CS649
Sensor Networks
Lecture 23: Transport Protocols II

Andreas Terzis
http://hinrg.cs.jhu.edu/wn06/
Based on presentation by Bret Hull at Sensys 2004
Congestion is a problem in wireless networks

- Difficult to provision bandwidth in wireless networks
  - Unpredictable, time-varying channel
  - Network size, density variable
  - Diverse traffic patterns
- The result is congestion collapse
Outline

- Quantify the problem in a sensor network testbed
- Examine techniques to detect and react to congestion
- Evaluate the techniques
  - Individually and in concert
  - Explain which ones work and why
Investigating congestion

- 55-node Mica2 sensor network
- Multiple hops
- Traffic pattern
  - All nodes route to one sink
- B-MAC [Polastre], a CSMA MAC layer
Congestion dramatically degrades channel quality
Why does channel quality degrade?

- **Wireless is a shared medium**
  - Hidden terminal collisions
  - Many far-away transmissions corrupt packets
Per-node throughput distribution

Complementary CDF
(Fraction of nodes)

Percent of offered load received at sink

0.25 pps
Per-node throughput distribution

![Complementary CDF](Fraction of nodes)

- 0.25 pps
- 0.5 pps

Percent of offered load received at sink

0 20 40 60 80 100

0 0.2 0.4 0.6 0.8 1
Per-node throughput distribution

Complementary CDF (Fraction of nodes)

Percent of offered load received at sink

Spring 2006
Per-node throughput distribution
Goals of congestion control

- **Increase network efficiency**
  - Reduce energy consumption
  - Improve channel quality
- **Avoid starvation**
  - Improve the per-node end-to-end throughput distribution
Hop-by-hop flow control

- **Queue occupancy-based congestion detection**
  - Each node has an output packet queue
  - Monitor instantaneous output queue occupancy
  - If queue occupancy exceeds $\alpha$, indicate local congestion
Hop-by-hop flow control

- **Hop-by-hop backpressure**
  - Every packet header has a congestion bit
  - If locally congested, set congestion bit
  - Snoop downstream traffic of parent

- **Congestion-aware MAC**
  - Priority to congested nodes
Rate limiting

- **Source rate limiting**
  - Count your *parent’s number of descendents*
  - Limit your sourced traffic rate, *even if hop-by-hop flow control is not exerting backpressure*
## Congestion control strategies

<table>
<thead>
<tr>
<th>No congestion control</th>
<th>Nodes send at will</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occupancy-based hop-by-hop flow control</td>
<td>Detects congestion with queue length and exerts hop-by-hop backpressure</td>
</tr>
<tr>
<td>Source rate limiting</td>
<td>Limits rate of sourced traffic at each node</td>
</tr>
<tr>
<td><strong>Fusion</strong></td>
<td>Combines occupancy-based hop-by-hop flow control with source rate limiting</td>
</tr>
</tbody>
</table>
Evaluation setup

- Periodic workload
- Three link-level retransmits
- All nodes route to one sink using ETX
  - ETX estimates the # times a packet will have to be transmitted before it is successfully received
    - Route metric = Sum of link metrics
    - Best metric is one
    - Penalizes routes with high fwd/reverse loss rates
- Average five hops to sink
- –10 dBM transmit power
- 10 neighbors average
Metric: network efficiency

\[ \eta = \frac{\sum_{p \in \text{Received}} \text{hops}(p)}{\text{total transmit count}} \]

**Interpretation:** the fraction of transmissions that contribute to data delivery.

- Penalizes:
  - Dropped packets (buffer drops, channel losses)
  - Wasted retransmissions

2 packets from bottom node, no channel loss, buffer drop, 1 received:
\[ \eta = \frac{2}{1+2} = \frac{2}{3} \]

1 packet, 3 transmits, 1 received:
\[ \eta = \frac{1}{3} \]
Hop-by-hop flow control improves efficiency
Hop-by-hop flow control conserves packets

No congestion control

Hop-by-hop flow control
Metric: imbalance

\[ \zeta(i) = \frac{\text{received}_\ast(i)}{\text{received}_i(\text{parent}(i))} \]

**Interpretation:** measure of how well a node can deliver received packets to its parent

- \( \zeta = 1 \): deliver all received data
- \( \zeta \uparrow \): more data not delivered
Periodic workload: imbalance

CDF (Fraction of nodes)

Node throughput imbalance

No congestion control
Hop-by-hop flow control
Fusion
Rate limiting decreases sink contention

No congestion control

With only rate limiting
Rate limiting provides fairness
Hop-by-hop flow control prevents starvation
Fusion provides fairness and prevents starvation
Synergy between rate limiting and hop-by-hop flow control

Graph showing efficiency as a function of per-node offered load (pps). The graph compares different control strategies:
- No congestion control
- Fusion
- Hop-by-hop flow control
- Rate limiting

The x-axis represents per-node offered load (pps), while the y-axis represents efficiency. The data points for each control strategy are plotted, showing how efficiency decreases as load increases.
Alternatives for congestion detection

- **Queue occupancy**
- **Packet loss rate**
  - TCP uses loss to infer congestion
  - Keep link statistics: stop sending when drop rate increases
- **Channel sampling [Wan03]**
  - Carrier sense the channel periodically
  - *Congestion*: busy carrier sense more than a fraction of the time
Comparing congestion detection methods

- No congestion control
- Occupancy
- Channel sampling

Efficiency vs. Per-node offered load (pps)