TELE303 Mobile Systems

Lecture 8 – TCP in MANETs

Jeremiah Deng
TELE Programme / InfoSci
University of Otago, 2016

Outline

- Cross-layer design
  - Why?
  - Examples
- TCP performance
  - Review
  - TCP on MANETs
  - Design issues

Cross-Layer Design

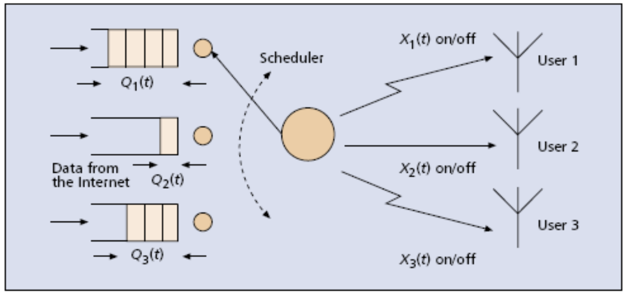
Challenges

- Wireless channels are a difficult and capacity-limited broadcast communications medium.
- Traffic patterns, user locations, and network conditions are constantly changing.
- Applications are heterogeneous with hard constraints that must be met by the network.
- Energy and delay constraints change fundamental design principles.

Cross-Layer Design

- Cross-Layer Design is a way of achieving information sharing between all the layers in order to obtain highest possible adaptability of any network.
- Co-operation between multiple layers to combine the resources and create a network that is highly adaptive

Multi-User Scheduling



- Shakkottai & Rappaport (2002)
- Scenario: base station schedules downlink traffic to multiple mobile users; channels are not stable and have equal chances to be on/off

Naïve Scheduling

- Round-robin
- Each user gets 1/3 of the slots
- On an average, each user gets a data rate of 1/6 packet/slots

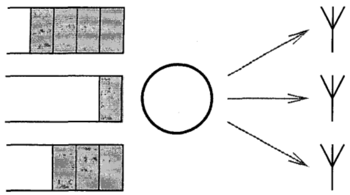
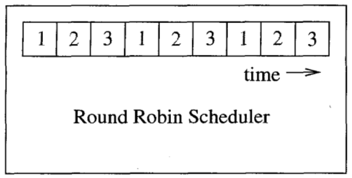


Fig. 2. Time-division scheduling: Queues at the base-station.



Smart Scheduling

- Suppose base station has knowledge about the channel status
- Try a simple policy: only send randomly among the 'on' channels
- What is the data rate for each user?
  - Chance of no sending: $(1/2)^3=1/8$
  - Total data rate: 1-(1/8)=7/8 packet/slots
  - Data rate for each user is therefore 7/24 packet/slots

Cross-Layer Design: A New Paradigm

- Application
- Transport
- Network
  - Access
- Link
- Physical

Diversity
Adaptability
Scheduling
End-to-End Metrics

Substantial gains in throughput, efficiency, and QoS can be achieved with cross-layer adaptation

TCP - Review

...

Transmission Control Protocol (TCP)

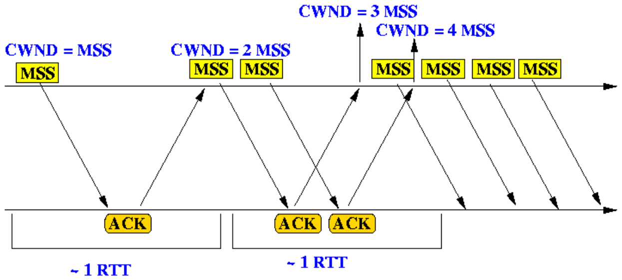
- End-to-end semantics
- Cumulative acknowledgements
- ACKs acknowledge all contiguously received data
  - Acknowledgements sent to TCP sender confirm delivery of data received by TCP receiver
- Reliable ordered delivery
- Reliability achieved by means of retransmissions if necessary
- Implements congestion avoidance and control

Performance Metrics

- Delay (latency)
- Delay variation (or jitter)
- Packet loss rate
- Throughput/Bandwidth: Application bytes/bits transferred per second.
- Throughput variation
- Fairness: Long-term/short-term fairness among all TCP/UDP flows.
- Resource consumption: Amount of resource consumed, e.g. CPU cycles, memory usage, battery, etc.

TCP Slow Start

- Congestion window (CWND) = the number of packets the sender is allowed to send
  - The larger the window size, the higher the throughput.
- Initial value is one segment.
- Slow start: exponential increase of cwnd
  - Doubling cwnd every RTT as successful ACKs are received.
  - Until the slow start threshold (ssthresh) is reached.
  - Then linearly increase cwnd.



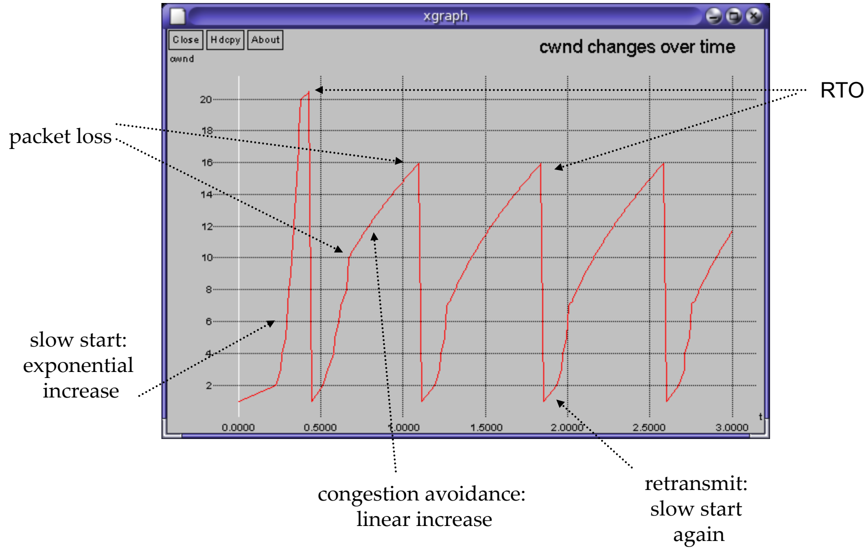
Congestion Avoidance and Control

- When cwnd reaches slow-start threshold, congestion avoidance is performed
- Congestion avoidance: cwnd increases linearly with time during congestion avoidance
  - Rate of increase could be lower if sender does not always have data to send
- Congestion control: upon detecting packet loss, TCP sender assumes network congestion and reduces cwnd
  - This also reduces network throughput.

Retransmission Timeout (RTO)

- On a timeout, cwnd is reduced to the initial value of one segment
- The slow start threshold is set to half the window size before packet loss
  - more precisely,
    - ssthresh = max(min(cwnd,adwin)/2, 2) MSS
    - adwin: receiver's advertised window
- Slow start is initiated
- RTO is costly.

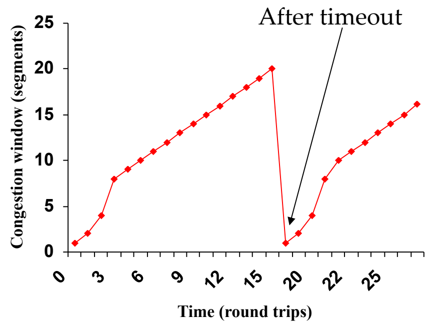
Example Trace from ns-2

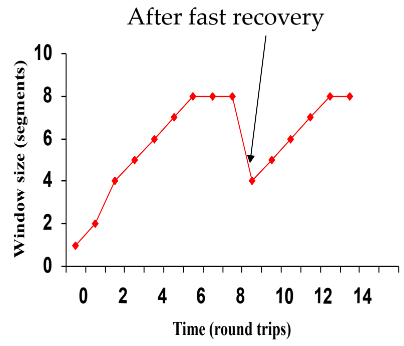


Fast Retransmit

- Fast Retransmit: retransmit after 3 duplicate acks (got 3 additional packets without getting the one you are waiting for)
  - Prevents expensive timeouts.
  - Fast recovery: cwnd back to ssthresh; no need to go into "slow start" again
- At steady state, CWND oscillates around the optimal window size
- With a retransmission timeout, slow start is triggered again.

RTO vs Fast-retransmit



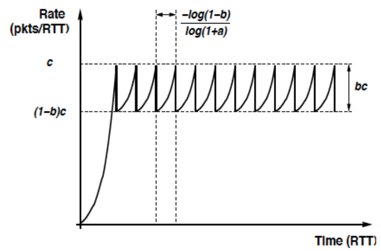


TCP Implementations

- TCP Reno
  - Fast retransmit of the missing data packet
  - Fast recovery
- TCP NewReno
  - Multiple packet losses in a window of data
- SACK (Selective ACK)
- More recently: FAST TCP, TCPW, STCP, BIC ...

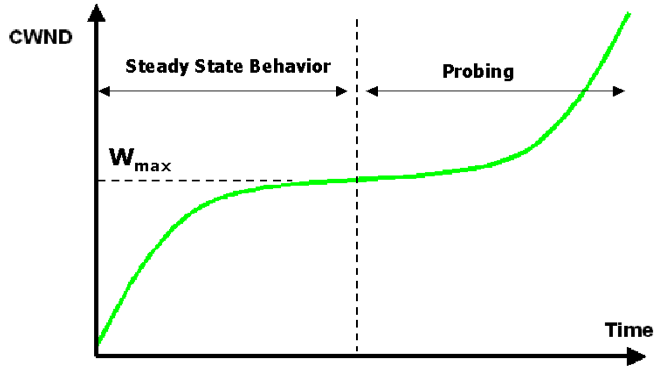
A New cwnd Rule

No congestion: cwnd=cwnd+0.01
1st congestion: cwnd=cwnd-0.125*cwnd



Scalable TCP
Source: http://datatag.web.cern.ch/datatag/papers/pfldnet2003-ctk.pdf

A New Growth Curve?



The new BIC aims at better stability, friendliness, and scalability.
Source: http://research.csc.ncsu.edu/netsrv/?q=content/bic-and-cubic

TCP in MANETs

• • •

TCP Performance in MANETs

- Several factors affect TCP performance in MANET:
  - Wireless transmission errors
  - Data and ACK packet collision
  - Multi-hop routes on shared wireless medium
    - For instance, adjacent hops typically cannot transmit simultaneously
    - Contention of MAC signals causes routing delay, which may then trigger RTO.
  - Route failures due to mobility

Fast Retransmit?

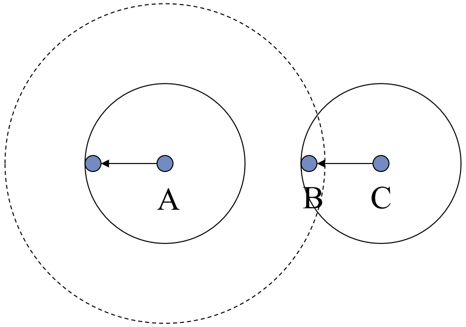
- Random errors may cause Fast Retransmit
- Fast retransmit results in
  - retransmission of lost packet
  - reduction in congestion window
- Reduction in congestion window reduces the throughput
- ☹ Reducing congestion window in response to errors is unnecessary

Good Congestion Response?

- Sometimes congestion response may be appropriate in response to errors.
- On a CDMA channel, errors occur due to
  - Interference from other user, and
  - Noise.
- Interference due to other users is an indication of congestion. If such interference causes transmission errors, it is appropriate to reduce congestion window
- If noise causes errors, it is **not** appropriate to reduce window.

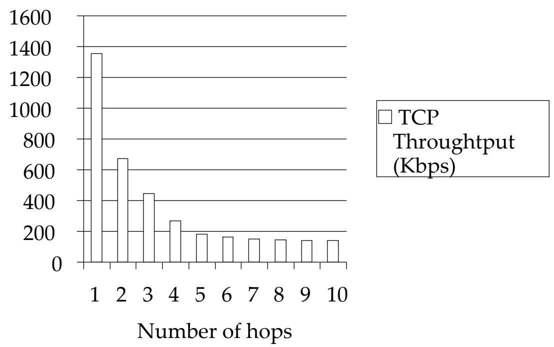
Exposed Terminal Problem

- Hidden terminal is not the only challenge for a distributed wireless MAC protocol
- A blocks B, and C doesn't know what is happening (B is exposed)



Throughput over Multi-Hop

- Connections over multiple hops are at a disadvantage compared to shorter connections
  - Have to contend for wireless access at each hop

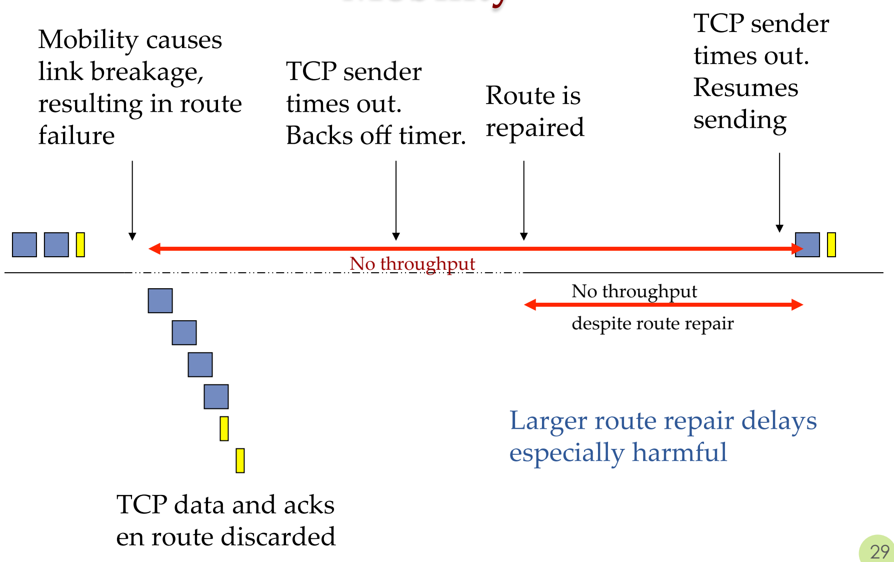


TCP Throughput using 2 Mbps 802.11 MAC

Increasing Number of Hops

- Packet transmission can occur on at most one hop among three consecutive hops
  - Increasing the number of hops from 1 to 2, 3 results in increased delay, and decreased throughput
- Increasing number of hops beyond 3 allows simultaneous transmissions on more than one link
  - However, degradation continues due to contention between TCP Data and ACKs traveling in opposite directions

Throughput Generally Degrades with Mobility



Multiple Factors …

- Multiple factors may affect the TCP performance of MANETs
- Besides mobility, interference and MAC contention, what's more:
  - 802.11 MAC is invisible to TCP/UDP
  - Presents itself as a reliable but time-varying channel
  - Time-varying behaviour can be mistaken as 'congestion', leading towards reduced throughput

Recap

- Cross-layer design
- Possible factors that affect the TCP performance of MANETs
- Possible solutions?
  - Reading / References

- WED/THU: Lab 3 w/ ns-2

References

- S. Shakkottai et al., "Cross-Layer Design for Wireless Networks", *IEEE Communications Magazine*, Vol. 41, No. 10, Oct. 2003, pp. 74-80.
- Koutsonikolas et al., "On TCP throughput and window size in a multihop wireless network testbed", http://dx.doi.org/10.1145/1287767.1287778
- Tannenbaum, Chapter 6, *Computer Networks*, 4th Ed., 2003.