CMPE 257: Wireless and Mobile Networking

Katia Obraczka
Computer Engineering
UCSC Baskin Engineering

Lecture 11
Student Presentations Schedule

- May 21: Sam and Anuj
- May 26: Larissa and Armando
- May 28: Daniel and Alan
- June 2: Ben and Maziar
- June 4:

  - Larissa: user mobility and modeling (May 26)
  - Daniel: location management and geocasting (May 28)
  - Ben: secure routing in MANETs (June 2)
  - Armando: game theory in MANETs (May 26)
  - Alan: SDN (May 28)
  - Sam: propagation models (May 21)
  - Anuj: IoT (May 26)
  - Maziar: ICN (June 2)
Reliable Point-to-Point Transport: Outline

✓ TCP/IP basics.
✓ Impact of transmission errors on TCP performance.
✓ Approaches to improve TCP performance on wireless networks.
  ✓ Classification.
  • TCP on cellular (last-hop wireless).
  • TCP on MANETs.
Classification

• Based on who takes the action and
• What kind of action taken.

• E2E versus cross-layer approaches.
  – E2E approaches: connection end points try to distinguish between congestion and non-congestion losses.
  ✓ Cross-layer approaches: combination of e2e and intermediate node participation.
Cross-Layer Approaches

✓ Link layer error recovery.
✓ Link layer retransmission.
  ✓ TCP-awareness.
  ✓ TCP-unawareness.
✓ Split connection.
E2E Approaches

- Strict E2E versus E2E with intermediate node involvement.
- E2E with intermediate node involvement:
  - Explicit notifications.
Explicit Notification Schemes

General Philosophy

• Approximate ideal TCP behavior.
  – Ideally, TCP sender should simply retransmit a packet lost due to transmission errors without taking any congestion control actions.

• A node determines whether packets are lost due to errors and informs sender using an “explicit notification”.

• Sender, on receiving the notification, does not reduce congestion window, but retransmits lost packet.
Explicit Notification Schemes

• Motivated by Explicit Congestion Notification (ECN) proposals [Floyd94].

Variations proposed in literature differ in:
• Who sends explicit notification.
• How they decide to send explicit notification.
• What sender does on receiving notification.
Explicit Loss Notification

[Balakrishnan98]

- MH is the TCP sender.
- Wireless link first on path from sender to receiver.
- Base station keeps track of holes in packet sequence.
- When a dupack is received from the receiver, BS compares the dupack sequence number with recorded holes.
  - If there is a match, an ELN bit is set in dupack.
ELN

• When sender receives dupack with ELN set, it retransmits packet, but does not reduce congestion window.
Explicit Loss Notification
[Biaz99thesis]

- Adapts ELN proposed in [Balakrishnan98] for the case when MH is receiver.
- Caches TCP sequence numbers at base station, similar to Snoop. But does not cache data packets, unlike Snoop.
- Duplicate acks are tagged with ELN bit before being forwarded to sender if sequence number for the lost packet is cached at BS.
- Sender takes appropriate action on receiving ELN.
ELN [Biaz99thesis]

Sequence numbers cached at base station

Dupack with ELN

BS

MH

36 37 37

39 38 37

39 38 37
Explicit Bad State Notification
[Bakshi97]

- MH is TCP receiver.
- BS attempts to deliver packets to MH using link layer retransmission scheme.
- If packet cannot be delivered using “small” number of retransmissions, BS sends a Explicit Bad State Notification (EBSN) message to TCP sender.
- When TCP sender receives EBSN, timeout delayed.
  - Timeout delayed when wireless channel in bad state.

- E2E or cross-layer?
Partial Ack Protocols

[Cobb95][Biaz97]

- Send two types of acknowledgements.
- Partial ACK informs sender that a packet was received by an intermediate host (typically, base station).
- Normal TCP cumulative ACK needed by sender for reliability purposes.
Partial Ack Protocols

• When packet for which partial ack is received detected to be lost, sender does not reduce its congestion window
  – Loss assumed to be due to wireless errors.
Variations

• Base station may or may not locally buffer and retransmit lost packets.
Strict E2E Schemes
Receiver-Based Scheme
[Biaz98Asset]

- MH is TCP receiver.
- Receiver uses heuristics to “guess” cause of packet loss.
- When receiver believes that packet loss is due to errors, it sends notification to sender.
- TCP sender, on receiving notification, retransmits lost packet, without reducing congestion window.
Heuristics

• Receiver uses inter-arrival time between consecutively received packets to guess cause of packet loss.

• On determining a packet loss as being due to errors, the receiver may:
  – Tag corresponding dupacks with an ELN bit, or
  – Send an explicit notification to sender.
Receiver-Based Scheme

- Packet loss due to congestion

### Diagram

- FH
- BS
- MH

- Congestion loss

```
12 11 10
FH -- BS -- MH
```

```
12 10
FH -- BS -- MH
```

```
11

T
```

CMPE 257 Spring'15
Receiver-Based Scheme

- Packet loss due to transmission error

![Diagram showing FH, BS, and MH with packet numbers 12, 11, and 10, and error loss between 12 and 11.]
Sender-Based Discrimination Scheme
Sender-Based Discrimination
[Biaz98ic3n,Biaz99techrep]

- Sender can attempt to determine cause of a packet loss
- If packet loss determined to be due to errors, do not reduce congestion window
- Sender can only use statistics based on round-trip times, window sizes, and loss pattern.
  - Unless network provides more information (example: explicit loss notification)
Heuristics for Congestion Avoidance

• Define condition $C$ as a function of congestion window size and observed RTTs.

• Condition $C$ evaluated for new RTT.

• If ($C == \text{True}$) reduce congestion window.
Heuristics for Congestion Avoidance: Some proposals

- TCP Vegas [Brakmo94]

  expected throughput  \( ET = \frac{W(i)}{RTT_{min}} \)

  actual throughput  \( AT = \frac{W(i)}{RTT(i)} \)

  Condition \( C = (ET - AT > \beta) \)
Sender-Based Heuristics

• Record latest value evaluated for condition C
• When a packet loss is detected:
  – If last evaluation of C is TRUE, assume packet loss due to congestion.
  – Else assume packet loss due to transmission errors.
• If packet loss determined to be due to errors, do not reduce congestion window
Sender-Based Heuristics: Disadvantage

• Does not work quite well enough!!

Reason

• Not much correlation between observed short-term statistics, and onset of congestion.
Sender-Based Heuristics: Advantages

• Only sender needs to be modified

But needs better heuristics:
  – E.g., investigate longer-term heuristics.
Reliable Point-to-Point Transport Layer: Outline

✓ TCP/IP basics.
✓ Impact of transmission errors on TCP performance.
✓ Approaches to improve TCP performance on wireless networks.
  ✓ Classification.
✓ TCP on cellular.
  ▪ TCP on MANETs.
TCP in Mobile Ad Hoc Networks
Issues

• Route changes due to mobility.
  – Frequent route changes may cause OOO delivery.

• Wireless transmission errors.
  – Problem compounded due to multiple hops.

• MAC
  – MAC protocol can impact TCP performance.
Throughput over Multi-Hop Wireless Paths [Gerla99]

• When contention-based MAC protocol is used, connections over multiple hops are at a disadvantage compared to shorter connections.
  – They have to contend for wireless access at each hop.
  – Delay or drop probability increases with number of hops.
Analysis of TCP Performance over MANETs [Holland99]

• Impact of mobility.
• Simulation study.
• Performance metric: throughput.
  – Baseline: ideal (expected) throughput.
    • Upper bound.
    • Static network.
Throughput versus Hops

TCP throughput over 2 Mbps 802.11 MAC, fixed, linear MANET.
Expected Throughput

\[ \text{expected \_ throughput} = \frac{\sum_{i=1}^{\infty} t_i \cdot T_i}{\sum_{i=1}^{\infty} t_i} \]

- \( T_i \) is measured throughput for \( i \) hops using linear chain topology.
- \( t_i \) time duration of TCP connection over shortest path containing \( i \) hops.
Throughput **versus** speed

Throughput decreases with speed...

Average Throughput (Over 50 runs)

Expected

Actual
Throughput versus Speed

But not always...

Actual throughput

Mobility pattern #

30 m/s

20 m/s
Impact of Mobility
TCP Throughput

Expected throughput (Kbps)

2 m/s

10 m/s
Impact of Mobility

Expected throughput

20 m/s

30 m/s

Actual throughput
Why Throughput Degrades

mobility causes link breakage, resulting in route failure

TCP data and acks en route discarded

Route is repaired

TCP sender starts sending packets again

No throughput

No throughput despite route repair
Why Throughput Degrades?

mobility causes link breakage, resulting in route failure

TCP sender times out. Backs off timer.
Route is repaired
TCP sender resumes sending

TCP data and acks en route discarded

Larger route repair delays especially harmful

No throughput despite route repair
Why Throughput Improves?
Low Speed Scenario

1.5 second route failure

Route from A to D is broken for ~1.5 second. When TCP sender times out after 1 second, route still broken. TCP times out after another 2 seconds, and only then resumes.
Why Throughput Improves?

Higher Speed Scenario

0.75 second route failure

Route from A to D is broken for ~ 0.75 second.

Before TCP sender times (after 1 second), route is repaired.
Why Throughput Improves?

General Principle

- TCP timeout interval somewhat independent of speed.
- Network state at higher speed, when timeout occurs, may be more favorable than at lower speed.
- Network state:
  - Link/route status.
  - Route caches.
  - Congestion.
How to Improve Throughput

• Network feedback.
• Inform TCP of route failure explicitly.
• Let TCP know when route is repaired.
  – Probing.
  – Explicit notification.
• Reduce repeated TCP timeouts and backoff.
ELFN

- Explicit Link Failure Notification.
- Piggyback notification onto DSR’s route failure message to sender.
- TCP responds by disabling congestion control until route is fixed.
  - Disable retransmission timers.
  - When ACK is received, TCP restores state and resumes normal operation.
Performance Improvement

Without network feedback

With feedback

Actual throughput

Ideal throughput

2 m/s speed
Performance Improvement

Without network feedback

With feedback

Ideal throughput
30 m/s speed
Performance with Explicit Notification

![Graph showing throughput as a fraction of ideal versus mean speed (m/s)]

- **Base TCP**
- **With explicit notification**

Throughput as a fraction of ideal vs. mean speed (m/s)
Issues: Network Feedback

- Network knows best (why packets are lost).

  + Network feedback beneficial.
  - Need to modify transport & network layer to receive/send feedback

- Need mechanisms for information exchange between layers.
Impact of Route Caching

• Route caching has been suggested as a mechanism to reduce route discovery overhead (e.g., DSR).

• Each node may cache one or more routes to given destination.

• When route from S to D detected as broken, node S may:
  – Use another cached route from local cache, or
  – Obtain a new route using cached route at another node.
To Cache or Not to Cache

Actual throughput (as fraction of expected throughput) vs. Average speed (m/s) for Cache and NoCache scenarios.
Why Performance Degrades With Caching

• When a route is broken, route discovery returns cached route from local cache or from nearby node.
• Cached routes may also be broken.

| timeout due to route failure | timeout, cached route is broken | timeout, second cached route also broken |
To Cache or Not to Cache

- Caching can result in faster route “repair”.
- But, faster does not necessarily mean correct.
- If incorrect repairs occur often enough, caching performs poorly.
- Need mechanisms for determining when cached routes are stale.
Caching and TCP Performance

• Caching can reduce overhead of route discovery even if cache accuracy is not very high.

• But if cache accuracy is not high enough, gains in routing overhead may be offset by loss of TCP performance due to multiple timeouts.
Window Size After Route Repair

- When route breaks: may be too optimistic or may be too conservative.
- **Better be conservative** than overly optimistic
  - Reset window to small value after route repair.
  - TCP needs to be aware of route repair (Route Failure and Route Re-establishment Notifications).
RTO After Route Repair

• If new route longer, RTO may be too small, leading to timeouts.
• New RTO = function of old RTO, old route length, and new route length.
  – Example: new RTO = old RTO * new route length / old route length
  – Not evaluated yet.
TCP Over Different Routing Protocols [Dyer2001]

• Impact of routing algorithm on TCP performance.
  – Metrics: connect time, throughput and overhead.

• On-demand routing.
  – AODV and DSR.
  – ADV: combines on-demand with proactive updates.

• Sender-based heuristic to improve TCP’s performance.
Fixed-RTO

• TCP does not exponentially back-off the RTO after 2 consecutive timeouts.
• Uses *sender-based* heuristic to distinguish between congestion and “route failure” losses.
  – Route failure assumed if 2 consecutive timeouts.
  – Unack’d packet retransmitted.
  – No RTO backoff in the second (and +) timeout.
  – RTO remains fixed until retransmission is ack’d.