughput using a closed-form expression.

CMT: Congestion Window Management

- CWND Update Policy
- CWND Optimization
- Performance Results
- Summary & Contributions

CWND Update Policy

- *policy*$_1$:SCTP's current CWND update policy
  - SCTP grows its CWND by 1 every RTT.
  - CWND is unbounded. Even when flow control stops packets from being transmitted, the CWND continues to grow.
- What impact will *policy*$_1$ have on CMT?
  - Lowers utilization and throughput potential.
  - One destination address can monopolize the RWND.
- Solution
  - Limit the sum of all CWNDs to the size of the RBUF.
  - Limit the size of a path's CWND to its corresponding bandwidth delay product (BDP).
  - Apply local or "greedy" optimization.
    - Rank destination addresses according to bandwidth potential.
    - Sets precedence to grow CWND's of higher ranked destinations first.

CWND Update Policy

- Algorithm name: $policy_2$.

Sum of CWNDS is less than the RBUF.

Limit a destination's CWND to its BDP.

Decrease CWND of lower ranked dest when CWND of higher ranked dest Is blocked.

```
let d be the destination updating its CWND;
let d.BDP be the bandwidth delay product of destination d;
let d.CWND be the congestion window of destination d;
let d.PBA be the value of partial.bytes.acked for destination d;
let CWND_sum be the sum of all congestion windows;
let RWND_max be the maximum size of the receive window;
let D be the set of all destinations with RTTs higher than d in descending order;
if d.PBA ≥ d.CWND then
  if CWND_sum < RWND_max then
    if d.CWND < d.BDP then
      increment d.CWND;
    else if d.CWND > d.BDP then
      decrement d.CWND;
    end if
  else if d.CWND < d.BDP and D ≠ {∅} then
    for all i in D do
      if i.CWND > 1 then
        decrement i.CWND;
        increment d.CWND;
        break;
      end if
    end for
  end if
end if
```

CWND Optimization

- Optimal performance (i.e. maximum throughput) can be linked to the size of each destination's CWND.

- Two optimization methods:
  - Dynamic Congestion Window Management (i.e., $policy_2$)
  - Static Congestion Window Management (ILP and heuristic)

CWND Optimization

- Static Congestion Window Management
  - Generate a set of CWND limits to maximize throughput during CMT.
  - Uses CMT performance model (i.e., Markov or renewal model).
- Integer Linear Program

$$
\begin{aligned}
\max \sum_{i}^{n} \eta(b_i, d_i, p_i, c_i^{\max}, t_i^{\max}) \\
\text{s.t.} \quad \sum_{i}^{n} c_i^{\max} \le r, \\
1 \le c_i^{\max} \le \lceil b_i \cdot d_i + 1 \rceil, \\
c_i^{\max} \in \mathbb{Z}, \\
\forall i \in N.
\end{aligned}
$$

$n \equiv$ number of paths

$b_i \equiv$ bandwidth of path i (packets per second)

$d_i \equiv$ delay on path i (seconds)

$c_i^{\max} \equiv$ CWND limit on path i (packets)

$t_i^{\max} \equiv$ maximum timeout on path i (seconds)

$r \equiv$ size of the receive buffer (packets)

CWND Optimization

- Heuristic
  - ILP can take a long time to converge.
  - Heuristic reduces the number of searches needed to find a solution.
  - Uses a subset of values when searching for the best set of CWND limits.

$$\left\{ kx : 1 \le x \le \left\lfloor \frac{a}{k} \right\rfloor, a = \min\left(r, \left\lceil b_i \cdot d_i + 1 \right\rceil\right), k \le a \right\}$$

$a \equiv$ total number of different CWND limits

$k \equiv$ decreases the search space

$b_i \equiv$ bandwidth on path i

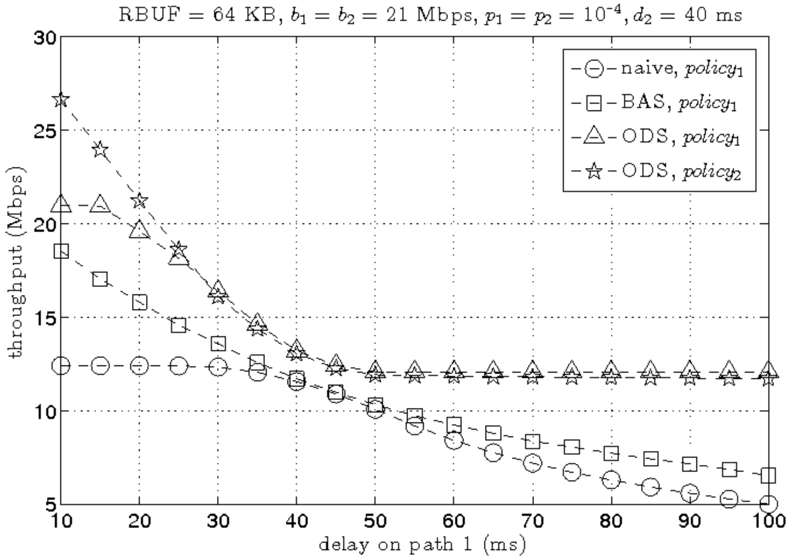
$d_i \equiv$ delay on path i

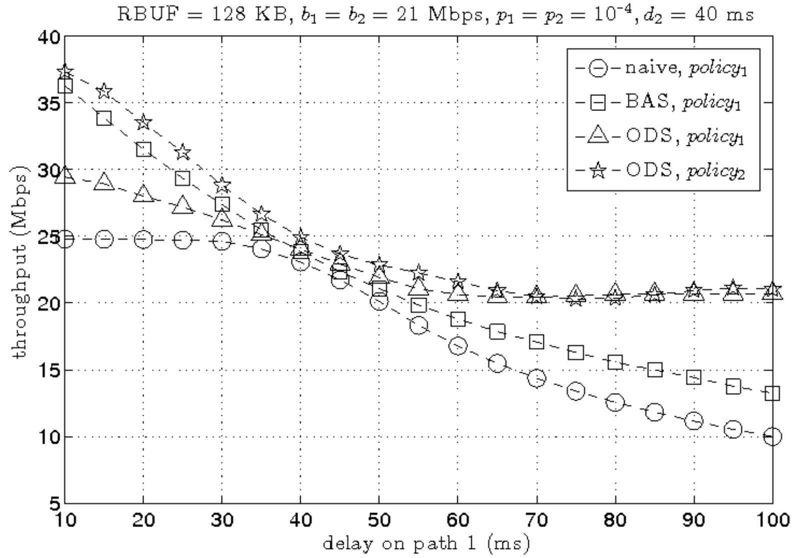
$r \equiv$ size of the receive buffer

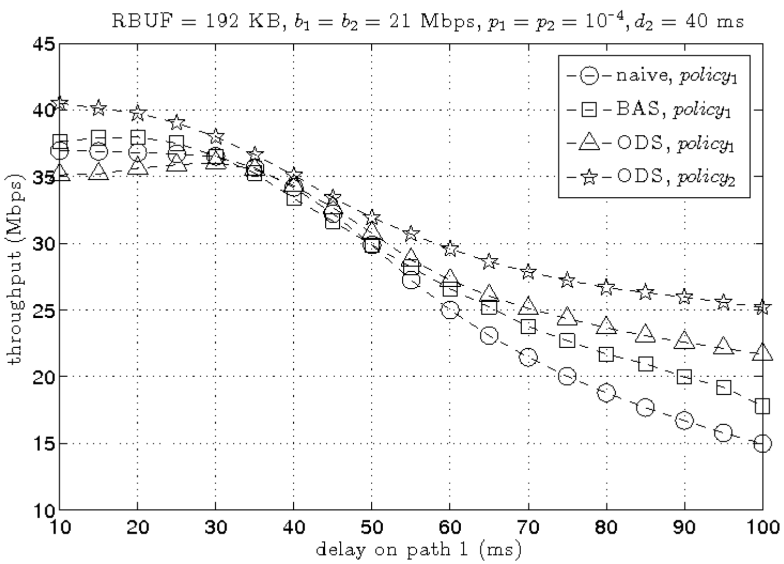
Performance Results

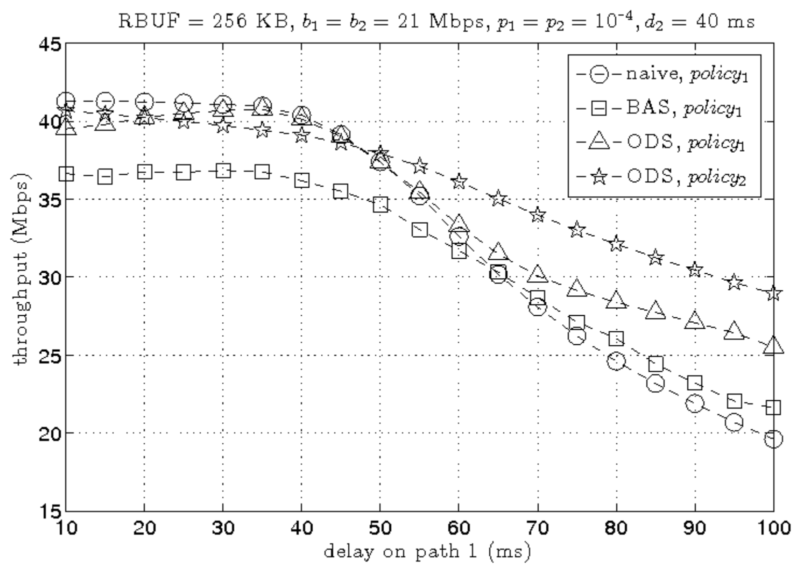
- CWND Update Policy
  - Revisit delay-based disparity and compare *policy*$_1$ *vs. policy*$_2$
- CWND Optimization
  - Dynamic vs. Static CWND Management
  - Heuristic

Performance Results

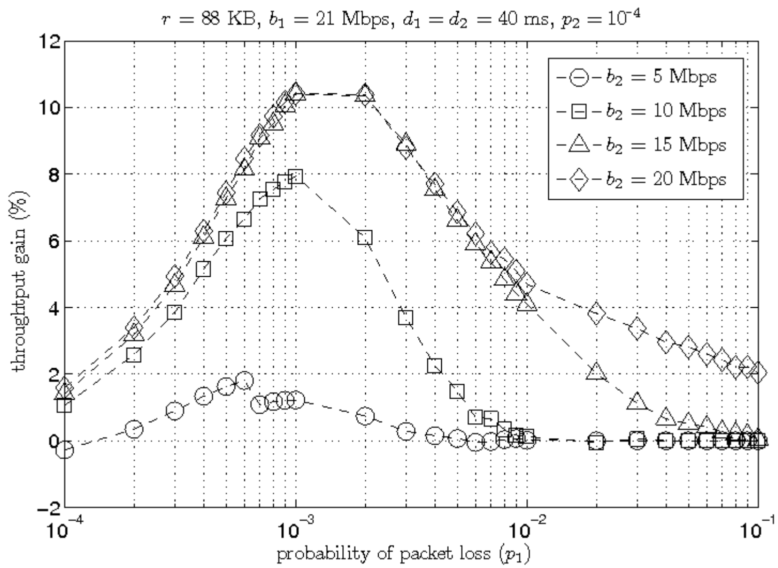


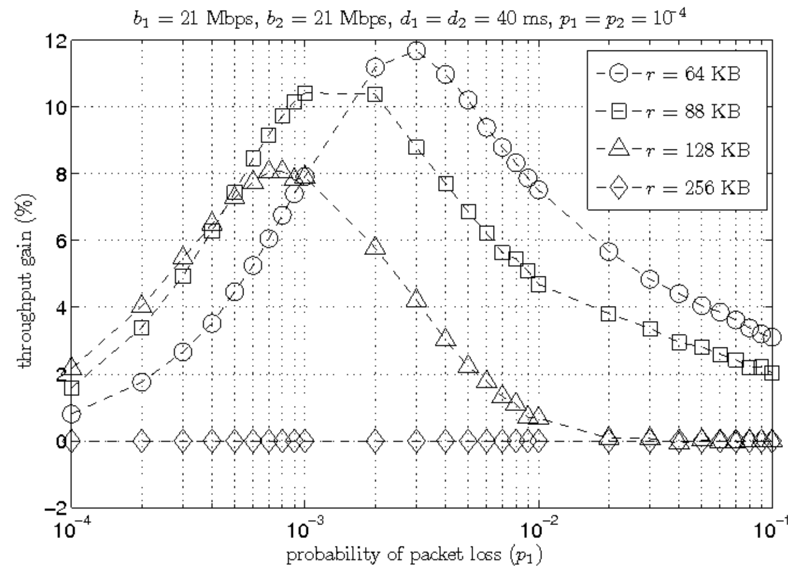


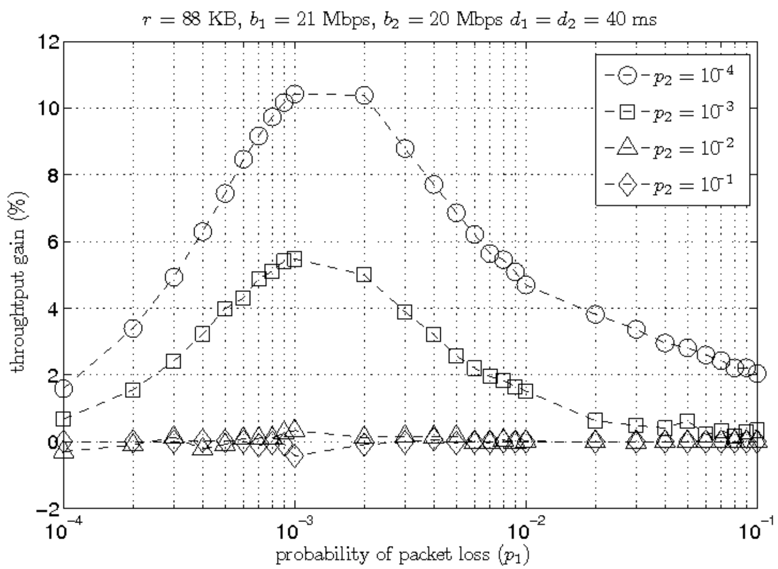


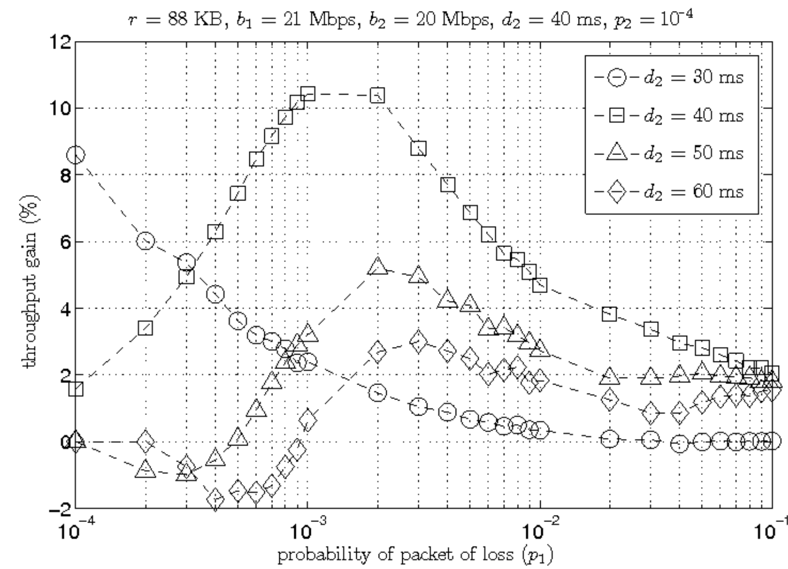


Performance Results



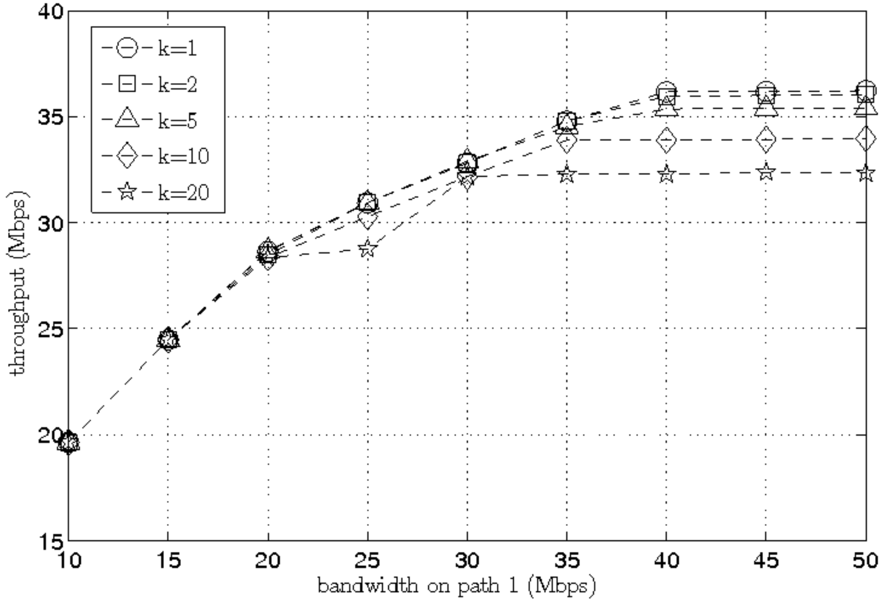


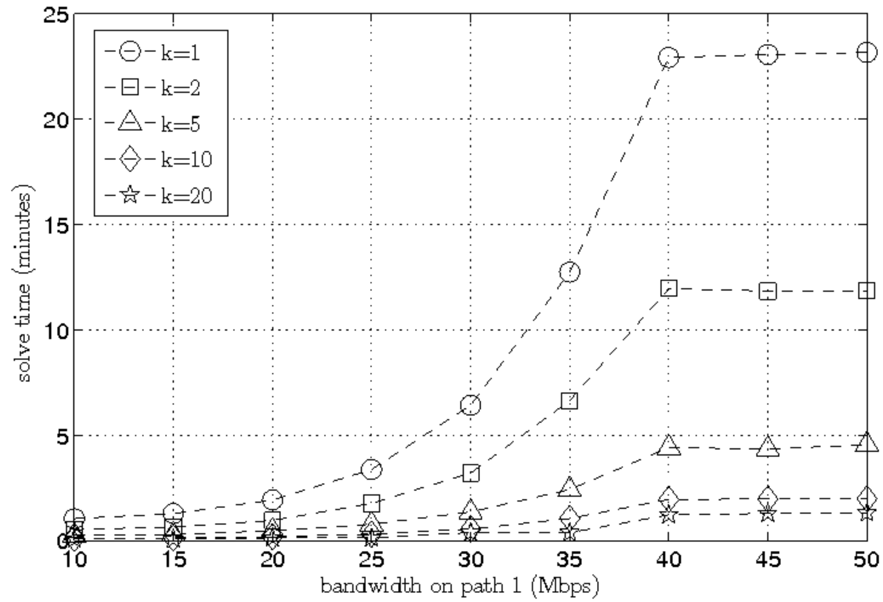




Performance Results

- Heuristic
  - Parameters: r = 128 KB, $p_1 = p_2 = 10^{-4}$, $d_1 = d_2$ = 40 ms, b_2 = 10 Mbps, k (variable), b_1 (variable).





Higher values of *k* yield higher throughput.

Lower values of *k* take less time to find a solution.

Summary

- Developed a new CWND update policy for CMT.
  - Compared *policy*$_1$ to *policy*$_2$ under delay-based disparity.
- Created an ILP to solve the static CWND management optimization problem.
  - Compared dynamic and static CWND management under different network scenarios.
  - Static CWND management yields better results but requires system knowledge (e.g, loss rate) and increases computational complexity.
- Reduced computational complexity by developing a simple heuristic.
  - Evaluated our heuristic using various subsets of CWND limits.
  - Using larger values of *k* lowers performance capabilities but also reduces computational requirements.

Open Challenges

Challenges

- CMT: Scheduling
  - Problem: ODS is a search algorithm that has some computational requirements.
  - Develop a closed-form expression that imitates ODS.
  - Implement ODS in the Linux kernel.

Open Challenges

- CMT: Modelling
  - Problem: perfect scheduling was assumed to avoid receive buffer blocking due to loss-based disparity.
  - Incorporate the effects of loss-based disparity into the model.

Open Challenges

- CMT: CWND Management
  - Problem: short term gains are not considered during the static optimization process.
  - Include the short term gains into the CMT model for static CWND management.
  - Develop a solution method using a metaheurstic (e.g., simulated annealing, tabu search, genetic algorithms).
  - Formulate an optimal decision policy using a Markov Decision Process (MDP).

Western

UNIVERSITY · CANADA