Improving TCP Fairness with MarkMax Policy

Natalia Osipova

joint work with

Alberto Blanc and
Konstantin Avrachenkov
Problem

- link capacity shearing:
  - TCP with different RTTs share a bottleneck link: TCP with smaller RTTs take a larger share of bandwidth
- share of the link capacity is proportional to

  \[ RTT^\alpha, \ 1 < \alpha < 2 \quad [\text{Laksman and Madhow, 1997}] \]

  \[ RTT^{0.85} \quad [\text{E. Altman, C. Barakat, E. Laborde, P. Brown, and D. Collange, 2000}] \]
Solutions

- standard – DropTail policy – not fair
- RED policy – more fair distribution of the capacity
- CHOKe, MLC(l), BLUE, GREEN, etc…
- based on: drop a packet with a certain probability that is a function of the state of the queue
- no differentiation between flows
MarkMax

- flow-aware AQM packet dropping scheme

- main idea:
  which connection should reduce its sending rate instead of common: which packet should be dropped.
MarkMax

- flow differentiation
- give priority to short flows
- concentrate on long flows with the largest backlog (heavy-hitter counters, hash tables)
- ECN flag instead of packets drop
MarkMax – questions

- when to send a congestion signal?
- which connection to cut?
  - according to the sending rate
- how to detect the sending rate at the bottleneck?
  - highly correlated with the backlog
MarkMax algorithm

- queue size reaches threshold
  - one selected connection is cut
  - biggest backlog
  - packet is marked with ECN flag

- three threshold scheme
  - packet model with non-zero propagation and queueing delays
MarkMax algorithm

- do nothing
- cut one selected connection and wait until reach zone
- select and cut connection every time a new packet arrives

$q$ – queue size,
$t$ – time,
$\theta_l < \theta < \theta_h$ – thresholds

Algorithm:

enqueue packet
if $q \leq \theta_l$ or $q \geq \theta_h$
    then $flag \leftarrow TRUE$
if $q \geq \theta$ and $flag=TRUE$
    then a. select connection
       b. set the ECN flag in the first packet of the selected connection from the head of the queue
       c. $flag \leftarrow FALSE$
MarkMax – thresholds selection

- high $\theta_h$
- slow system reaction – long waiting time
- low $\theta_l$
- not reached – system behaves as DropTail
MarkMax – thresholds selection

- low \( \theta_l \)
- high \( \theta_h \)
- provide multiple cuts

\[ q \text{ oscillates} \]
MarkMax – thresholds selection

- $\theta_h$ not reached
- $\theta_l$ is reached
- one cut is enough every time
- experimental results:
  \[
  \theta_l = 0.85 \theta, \\
  \theta_h = 1.15 \theta
  \]
Fluid model

- simplify calculations
- cut flow with the biggest sending rate
- biggest backlog -> biggest *average* sending rate
- fluid model simulations:
  - threshold is reasonably small, then
  - results for biggest sending rate and biggest backlog are nearly the same
Fluid model

- $N$ TCP connections - flows
- $RTT_i$ – constants,
- $\lambda_i(t) = \lambda_{0,i} + \alpha_i t$, \hspace{1em} ($\alpha_i = 1/(RTT_i)^2$), – sending rate of i-th flow
- $\lambda(t) = \sum_i \lambda_i(t)$, – total sending rate
- $\mu$ – rate with which data leaves the buffer
Fluid model – MarkMax

- MarkMax modeling:
  
  \[ \lambda^+ = \sum_{j \neq i} \lambda_j^- + \beta \lambda_i^- \]

  when \( x(\lambda) = \theta \)

  \( \lambda^+ < \mu \) – stop cutting

- cut: rate is multiplied by fixed parameter \( \beta, \, 0 < \beta < 1 \)

- source reacts immediately

- one threshold
  - no oscillations
  - sending rate known exact
Fluid model

- Mathematical results: threshold selection

\[ \text{if } \theta < \frac{\mu^2}{2\alpha} \frac{(1 - \beta)^2}{(N - 1 + \beta)^2}, \quad \text{then } \lambda^+ < \mu \text{ after a single cut,} \]

\[ \text{if } \theta > \frac{\mu^2}{2\alpha} \left(1 - \frac{\beta \mu}{\mu + \sqrt{2\alpha \theta}}\right)^2, \quad \text{then } x(\lambda) > 0, \]

positive backlog and full link utilization

- Obtained theoretical results confirmed by the NS2 simulations
NS2 simulations

- NS2 simulator
- TCP NewReno
- MarkMax realization
- MarkMax and DropTail comparison
NS2 simulations – metrics and parameters

Metrics:

- $\rho$ – bottleneck link utilization
- $\bar{T}$ – average queueing delay

$$J = \frac{\left(\sum_{i=1}^{N} g_i\right)^2}{N \sum_{i=1}^{N} g_i^2}, \quad \text{Jain’s index,} \quad g_i \quad \text{goodputs}$$

Parameters:

- $\delta_i$ – propagation and queue delays,
- $\frac{\delta_2}{\delta_1} = 3; 7; 10; 20; 50$
NS2 simulations – results scheme1

Parameters:
\( \mu = 70 \text{ Mbit/s}, \quad \mu_1 = \mu_2 = 300 \text{ Mbit/s}, \)
\( \delta_1 = 12 \text{ ms}, \)
\( \delta_2 / \delta_1 = 3; 7; 10; 20, \)
MarkMax: \( \theta = 240 \text{ MSS}, \quad \theta_1 = 200 \text{ MSS}, \quad \theta_2 = 280 \text{ MSS}, \)
DropTail: \( \theta_{\text{DT}} = 240 \text{ MSS}. \)

<table>
<thead>
<tr>
<th>( \frac{\delta_2}{\delta_1} )</th>
<th>DT</th>
<th>MM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( J )</td>
<td>( \rho )</td>
</tr>
<tr>
<td>3</td>
<td>0.9893</td>
<td>0.9751</td>
</tr>
<tr>
<td>7</td>
<td>0.7540</td>
<td>0.9720</td>
</tr>
<tr>
<td>10</td>
<td>0.5361</td>
<td>0.9563</td>
</tr>
<tr>
<td>20</td>
<td>0.5484</td>
<td>0.9993</td>
</tr>
</tbody>
</table>
NS2 simulations – results scheme2

Parameters:
\[ \mu = 70 \text{ Mbit/s}, \quad \mu_1 = \mu_2 = 300 \text{ Mbit/s}, \]
\[ \delta_1 = 12 \text{ ms}, \quad \delta_3 = \delta_2 \]
\[ \delta_2/\delta_1 = 7; 10; 20; 50, \]

MarkMax: \( \theta = 240 \text{ MSS}, \ \theta_1 = 200 \text{ MSS}, \ \theta_h = 280 \text{ MSS}, \)
DropTail: \( \theta_{\text{DT}} = 240 \text{ MSS}. \)

<table>
<thead>
<tr>
<th>( \frac{\delta_2}{\delta_1} )</th>
<th>DT</th>
<th>MM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( J )</td>
<td>( \rho )</td>
</tr>
<tr>
<td>7</td>
<td>0.8561</td>
<td>0.9338</td>
</tr>
<tr>
<td>10</td>
<td>0.7769</td>
<td>0.9497</td>
</tr>
<tr>
<td>20</td>
<td>0.6910</td>
<td>0.9146</td>
</tr>
<tr>
<td>50</td>
<td>0.5244</td>
<td>0.9262</td>
</tr>
</tbody>
</table>
NS2 simulations – results comparison

- Congestion window: MarkMax and DropTail

$$\frac{\delta_2}{\delta_1} = 7$$
NS2 simulations – results comparison

- Congestion window: MarkMax and DropTail

\[ \frac{\delta_2}{\delta_1} = 10 \]
Conclusion and future work

- New AQM algorithm
- Fluid model - theoretical results
- NS2 simulations - confirm theoretical results

Future work:
- Multiple connections – cut several connections at a time
- More complex network topology
Thank you for your attention!

Questions?