Buffer sizing in 802.11 Wireless Mesh Networks

Electrical Engineering Day
Basem Shihada
‘Bufferbloat’

TCP Congestion window/RTT

Experiment Time (sec)

Time (ms)

Congestion window (Bytes)
Impact of large buffers

• TCP cwnd grows to fill available (large) buffers
  – Impacts TCP stability
  – Increases queueing delays for other flows sharing the buffer
Problem Statement

Large buffers $\rightarrow$ high throughput, high delays
small buffers $\rightarrow$ low utilization, low delays

• Determine buffer size to balance throughput & delay trade-off in WMNs
Outline

• Buffer sizing in wired networks
• Wireless challenges
• Bottlenecks and buffers in WMNs
• Performance evaluation
• Conclusions
Buffer sizing in wired networks

- Router needs a buffer size of $B = 2T \times C$
  - $2T$ is the two-way propagation delay
  - $C$ is the bottleneck link capacity
Wireless challenges

- Wireless link: abstraction for shared spectrum
  - Bottleneck spread over multiple nodes

- Variable network capacity
  - Sporadic noise and interference
  - Random MAC scheduling
Collision Domains

• Set of interfering links that contend for channel access

2-hop interference model: approximates RTS/CTS use in 802.11
Bottleneck Collision Domain

- Set of links that contend with max. no. of links
  - Limits the end-to-end rate of a flow
Cumulative Bottleneck Buffers

- Sum of buffers of nodes in the bottleneck collision domain

Neighborhood buffer size is sum of buffers of nodes 0 through 5
Two part problem

1) Determine bottleneck buffer $B$

2) Assign $b_i$ to nodes s.t. $\sum_{i \in \text{bottleneck}} b_i = B$
Step 1: Bottleneck Buffer Size

\[ B = 2T \times C \]

- Bottleneck fully utilized as long as any node in the bottleneck has a packet to transmit
- Account for channel variations by using loose bounds on \( T \) and \( C \) values
Step 2: Per-node buffer

- **Strategy 1**: Equal division: \( \frac{B}{\text{# nodes}} \)
  
  - But drops closer to source are preferable to drops closer to destination
Step 2: Per-node buffer

- **Strategy 2**: Introduces cost function s.t. cost of drop increases with hop count

\[
\min \sum_{i=1}^{M} \text{Drop probability} \times \text{cost function}
\]

subject to \( \sum_{i=1}^{M} b_i = B \)
and \( b_i \geq 0, \forall i \in M \)

where \( M \) is the number of nodes in the bottleneck collision domain
Step 2: Per node buffer

• If the cost of a packet drop increases linearly with hop count:

\[ b_1 : b_2 : \ldots : b_M = 1 : \sqrt{2} : \ldots : \sqrt{M} \]
Performance Comparisons

• Compare with
  – Default ns-2 buffer size (50 pkts)
  – TCP with adaptive pacing (TCP-AP)
    • Space packet transmissions over a 4-hop propagation delay
Performance Evaluation: Single flow

- Key observation: Collectively sizing buffers lead to small buffers (1-3 pkts) at nodes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Normalized goodput</th>
<th>Normalized RTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 pkt buffer</td>
<td>1</td>
<td>20.3</td>
</tr>
<tr>
<td>TCP-AP</td>
<td>0.90</td>
<td>1</td>
</tr>
<tr>
<td>Neighborhood buffer sizing</td>
<td>0.96</td>
<td>2.2</td>
</tr>
</tbody>
</table>

Performance statistics averaged over multiple topologies
Performance Evaluation: Multi-flows

Large FTP + G.729 VoIP

<table>
<thead>
<tr>
<th>Scheme</th>
<th>FTP</th>
<th>VoIP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Goodput (Kb/s)</td>
<td>RTT (ms)</td>
</tr>
<tr>
<td>50 pkt buffer</td>
<td>261</td>
<td>388</td>
</tr>
<tr>
<td>TCP-AP</td>
<td>240</td>
<td>54</td>
</tr>
<tr>
<td>Neighborhood buffer sizing</td>
<td>250</td>
<td>87</td>
</tr>
</tbody>
</table>
### Performance Evaluation: Multi-flows

<table>
<thead>
<tr>
<th>Scheme</th>
<th>FTP</th>
<th>VoIP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Goodput (Kb/s)</td>
<td>RTT (ms)</td>
</tr>
<tr>
<td>50 pkt buffer</td>
<td>382</td>
<td>300</td>
</tr>
<tr>
<td>TCP-AP</td>
<td>339</td>
<td>33</td>
</tr>
<tr>
<td>Neighborhood buffer sizing</td>
<td>368</td>
<td>71</td>
</tr>
</tbody>
</table>

Large FTP + G.729 VoIP
Conclusions

• Shared wireless spectrum requires rethink of bottlenecks and buffers
• Propose mechanisms for sizing bottleneck buffers and distributing it among nodes
• Simulations improve RTT by 6x - 10x over plain TCP with large buffers
Questions/Comments/Feedback

kamran.jamshaid@kaust.edu.sa