The Impact of Multihop Wireless Channel on TCP Performance

Zhenghua Fu, Haiyun Luo, Petros Zerfos, Lixia Zhang and Mario Gerla

Department of Computer Science, University of California at Los Angeles, Los Angeles

IEEE Transactions on Mobile Computing, 2005

Outline

- Introduction
- Link-layer Contention and Spatial Channel Reuse
- TCP Window Size and Throughput
- Packet Loss in Multihop Wireless Networks
- TCP Performance Improvement
  - Link RED
  - Adaptive Pacing
- Performance Evaluation
- Conclusions

Introduction (1/2)

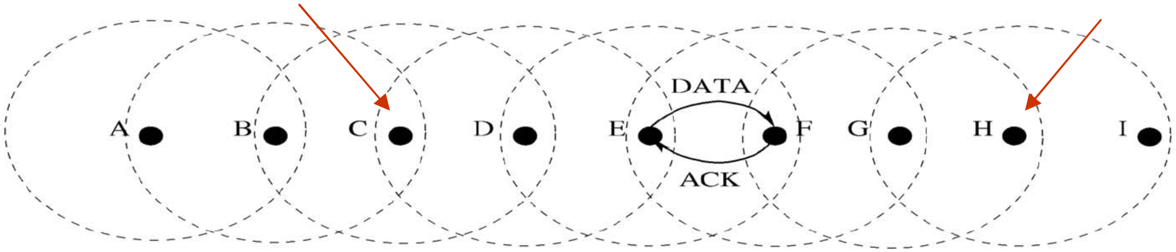
- Two unique char. of IEEE 802.11 multihop wireless networks may greatly affect TCP performance
  - Contention for the access to the shared wireless channel is *location-dependent* (Hidden/Exposed terminal problems).
  - Improving channel utilization through *spatial reuse.*
- Optimal window size W* *exists* at which TCP achieves the highest throughput via maximum spatial reuse.
- But, TCP grows its window size much larger than W*.

Introduction (2/2)

- Analysis of the packet loss reveals the reason for the TCP throughput decrease
  - Packet droppings due to link-layer contention offer the first sign of network overload.
  - The probability of packet dropping due to link contention increases as the offered load increases.
  - Saturates when every intermediate node along the forwarding path has a nonempty packet queue.

- Propose two link layer techniques to improve TCP throughput:
  - Link RED: finetune the wireless link's dropping probability to stabilize the TCP window size around W*
  - Adaptive pacing: better coordinate the spatial channel reuse

Link-layer Contention and Spatial Channel Reuse



Exposed terminal of transmission E->F

Hidden terminal of transmission E->F

- Two adjacent nodes are 200m apart.
- The transmission range of a node is 250m.
- The carrier sensing range is 550m.

- **For optimal spatial channel reuse**
  - {AE}, {BF}, {CG}, and {DH}) transmit alternatively

TCP Window Size and Throughput (1/6)

- TCP window size v.s throughput in multihop wireless networks using various configurations
  - chain, grid, cross and random network topologies

- **Chain Topology**
  - For an h-hop chain, the maximum number of simultaneous transmissions is upper bounded by h/4.
  - The pipe size over each hop is one packet (stop-and-wait)

    => The total pipe size over the entire packet forwarding is h/4.
  - TCP achieves the highest throughput with its window size being h/4.

TCP Window Size and Throughput (2/6)

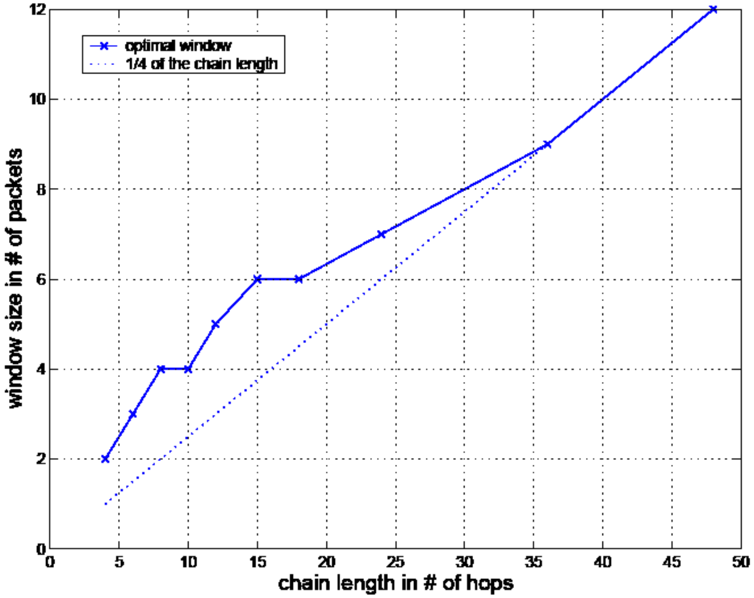


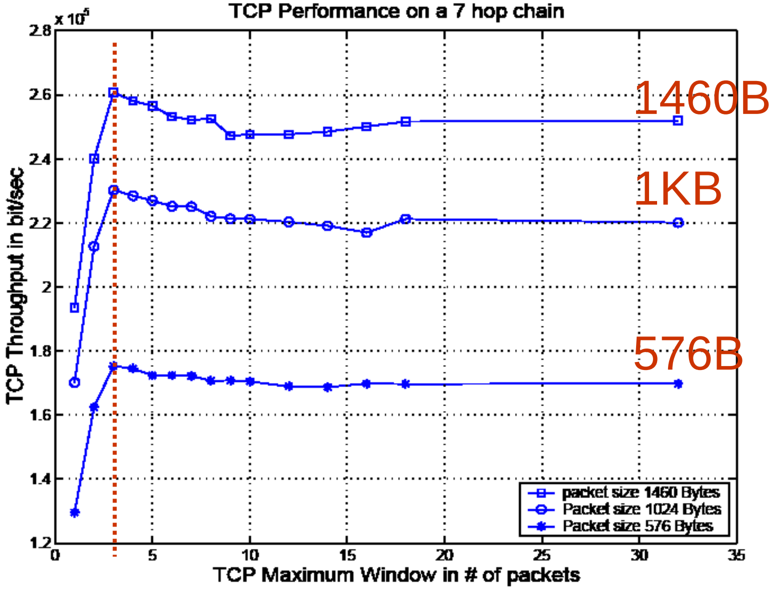
TABLE 1
Deviation of Queue Lengths

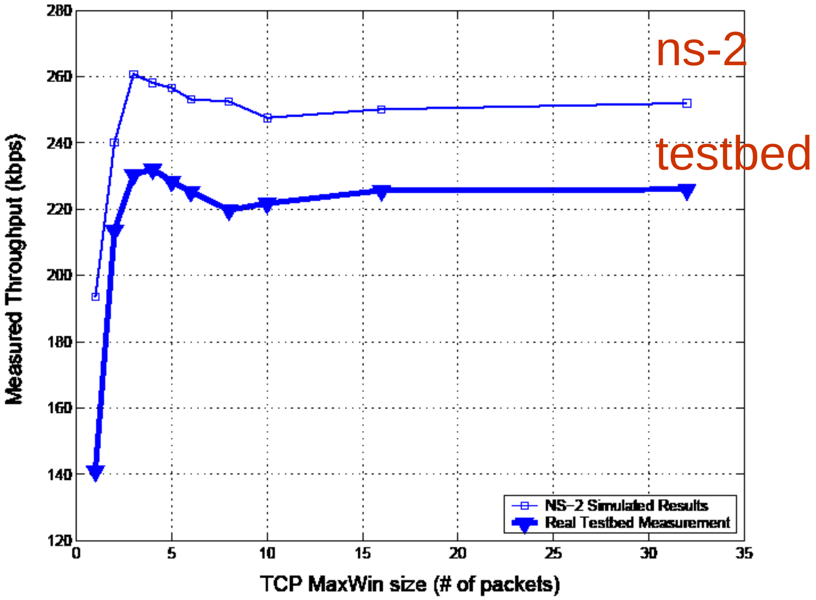
Chain length (hops)	4	7	10	16	48
Queue size deviation	1.45	1.31	1.23	1.10	1.05

Chain topologies of different lengths.

- Vary the MaxWin from 1 to 32 packets.
- Plot the MaxWin at which TCP achieves the maximum throughputs (W*)
- The figure show that W* and h/4 match reasonably well

TCP Window Size and Throughput (3/6)

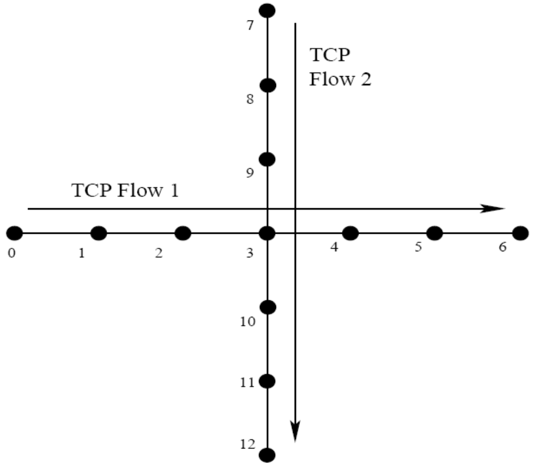




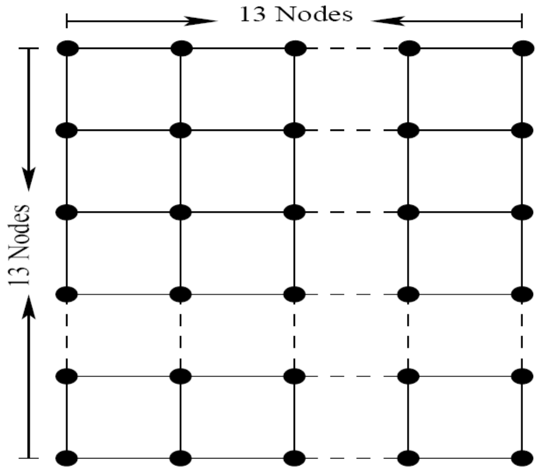
- W* is identical with different packet sizes of 576B, 1KB, and 1,460B.
- Comparisons between ns-2 simulations and testbed experiments

TCP Window Size and Throughput (4/6)

- **Cross topology**
  - W* for each flow is 2
  - Measured aggregate TCP window is 12 packets
  - 20% throughput decrease



- **Grid topology**
  - Run 4, 8, and 12 TCP flows
  - In all cases, the measure TCP window sizes are significantly larger than W*



TCP Window Size and Throughput (5/6)

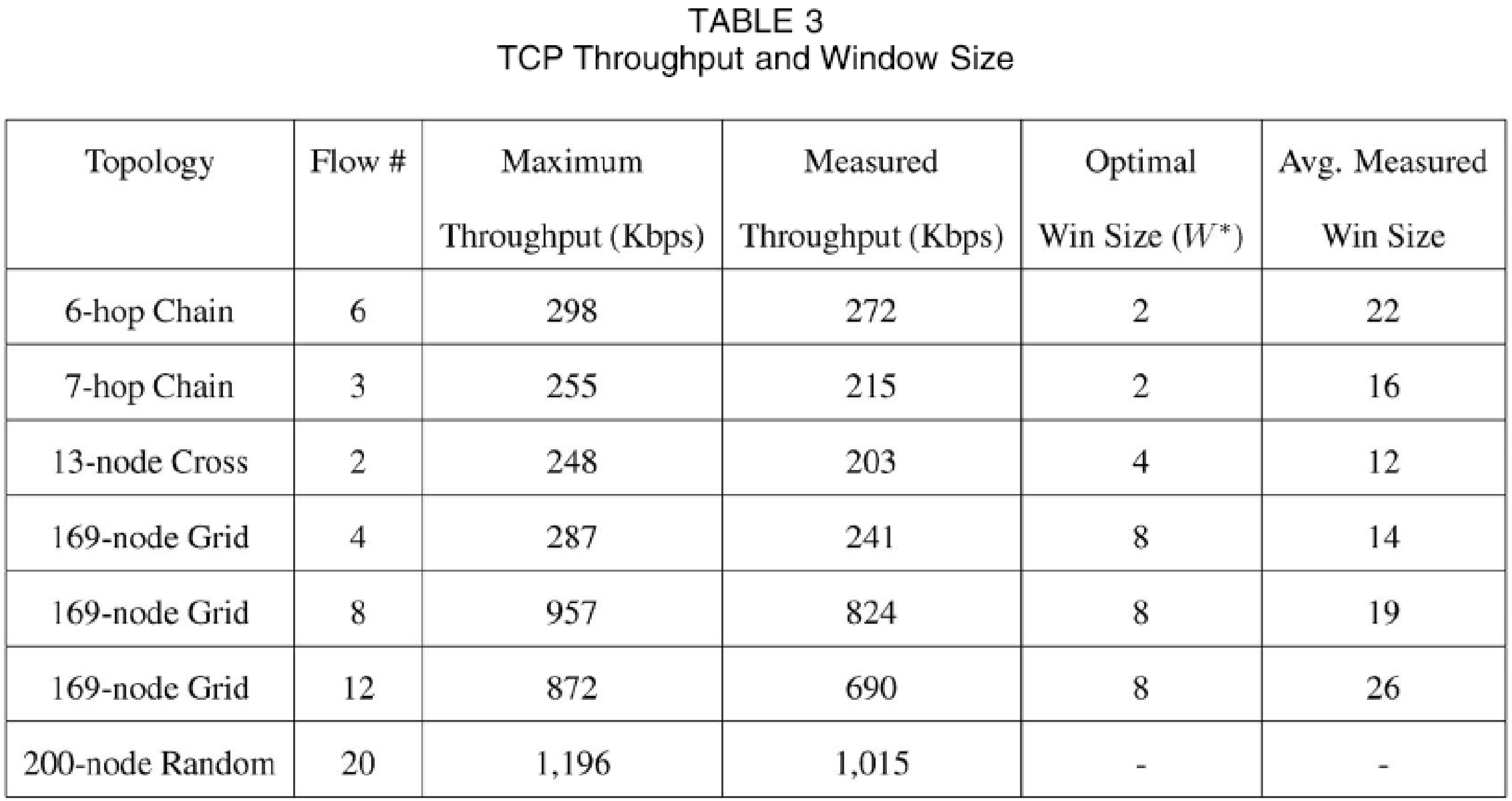


TABLE 3
TCP Throughput and Window Size

Topology	Flow #	Maximum Throughput (Kbps)	Measured Throughput (Kbps)	Optimal Win Size (W^*)	Avg. Measured Win Size
6-hop Chain	6	298	272	2	22
7-hop Chain	3	255	215	2	16
13-node Cross	2	248	203	4	12
169-node Grid	4	287	241	8	14
169-node Grid	8	957	824	8	19
169-node Grid	12	872	690	8	26
200-node Random	20	1,196	1,015	-	-

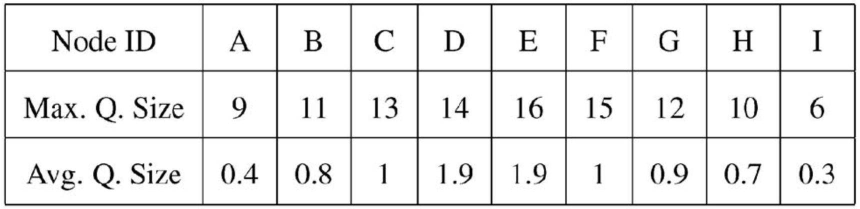
TCP Window Size and Throughput (6/6)

- **Summary**
  - For a given topology and traffic pattern, there exists a W* at which TCP achieves the highest throughput.
  - W* is a function of the number of hops the TCP flow traverses, independent of the bandwidth or delay at any intermediate node.
  - If we let MaxWin grow unbounded, an observation is that TCP throughput decreases by 4% to 21%.

Packet Loss in Multihop Wireless Networks (1/9)

- Packet Loss in Multihop Wireless Networks
  - Packet loss is dominated by link-layer contentions

TABLE 4
Queue Sizes in Packets



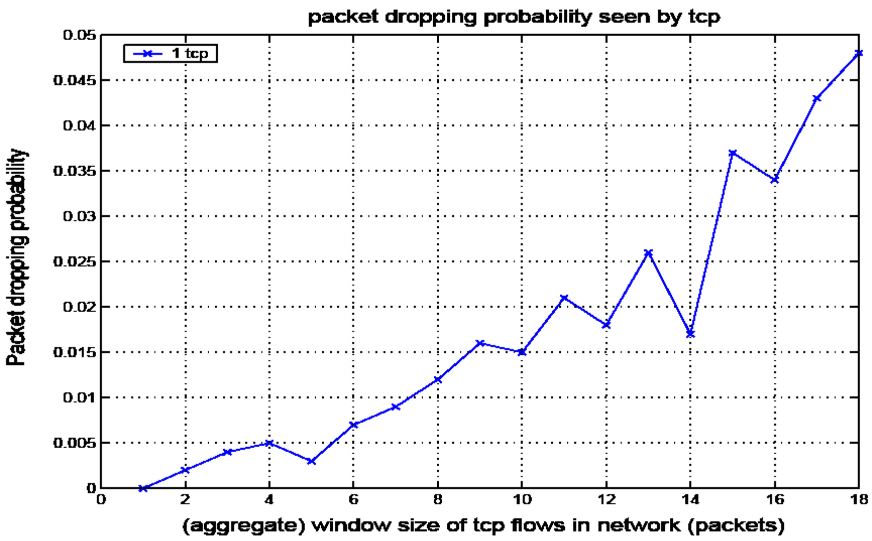
Node ID	A	B	C	D	E	F	G	H	I
Max. Q. Size	9	11	13	14	16	15	12	10	6
Avg. Q. Size	0.4	0.8	1	1.9	1.9	1	0.9	0.7	0.3

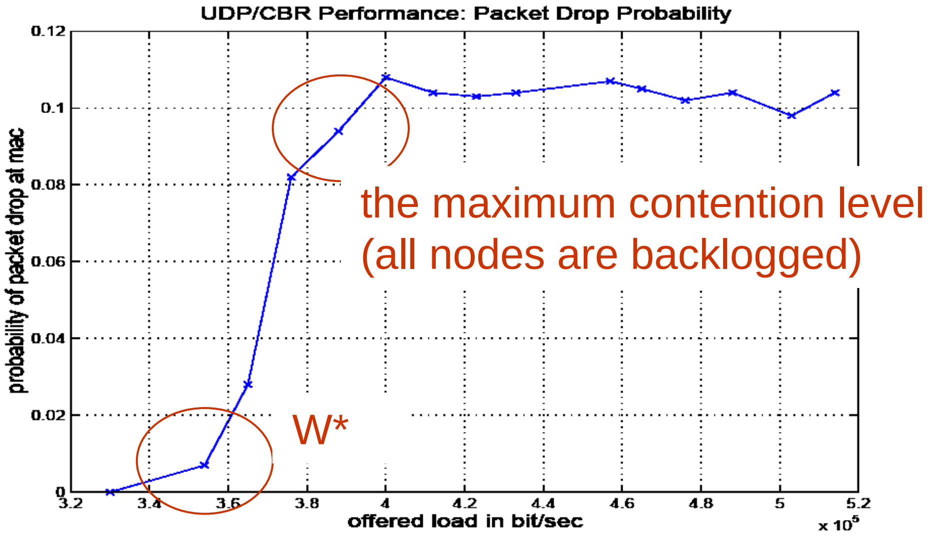
No packet drop due to buffer overflow.

  - TCP congestion control, designed to adapt to the packet loss due to buffer overflow, may not work well.

Packet Loss in Multihop Wireless Networks (2/9)

- Why TCP throughput decreases at CWND > W*?
  - The level of link-layer contention increases as the number of nodes that contend for the shared wireless channel increases.
    - A large number of nodes have backlogged queues.
    - The larger the TCP window size, the more packets in flight and the more nodes are backlogged.





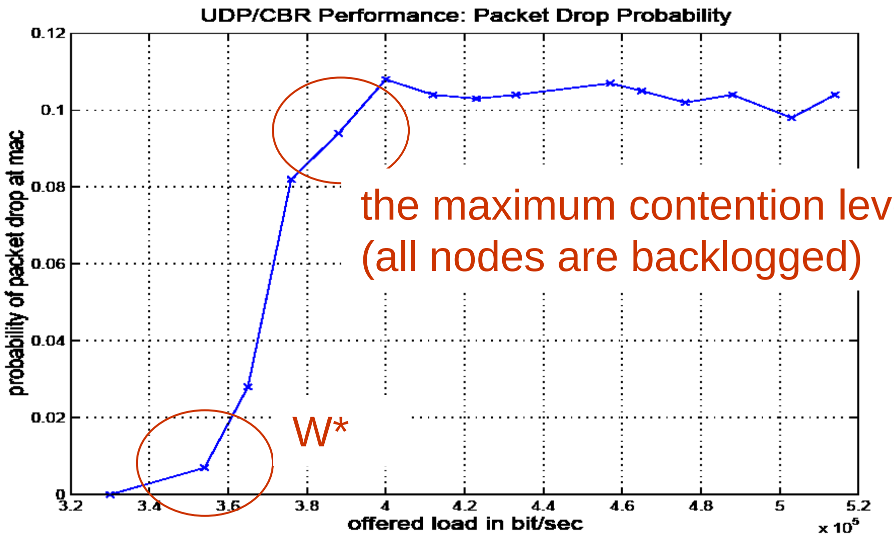
the maximum contention level (all nodes are backlogged)

W*

Packet Loss in Multihop Wireless Networks (3/9)

- TCP Window size cannot stay around W* since the packet dropping prob. is around zero.

- packet dropping prob. due to link-layer contention is insufficient to stabilize the TCP window size around optimal value.



the maximum contention level (all nodes are backlogged)

W*

Packet Loss in Multihop Wireless Networks (4/9)

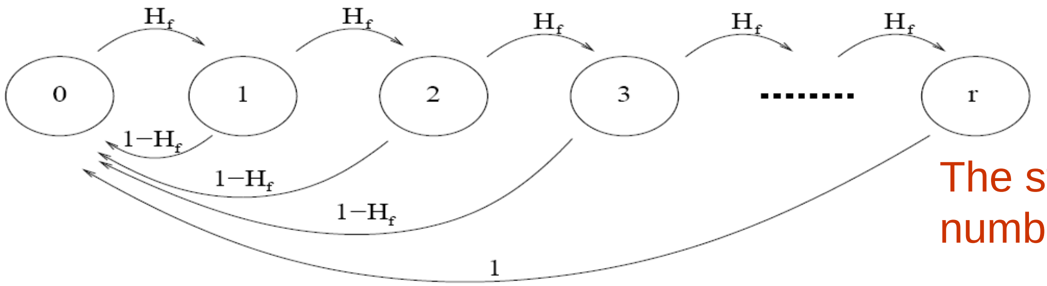
- Probability of Link-Layer Contention Induced Packet Drops
  - A Model for Hidden Terminal Effect
  - The probability that a node initiates RTS: CS_f
  - The probability that a subsequent successful DATA transfer: B_f
  - The probability that a flow f is hidden by some terminals: H_f
  - Therefore, we have

$$B_f = (1 - H_f) CS_f$$

$$H_f = 1 - \frac{B_f}{CS_f}$$

Packet Loss in Multihop Wireless Networks (5/9)

- Calculate the packet drop probability from the *hidden probability*
  - The packet is dropped after r unsuccessful RTS initiations.



The states represent the number of failed initiations

  - $p_r = \frac{1-H_f}{1-H_f^{r+1}} H_f^r.$ (based on the number of RTS initiations)
  - The average packet loss probability L_f for a given time slot is

$$\boxed{L_f = p_r \cdot CS_f = \frac{1-H_f}{1-H_f^{r+1}} H_f^r \cdot CS_f.}$$

per-flow packet drop probability.

Packet Loss in Multihop Wireless Networks (6/9)

- Loss probability for random topology (relate with the load)
  - Define the traffic load as number of backlogged nodes
  - m : number of backlogged senders
  - $C*$: maximum number of concurrent RTS initiations without collisions
  - $B*$: maximum number of concurrent successful DATA transmissions

$$c(m) = \left\lfloor m / \left\lceil \frac{m}{C*} \right\rceil \right\rfloor \qquad b(m) = \left\lfloor c(m) / \left\lceil \frac{c(m)}{B*} \right\rceil \right\rfloor$$

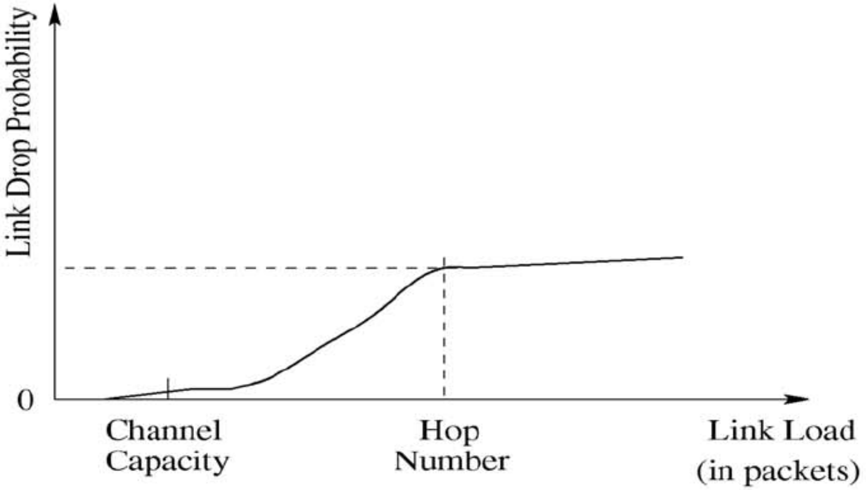
$$H_f(m) = 1 - \frac{b(m)}{c(m)}$$

$$L_f(m) = \frac{b(m)/m}{1-(1-(b(m)/c(m)))^{r+1}} \left(1 - \frac{b(m)}{c(m)}\right)^r$$

Packet Loss in Multihop Wireless Networks (7/9)

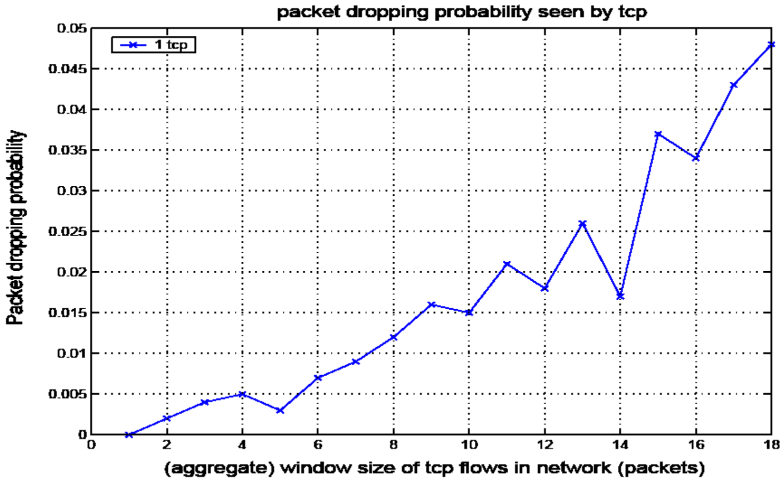
- Three regions of behavior
  - $m < B^*$ (underloaded)
    - $b(m) \approx c(m) \approx m$
    - $L_f(m) \approx 0$
  - $C^* > m > B^*$ (overloaded)
    - $b(m) \approx B^*,\ c(m) \approx m$
    - $L_f(m) \propto m$
  - $m > C^*$ (heavily loaded)
    - $b(m) \approx B^*,\ c(m) \approx C^*$
    - $L_f(m)$ saturated

$$L_f(m) = \frac{b(m)/m}{1-(1-(b(m)/c(m)))^{r+1}}\left(1-\frac{b(m)}{c(m)}\right)^r$$



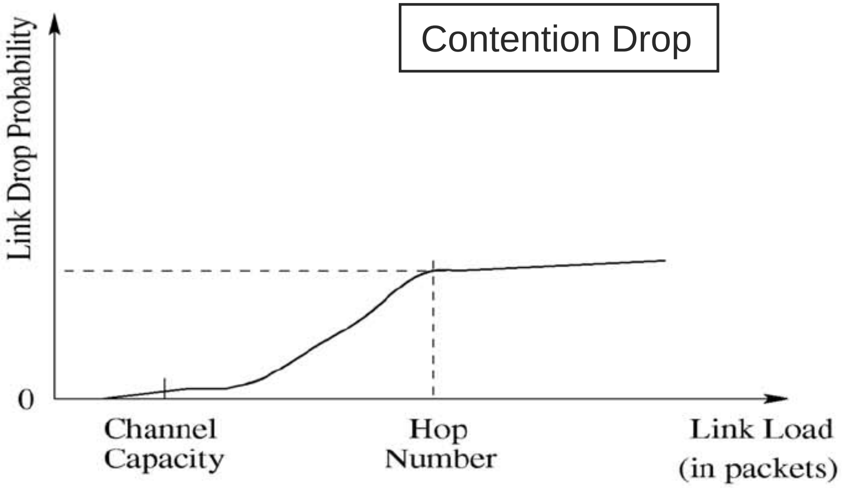
Packet Loss in Multihop Wireless Networks (8/9)

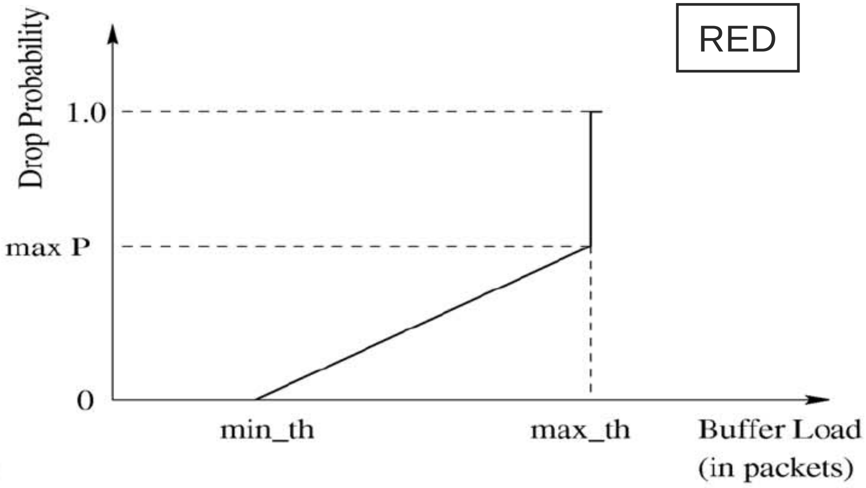
- **Discussions**
  - Why TCP Suffers from Throughput Decrease ?
    - When the window size grows beyond W*, the drop probability increase gradually until it stabilizes around a small value around 5%
    - The small drop prob. is insufficient to keep TCP around W*
    - The average window size is much larger than W*.



Packet Loss in Multihop Wireless Networks (9/9)

- Comparison to RED
  - Contention drop is a naturally built-in mechanism
  - Not useful for TCP in its current form *unless the loss/load curve is appropriately tuned*.
  - May happen before the network capacity is reached
  - RED drop probability reflects the local queue size, but contention drop probability reflects the global load level.





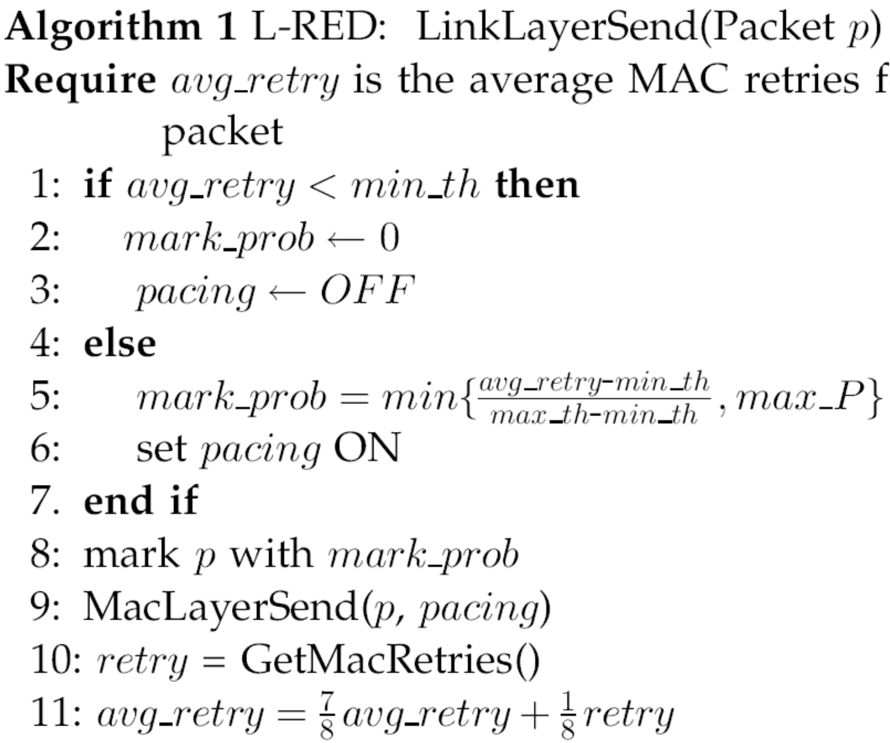
TCP Performance Improvement (1/3)

- Two link-layer designs to make contention drops beneficial to TCP flows.
    - Link RED
    - Adaptive Pacing

- **Link RED**
    - Control the TCP window size by tuning up the link-layer dropping probability according to perceived channel contentions.
    - LRED increases its packet dropping prob. linearly when the link-layer contention level exceeds a min_th. (Similar to the RED)
    - Maintain an average of the number of packet retransmissions as its contention level.

TCP Performance Improvement (2/3)

- **Link RED**
  - Maximum spatial channel reuse and minimum channel contention are achieved.

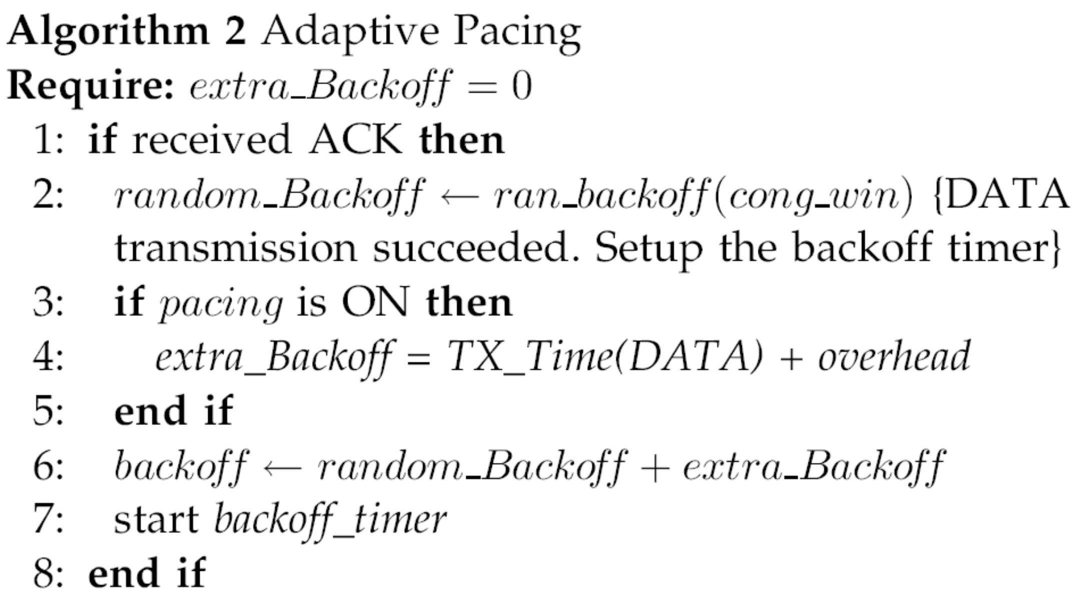


Algorithm 1 L-RED: LinkLayerSend(Packet p)

```
Require $avg\_retry$ is the average MAC retries for each packet
if $avg\_retry < min\_th$ then
    $mark\_prob \leftarrow 0$
    $pacing \leftarrow OFF$
else
    $mark\_prob = min\{\frac{avg\_retry - min\_th}{max\_th - min\_th}, max\_P\}$
    set $pacing$ ON
end if
mark $p$ with $mark\_prob$
MacLayerSend($p$, $pacing$)
$retry$ = GetMacRetries()
$avg\_retry = \frac{7}{8} avg\_retry + \frac{1}{8} retry$
```

TCP Performance Improvement (3/3)

- **Adaptive Pacing**
  - Balancing traffic among nodes can improve spatial channel reuse
  - Let a node backoff an additional packet transmission time if the traffic load is high
  - Enabled by LRED only when a node finds the average retransmission count be more than *min_th*.

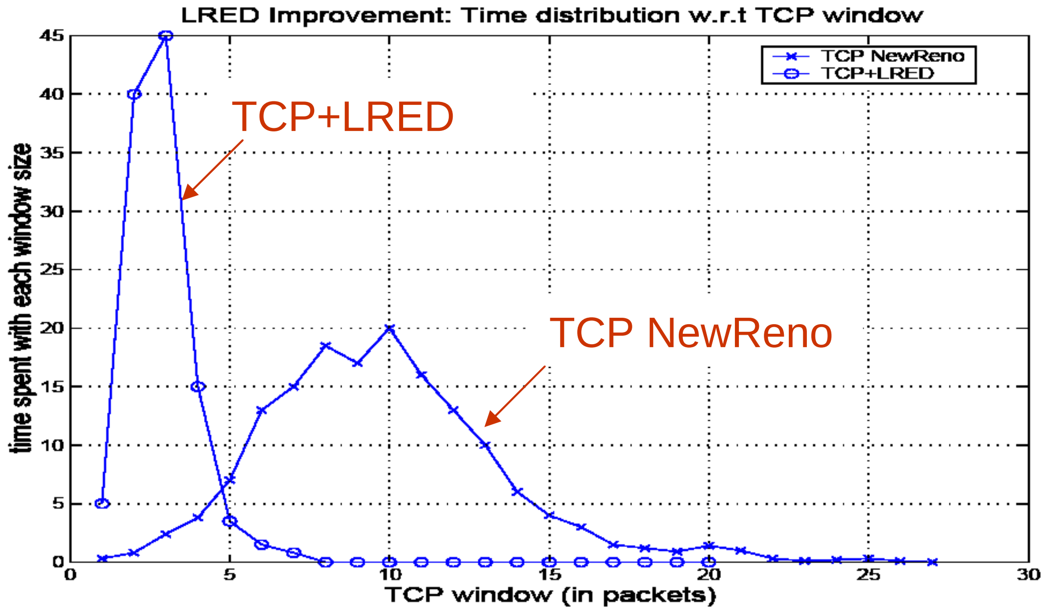


Algorithm 2 Adaptive Pacing

```
Require: $extra\_Backoff = 0$
if received ACK then
    $random\_Backoff \leftarrow ran\_backoff(cong\_win)$ {DATA transmission succeeded. Setup the backoff timer}
    if $pacing$ is ON then
        $extra\_Backoff = TX\_Time(DATA) + overhead$
    end if
    $backoff \leftarrow random\_Backoff + extra\_Backoff$
    start $backoff\_timer$
end if
```

Performance Evaluation (1/7)

- **LRED**
  - 7-hop chain topology
  - With LRED, spend most of the time with window size W* ≈3
  - The normal TCP grows its window much larger with an average size around 10 packets.



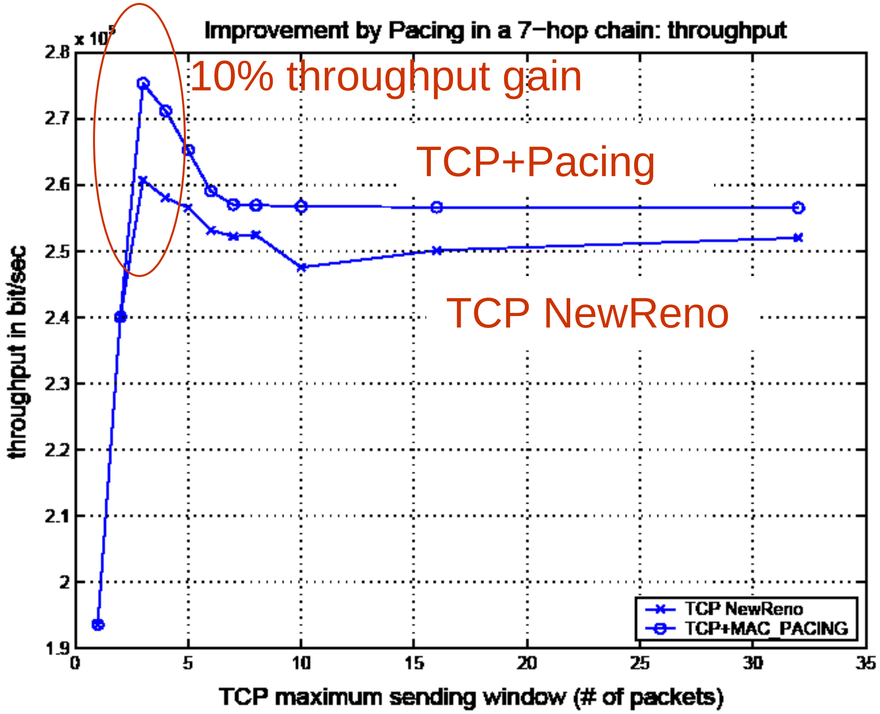
TCP+LRED

TCP NewReno

Performance Evaluation (2/7)

- **Adaptive Pacing**
  - 7-hop chain topology
  - Evaluate in terms of
    - Throughput gain
    - Link-layer contention induced packet drops
    - TCP RTT

Throughput gain



10% throughput gain

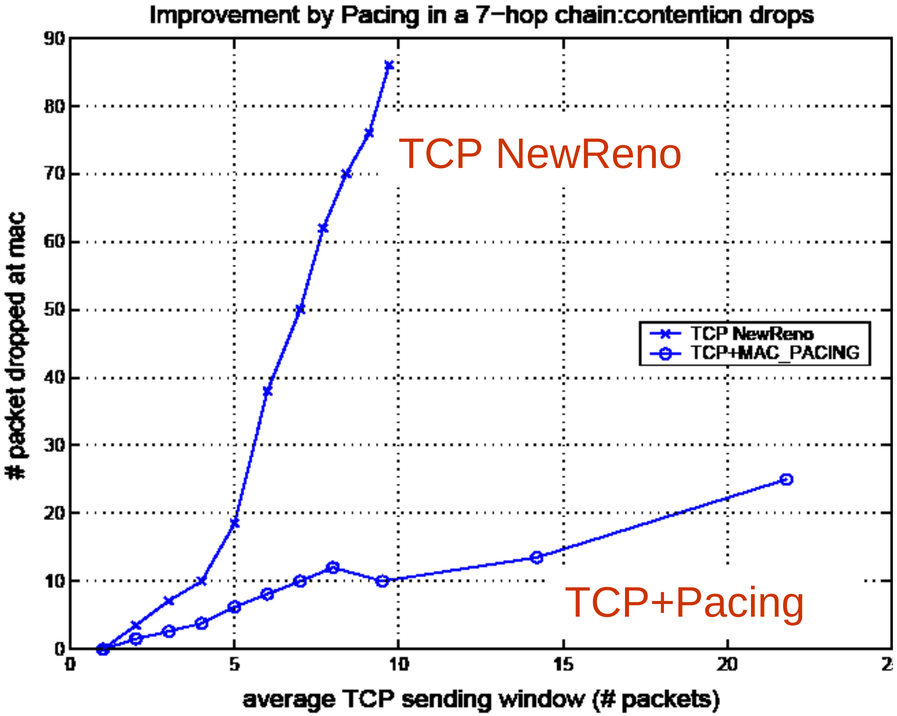
TCP+Pacing

TCP NewReno

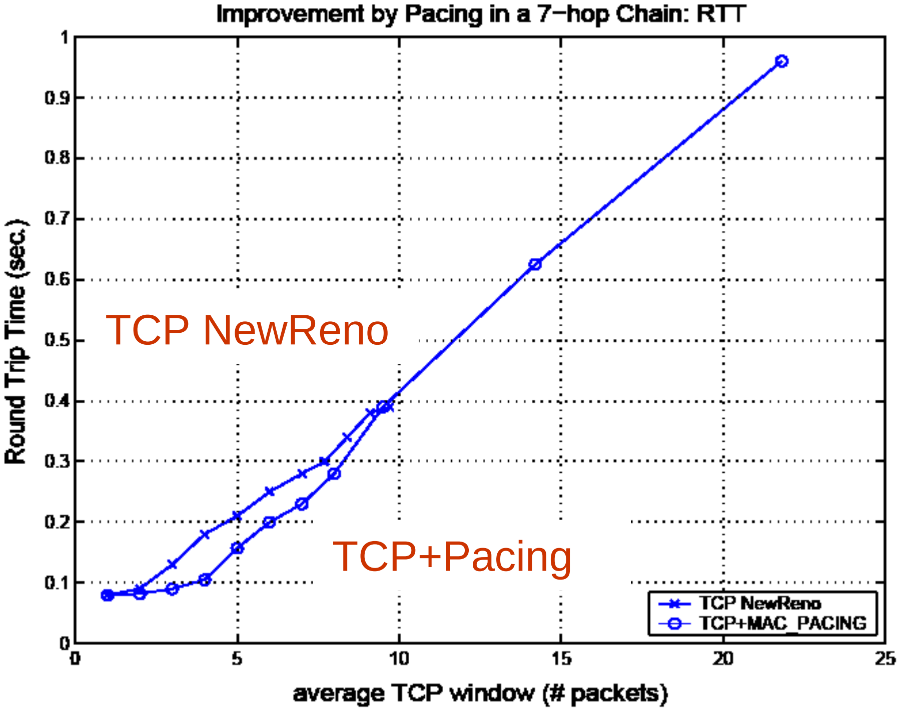
Performance Evaluation (3/7)

- **Adaptive Pacing**
  - Pacing has significantly reduced packet drops due to contention and also slightly reduces RTT

Contention drops



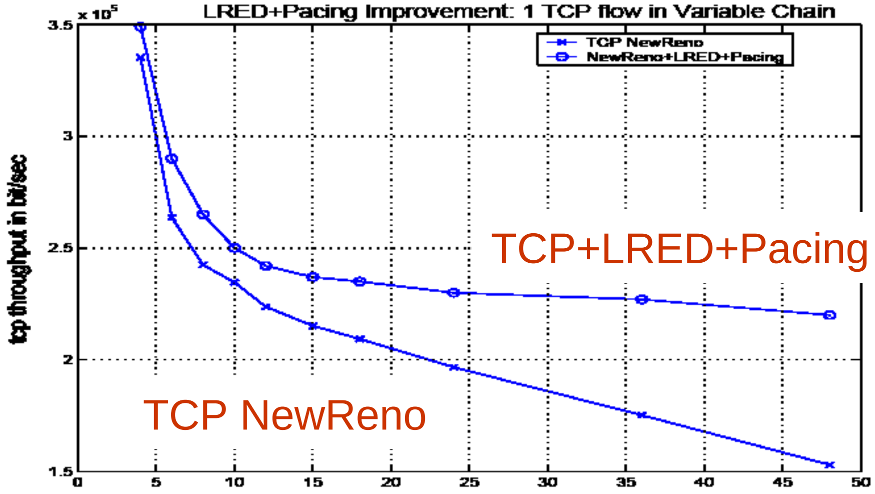
RTT



Performance Evaluation (4/7)

- **LRED + Adaptive Pacing**
  - *Chain topology*

1 flow

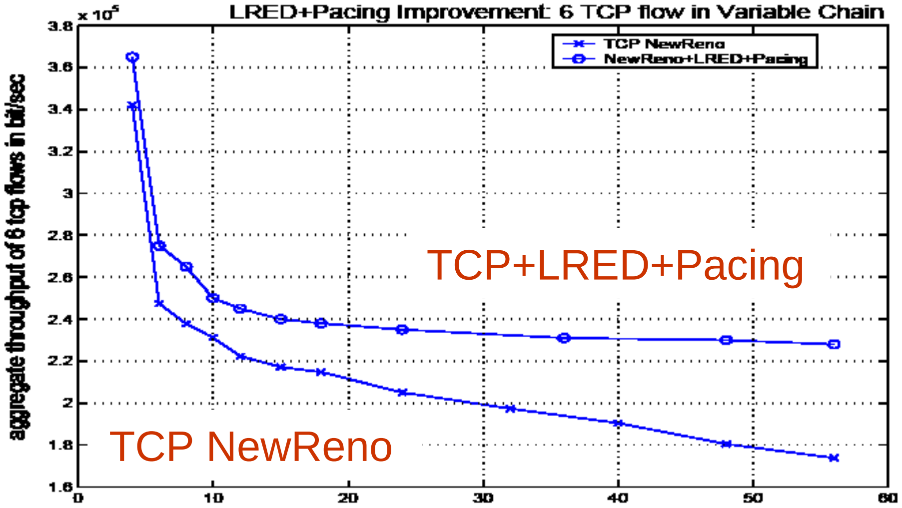


TCP+LRED+Pacing

TCP NewReno

chain length in # of hops

6 flows



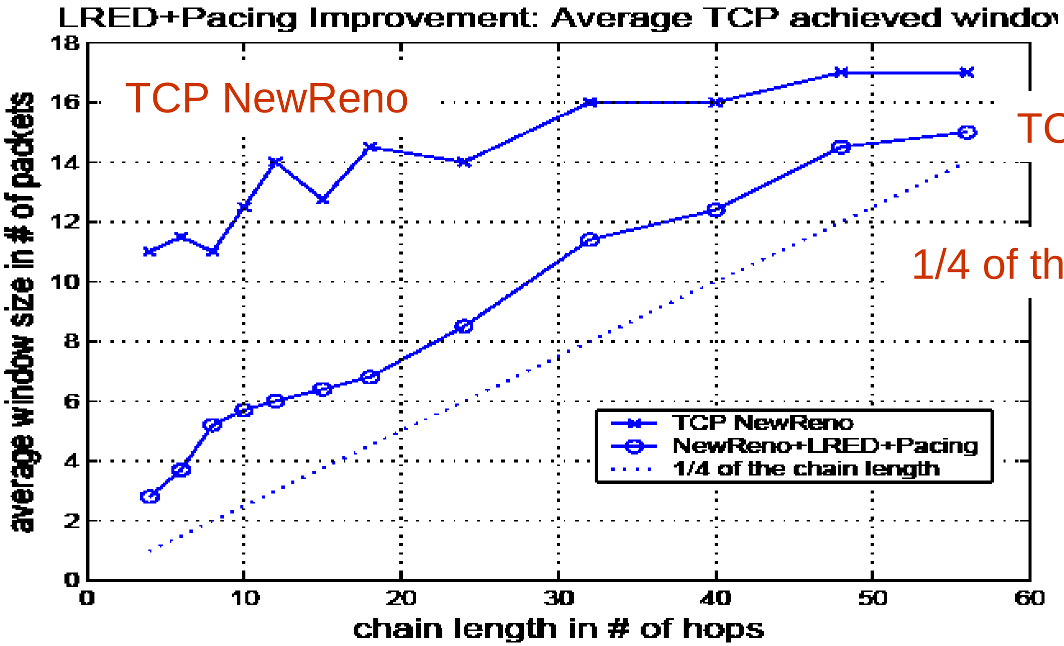
TCP+LRED+Pacing

TCP NewReno

chain length in # of hops

Performance Evaluation (5/7)

- **LRED + Adaptive Pacing**
    - *Chain topology*



TCP NewReno

TCP+LRED+Pacing

1/4 of the chain length

Performance Evaluation (6/7)

- **LRED + Adaptive Pacing**
  - *13-node cross topology and run two TCP flows*

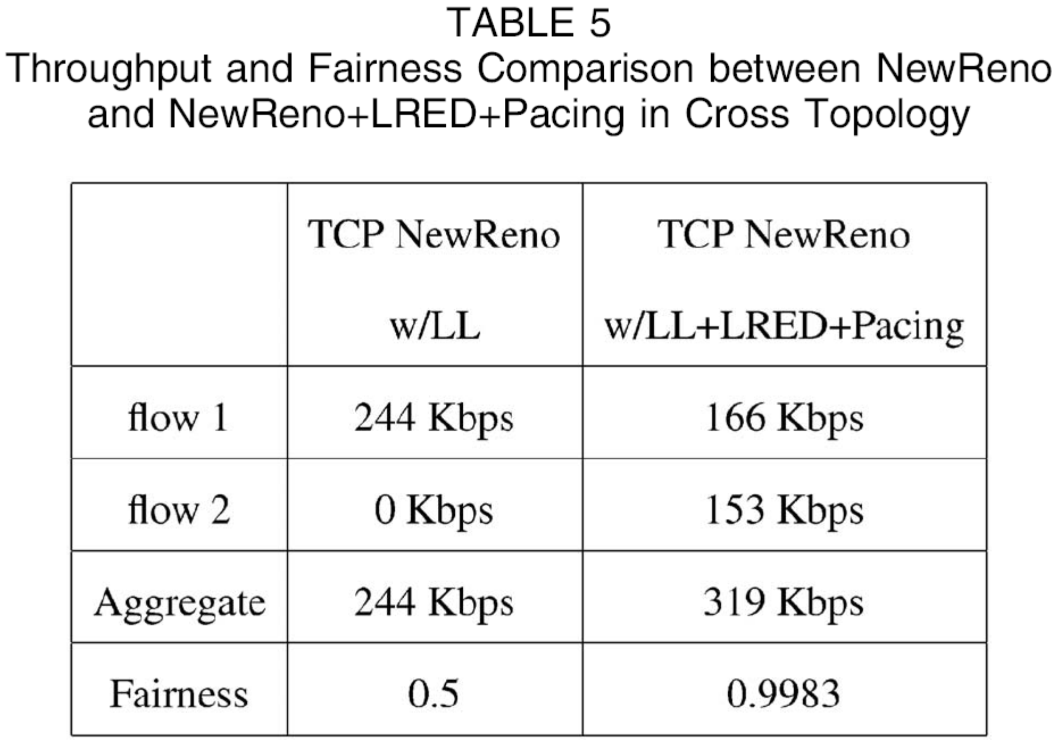


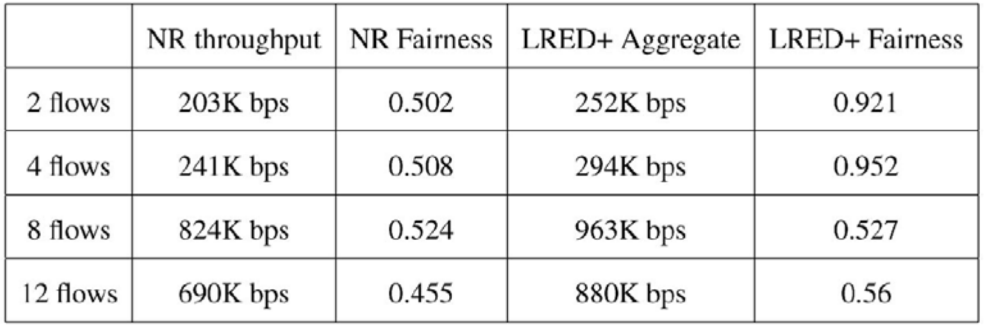
TABLE 5
Throughput and Fairness Comparison between NewReno
and NewReno+LRED+Pacing in Cross Topology

	TCP NewReno w/LL	TCP NewReno w/LL+LRED+Pacing
flow 1	244 Kbps	166 Kbps
flow 2	0 Kbps	153 Kbps
Aggregate	244 Kbps	319 Kbps
Fairness	0.5	0.9983

Performance Evaluation (7/7)

- **LRED + Adaptive Pacing**
  - *Grid topology*

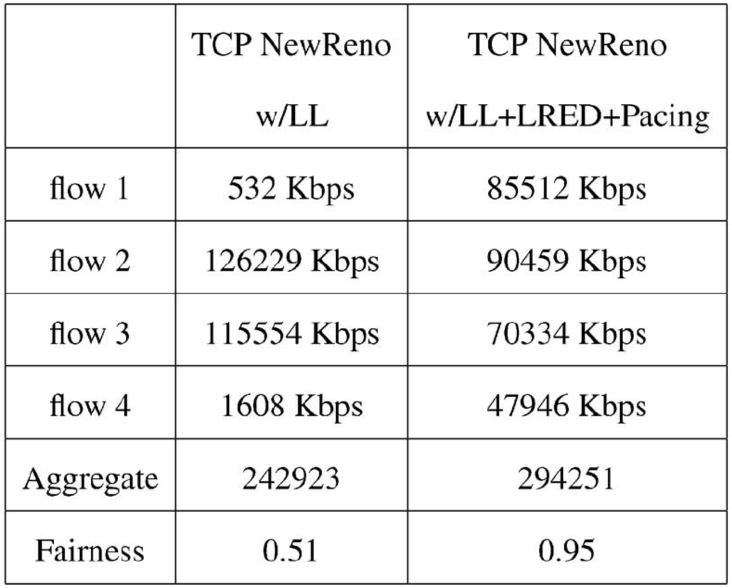
TABLE 6
Aggregate Throughput and Fairness Comparison between NewReno (NR) and NewReno+LRED+Pacing (LRED+)



	NR throughput	NR Fairness	LRED+ Aggregate	LRED+ Fairness
2 flows	203K bps	0.502	252K bps	0.921
4 flows	241K bps	0.508	294K bps	0.952
8 flows	824K bps	0.524	963K bps	0.527
12 flows	690K bps	0.455	880K bps	0.56

Grid topology with 2, 4, 8, and 12 flows

TABLE 7
Throughput and Fairness Comparisons between NewReno and NewReno+LRED+Pacing



	TCP NewReno w/LL	TCP NewReno w/LL+LRED+Pacing
flow 1	532 Kbps	85512 Kbps
flow 2	126229 Kbps	90459 Kbps
flow 3	115554 Kbps	70334 Kbps
flow 4	1608 Kbps	47946 Kbps
Aggregate	242923	294251
Fairness	0.51	0.95

Details for the cases of four flows

Conclusions

- Spatial channel reuse can improve channel utilization.
- A TCP window size W* (h/4) exists at which throughput is maximized by achieving best spatial reuse
- Standard TCP typically grows its average window much larger than W* in IEEE 802.11 networks
- Link layer techniques to improve TCP throughput
  - LRED
    - Tune the wireless link's drop probability to maintain CWND near W*
  - Adaptive Pacing
    - Increase the spatial reuse of the channel