On Hybrid TCP Congestion Control

J. Katto, K. Ogura, T. Fujikawa, K. Kaneko & S. Zhou

Dept. of Computer Science, Waseda University
Outline

- Research Backgrounds & TCP-Fusion
  - TCP Variants
  - TFRC: TCP Friendly Rate Control
  - TCP-Fusion

- Analytical Models & Performance Analysis
  - Ideal Models (Loss, Delay, Hybrid)
  - TCP-Fusion Extensions
  - Delay-based TFRC

- Conclusions
Backgrounds: TCP Variants (1)

- **Loss-based (AIMD: Additive Increase Multiplicative Decrease)**
  - TCP-Reno / NewReno / SACK
  - High-Speed TCP (IETF RFC 3649, Dec 2003)
  - Scalable TCP (PFLDnet 2003)
  - H-TCP (PFLDnet 2004)
  - TCP-Westwood (ACM MOBICOM 2001)

- **Delay-based (RTT Observation)**
  - TCP-Vegas (IEEE JSAC, Oct 1995)
  - FAST-TCP (INFOCOM 2004)

- **Hybrid (Loss & RTT)**
  - Gentle High-Speed TCP (PfHSN 2003)
  - TCP-Africa (IEEE INFOCOM 2005)
  - Compound TCP (PFLDnet 2006)
  - Adaptive Reno (PFLDnet 2006)
  - TCP-Illinois (ValueTools 2006)
  - YeAH-TCP (PFLDnet 2007)
  - TCP-Fusion (PFLDnet 2007)

Quite many proposals!
Backgrounds: TCP Variants (2)

- **Loss-based**
  - RTT increase: Loss
  - no RTT increase: Delay
  - $cwnd$: outgoing window size
  - $n$: number of packets
  - $RTT$ round

- **Delay-based**
  - $BDP$: Bandwidth-Delay Product
  - $loss$: loss
  - $buffer$: stored packets in buffer
  - $\alpha$: decrease rate when competing with AIMD

- **Hybrid**
  - $a=1$
  - $b=0.5$
  - $BDP (capacity)$
## Backgrounds: TCP Variants (3)

### Loss-based Variants

<table>
<thead>
<tr>
<th>Variants</th>
<th>Increase / Update</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP-Reno</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>HighSpeed TCP (HS-TCP)</td>
<td>$a(w) = \frac{2w^2 \cdot b(w) \cdot p(w)}{2 - b(w)}$ e.g. 70 (10Gbps, 100ms)</td>
<td>$b(w) = \frac{\log(w) - \log(W_{\text{max}})}{\log(W_{\text{max}}) - \log(W_{\text{min}})}(b_{\text{def}} - 0.5) + 0.5$ e.g. 0.1 (10Gbps, 100ms)</td>
</tr>
<tr>
<td>Scalable TCP (STCP)</td>
<td>0.01 (per every ACK)</td>
<td>0.875</td>
</tr>
<tr>
<td>BIC-TCP</td>
<td>additive increase (fast) binary search (slow) max probing (fast)</td>
<td>0.875</td>
</tr>
<tr>
<td>CUBIC-TCP</td>
<td>$w = 0.4 \left( t - \frac{3}{2} \sqrt{W_{\text{max}}} \right)^3 + W_{\text{max}}$</td>
<td>0.8</td>
</tr>
<tr>
<td>H-TCP</td>
<td>$\alpha \leftarrow 2(1 - \beta)\left(1 + 10.5 \cdot (t - TH)\right)$</td>
<td>$\beta \leftarrow \begin{cases} 0.5 &amp; \text{for } \frac{B(k + 1) - B(k)}{B(k)} &gt; 2 \ \frac{\text{RTT}<em>{\text{min}}}{\text{RTT}</em>{\text{max}}} &amp; \text{otherwise} \end{cases}$</td>
</tr>
<tr>
<td>TCP-Westwood (TCPW)</td>
<td>1</td>
<td>$\begin{cases} \text{RE} \cdot \text{RTT}<em>{\text{min}} / \text{PS} &amp; \text{(not congested)} \ \text{BE} \cdot \text{RTT}</em>{\text{min}} / \text{PS} &amp; \text{(congested)} \end{cases}$</td>
</tr>
</tbody>
</table>
### Backgrounds: TCP Variants (4)

- **Delay-based**

<table>
<thead>
<tr>
<th>Variants</th>
<th>Update</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP-Vegas</td>
<td>$w \left{ \begin{array}{ll} w+1 &amp; (no\ congestion) \ w &amp; (stable) \ w-1 &amp; (early\ congestion) \end{array} \right.$</td>
<td>0.75</td>
</tr>
<tr>
<td>FAST-TCP</td>
<td>$w \left{ \begin{array}{ll} 2w, (1-\gamma)w + \gamma \left( \frac{RTT_{\min}}{RTT} w + \alpha \right) \end{array} \right.$</td>
<td>0.5 (?)</td>
</tr>
</tbody>
</table>
## Backgrounds: TCP Variants (5)

### Hybrid

<table>
<thead>
<tr>
<th>Variants</th>
<th>Increase</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gentle HS-TCP</td>
<td>HS-TCP / Reno</td>
<td>HS-TCP</td>
</tr>
<tr>
<td>TCP-Africa</td>
<td>HS-TCP / Reno</td>
<td>HS-TCP</td>
</tr>
<tr>
<td>Compound TCP (CTCP)</td>
<td>$0.125 \cdot cwnd^{0.75} / \text{Reno}$</td>
<td>0.5</td>
</tr>
<tr>
<td>Adaptive Reno (ARENO)</td>
<td>$B/10\text{Mbps} / \text{Reno}$</td>
<td>$\begin{cases} 1 \ 0.5 \end{cases}$ (non congestion loss) (congestion loss)</td>
</tr>
<tr>
<td>YeAH-TCP</td>
<td>STCP / Reno</td>
<td>$\max\left(\frac{\text{RTT}_{\text{min}}}{\text{RTT}}, 0.5\right)$</td>
</tr>
<tr>
<td>TCP-Fusion</td>
<td>$\frac{B \cdot D_{\text{min}}}{PS} / \text{Reno}$</td>
<td>$\max\left(\frac{\text{RTT}_{\text{min}}}{\text{RTT}}, 0.5\right)$</td>
</tr>
</tbody>
</table>
Backgrounds: Models & TFRC (1)

- **TFRC: TCP Friendly Rate Control**

  * equation based rate control

  \[
  w: \text{cwnd when packet losses happen} \\
  p: \text{packet loss ratio} \\
  \text{RTT: round trip time} \\
  R: \text{(equivalent) rate} \\
  b: \text{delayed ACK counts}
  \]

  \[
  \begin{align*}
  p &= \frac{8}{3w^2} \\
  R &= \frac{PS}{RTT} \cdot \sqrt{\frac{3}{2p}}
  \end{align*}
  \]

  \[
  R_{\text{loss}} = \frac{PS}{RTT \sqrt{\frac{2bp}{3} + t_{\text{RTO,loss}} \cdot \frac{3bp}{8}} \cdot p(1+32p^2)}
  \]
VTP: Video Transport Protocol

\[ R_{i+1} = \frac{\text{ewnd}_{i+1}}{\text{RTT}_{i+1}} = \frac{R_i \times \text{RTT}_i + 1}{\text{RTT}_i + \Delta \text{RTT}} \]

\[ \Delta \text{RTT} = \text{RTT}_i - \text{RTT}_{i-1} \]

RTT_{min}: rate increase
RTT increase: compensate rate increase by \( \Delta \text{RTT} \)

similar behavior to TCP-Reno
Problems & Objectives

- **On Hybrid TCP Congestion Control**
  - Theoretical background is not sufficient (best performance is not clear)
  - Too many tuning parameters (we have to tune them by many simulations and implementations)
  - Building an analytical model for hybrid TCP congestion control
  - Theoretically validates its throughput efficiency and friendliness against legacy TCPs
  - Enables analytical parameter tuning
  - Formulates a new TFRC method
Basic mechanism (single flow case: no competing flow)

Window control by switching two modes (loss & delay):
① Keeps constant rate until RTT increases (delay mode)
② Performs as TCP-Reno (loss mode)
TCP-Fusion (2)

- Basic mechanism (two flow case: when competing with TCP-Reno)

Window control by switching two modes (loss & delay):
① Rapid increase of cwnd (delay ... throughput efficiency)
② Gradual decrease of cwnd (delay ... avoids congestion)
③ Performs as TCP-Reno (loss ... friendliness to legacy TCP)
TCP-Fusion (3)

- Actual window control algorithm (later)

\[
cwnd_{\text{new}} = \begin{cases} 
  cwnd_{\text{last}} + W_{\text{inc}} / cwnd_{\text{last}}, & \text{if } diff < \alpha \quad \rightarrow (2)-\overline{1} \\
  cwnd_{\text{last}} + (-diff + \alpha) / cwnd_{\text{last}}, & \text{if } diff > 3 \alpha \quad \rightarrow (2)-\overline{2} \\
  cwnd_{\text{last}}, & \text{otherwise} \quad \rightarrow (1)-\overline{1}
\end{cases}
\]

\[
cwnd_{\text{new}} = \text{reno}_- cwnd, \quad \text{if } cwnd_{\text{new}} < \text{reno}_- cwnd \quad \rightarrow (1)-\overline{2} \quad (2)-\overline{3}
\]

- $W_{\text{inc}}$: window increase parameter
- $\alpha$: target number of buffered packets
- $diff$: estimated number of buffered packets
Ideal Model (1)

- Ideal Model Definition
  - Loss-based (AIMD: TCP-Reno):
    - $cwnd += 1$ (per RTT round)
    - $cwnd *= 1/2$ (when packet losses happen)
  - Delay-based:
    - always fills a pipe (BDP) without causing RTT increase
  - Hybrid:
    - performs in delay-mode when RTT stays at its minimum
    - performs in loss-mode when RTT increase is observed
Ideal Model (2)

- Parameter definition
  - $w$: cwnd when packet losses happen
  - $W$: number of packets corresponding to BDP
  - $p$: packet loss rate

- Assumption
  - Next equation holds for random loss case (e.g. RED router and wireless) as well as for buffer overflow case
    
    $$p = \frac{8}{3w^2} \quad \text{(in case of TCP-Reno)}$$
Single Flow Model (1)

- Connection Topology

![Diagram of a single flow model with sender, bottleneck link, and receiver connected by a single TCP flow.]
Classification to three cases

(i) $W < \frac{w}{2}$ (low PER)  
always buffering  
(loss-mode)

(ii) $\frac{w}{2} < W < w$ (medium PER)  
vacant $\rightarrow$ buffering  
(delay $\rightarrow$ loss)

(iii) $w < W$ (high PER)  
always vacant  
(delay-mode)
### Single Flow Model (3)

- Transmitted packets and elapsed time per single congestion avoidance round

<table>
<thead>
<tr>
<th>TCP</th>
<th>CA round</th>
<th>(i) $W &lt; w/2$</th>
<th>(ii) $w/2 \leq W &lt; w$</th>
<th>(iii) $w \leq W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>transmitted packets</td>
<td>$\frac{3}{8}w^2$</td>
<td>$\frac{3}{8}w^2$</td>
<td>$\frac{3}{8}w^2$</td>
</tr>
<tr>
<td></td>
<td>elapsed time</td>
<td>$\frac{1}{2}w \cdot RTT_{min} + \frac{1}{8}(3w^2 - 4wW) \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2}w \cdot RTT_{min} + \frac{1}{2}(w-W)^2 \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2}w \cdot RTT_{min}$</td>
</tr>
<tr>
<td>Delay</td>
<td>transmitted packets</td>
<td>$\frac{1}{2}w \cdot W$</td>
<td>$\frac{1}{2}w \cdot W$</td>
<td>$\frac{1}{2}w \cdot W$</td>
</tr>
<tr>
<td></td>
<td>elapsed time</td>
<td>$\frac{1}{2}w \cdot RTT_{min}$</td>
<td>$\frac{1}{2}w \cdot RTT_{min}$</td>
<td>$\frac{1}{2}w \cdot RTT_{min}$</td>
</tr>
<tr>
<td>Hybrid</td>
<td>transmitted packets</td>
<td>$\frac{3}{8}w^2$</td>
<td>$\frac{1}{2}w \cdot W + \frac{1}{2}(w-W)^2$</td>
<td>$\frac{1}{2}w \cdot W$</td>
</tr>
<tr>
<td></td>
<td>elapsed time</td>
<td>$\frac{1}{2}w \cdot RTT_{min} + \frac{1}{8}(3w^2 - 4wW) \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2}w \cdot RTT_{min} + \frac{1}{2}(w-W)^2 \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2}w \cdot RTT_{min}$</td>
</tr>
</tbody>
</table>
Single Flow Model (4)

- Incorporation of timeout penalty
  - Timeout penalty of TCP-Reno
    \[ t_{\text{RTO,loss}} = T_0 \left( 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6 \right)/(1 - p) \]
  - Ratio of transmitted packets
    \[ t_{\text{RTO,delay}} = \frac{K_{\text{delay}}}{K_{\text{loss}}} \cdot t_{\text{RTO,loss}} \]
  - Estimated throughput
    \[ \frac{\text{transmitted packets}}{\text{elapsed time + timeout penalty}} \]
Single Flow Model (5)

- Result (1) PLR

When PLR is large (in regions (ii) & (iii)), delay based and hybrid TCPs drastically outperform loss based TCP.
Single Flow Model (6)

- Result (2) buffer size

- 1Gbps
- 1Gbps
- 100Mbps
- RTT = 40ms

PLR = 10^-6 (constant)
buffer size: variable
Two Flow Model (1)

- Connection Topology

Diagram:
- Senders
- Receivers
- Bottleneck link
- Loss-based TCP flow
- Loss-based or hybrid TCP flow
Two Flow Model (2)

Classification to three cases

(i) $W < w$ (low PLR)
always buffering (loss mode)

(ii) $w < W < 2w$ (medium PLR)
vacant $\rightarrow$ buffering (delay $\rightarrow$ loss)

(iii) $2w < W$ (high PLR)
always vacant (delay mode)

(1) $w$ is scaled to half value
(2) delay-based TCP is omitted
Two Flow Model (3)

- Transmitted packets and elapsed time per single congestion avoidance round

<table>
<thead>
<tr>
<th>TCP</th>
<th>CA round</th>
<th>(i) ( W &lt; w )</th>
<th>(ii) ( w \leq W &lt; 2w )</th>
<th>(iii) ( 2w \leq W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>transmitted packets</td>
<td>( \frac{3}{8}w^2 )</td>
<td>( \frac{3}{8}w^2 )</td>
<td>( \frac{3}{8}w^2 )</td>
</tr>
<tr>
<td>Hybrid</td>
<td>transmitted packets</td>
<td>( \frac{3}{8}w^2 )</td>
<td>( \frac{3}{8}w^2 + \frac{1}{4}(W-w)^2 )</td>
<td>( \frac{1}{2}w\cdot W - \frac{3}{8}w^2 )</td>
</tr>
<tr>
<td>(common) elapsed time</td>
<td>( \frac{1}{2}w\cdot RTT_{\text{min}} + \frac{1}{4}w(3w-2W) \cdot \frac{PS}{B} )</td>
<td>( \frac{1}{2}w\cdot RTT_{\text{min}} + \frac{1}{4}(2w-W)^2 \cdot \frac{PS}{B} )</td>
<td>( \frac{1}{2}w\cdot RTT_{\text{min}} )</td>
<td></td>
</tr>
</tbody>
</table>
Two Flow Model (4)

Result

When PLR is low (region (i)), hybrid TCP shows good friendliness to legacy TCP.
When PLR is high (regions (ii) & (iii)), hybrid TCP shows high throughput efficiency.
Actual window control algorithm

\[ cwnd_{new} = \begin{cases} 
    cwnd_{last} + \frac{W_{inc}}{cwnd_{last}}, & \text{if } \text{diff} < \alpha \\
    cwnd_{last} + (-\text{diff} + \alpha) / cwnd_{last}, & \text{if } \text{diff} > 3\alpha \\
    cwnd_{last}, & \text{otherwise}
\end{cases} \]

\[ cwnd_{new} = \text{reno}_\_cwnd, \text{ if } cwnd_{new} < \text{reno}_\_cwnd \]

\( W_{inc} \): window increase parameter
\( \alpha \): target number of buffered packets
\( \text{diff} \): estimated number of buffered packets
TCP-Fusion (4)

- $\alpha$ and $W_{inc}$:

  - $N$: number of flows
  - $B$: link bandwidth estimation
  - $RE$: rate estimation
  - $PS$: packet size
  - $B$ and $RE$ can be estimated by using TCP-Westwood mechanism

  $D_{min}$: minimum time resolution which can be detected by the end host

  $G$: minimum buffer size which can be estimated by end host

  $G$: minimum buffer size which can be estimated by end host

$$
\alpha = \frac{G}{N} = \frac{(B/N) \cdot D_{min}}{PS} \approx \frac{RE \cdot D_{min}}{PS}
$$

$$
W_{inc} \leq G = \frac{B \cdot D_{min}}{PS}
$$

Minimum packets which will not cause buffer overflow and scales to the number of flows.

The number of packets which can recover to $\alpha$ buffered packet state in a single RTT round.
Model Improvement (1)

- Single Flow Model (1) $W < w/2$

(i) $W + \alpha < w/2$

(ii) $W < w/2 \leq W + \alpha$

Diagram:
- cwnd vs. n
- Slope = 1
- Loss
- Hybrid
- TCP-Fusion

Equation:
\[
\text{slope} = \frac{W_{\text{inc}}}{W}
\]

Conditions:
- $W - w/2 + \alpha$
- $W_{\text{inc}} = W - w/2 + \alpha$
- $W_{\text{inc}}$
- $W - w/2 + \alpha$

Kattn lab.
Model Improvement (2)

- Single Flow Model (2) $W > w/2$

$$cwnd$$

- slope=$W_{inc}$
- extra packets
- vacant
- retransmission round

$$\text{loss}$$
$$\text{hybrid}$$
$$\text{TCP-Fusion}$$

$BDP$
Model Improvement (3)

- Two Flow Model (1) low PLR

(i) \( W + \alpha < w \)

(ii) \( W < w \leq W + \alpha \)

\[
S = \frac{(W - w + \alpha)^2(W_{inc} - 1)}{4(W_{inc} + 1)}
\]
Model Improvement (4)

- Two Flow Model (2) high PLR

\[ b = \frac{W - w}{2(1 + W_{inc})}, \quad c = \frac{W - w + 2\alpha}{2(1 + W_{inc})}, \quad d = \frac{W - w + \alpha}{2} \]

![Diagram showing cwnd with postponed packets, extra packets, and retransmission round.

1. Postponed packets with slope=\( W_{inc} \)
2. Extra packets with slope=1
3. Retransmission round with \( \alpha \) shift

- Loss (brown)
- Hybrid (green)
- TCP-Fusion (dotted black)
Model Improvement (5)

- Result (1) effect of parameter $\alpha$

Buffered packets:
- small $\alpha \Rightarrow$ small change
- large $\alpha \Rightarrow$ large change.

TCP-Fusion expels TCP Reno (i.e. unfriendly)
Model Improvement (6)

- Result (2) effect of parameter $W_{inc}$

- Window increase parameter:
  - large $W_{inc} \Rightarrow$ small change
  - small $W_{inc} \Rightarrow$ TCP-Fusion decreases but TCP Reno stays the same (inefficient, too conservative cwnd increase)
Delay-based TFRC (1)

- Two flow model for delay-based TCP

\[ \text{total} \]

\[ \text{delay} \]

\[ \text{BDP} \]
Delay-based TFRC (2)

Throughput derivation

\[ R_{delay} = \frac{1}{2} \frac{w \left( \frac{W}{M} + \frac{m}{M} \cdot \beta \right) \cdot PS}{w (RTT_{min} + m \cdot D_{min}) + t_{RTO, delay}} \]

\[ R_{delay} = \frac{B}{M} \frac{1 + \frac{m}{M} \cdot \frac{D_{min}}{RTT_{min}}}{\left( 1 + m \cdot \frac{D_{min}}{RTT_{min}} \right) + p \cdot \frac{B}{PS} \cdot t_{RTO, loss}} \]

\[ R_{delay} = \frac{B / M}{1 + p \cdot B / PS \cdot t_{RTO, loss}} \quad \text{(when } D_{min} = 0) \]

Shares bandwidth \( B \) by \( M \) flows.
Throughput decreases as \( p \) increases.
Delay-based TFRC (3)

- **Result**

\[ R_{\text{loss}} = \frac{\frac{2bp}{3} + t_{RTO,\text{loss}} \cdot \frac{3bp}{8} \cdot p(1 + 2p^2)}{RTT} \]

\[ R_{\text{delay}} = \frac{B / M}{1 + p \cdot B / PS \cdot t_{RTO,\text{loss}}} \]
Conclusions (1)

- Performance analysis of hybrid TCP
  - Ideal model (from Ref[10]):
    - Three models: loss-based, delay-based and hybrid
    - Single flow model and two flow (competing flow) model
    - Evaluations by analysis and simulations
  - Model improvement (specific to TCP-Fusion):
    - Incorporation of control parameters of TCP-Fusion:
      - number of buffered packets $\alpha$ and window increase parameter $W_{inc}$
    - Evaluations by analysis and simulations
  - Delay-based TFRC:
    - Derivation of TFRC based on delay-based TCP
    - Comparison with loss-based TCP & TFRC (only analysis)
Conclusions (2)

- Future work
  - Extension to many flows
  - Extension to wireless environment (e.g. incorporation of layer 2 retransmission)
  - Incorporation of recent loss-based TCPs (e.g. CUBIC and H-TCP)
  - Evaluations by implementations over actual Internet
  
  - Model sophistication
  - Extension to RTT fairness
  - Evaluations of reverse traffic and short-lived flow
特性解析: TCP-Fusion 拡張 (5)

- 競合フローモデル (3) 高廃棄率(続き)

![特性解析: TCP-Fusion 拡張 (5) 救命フローモデル (3) 高廃棄率(続き)](image-url)
特性解析: TCP-Fusion 拡張 (8)

■ $D_{\text{min}}$の影響

TCPタイマ解像度

$D_{\text{min}}$小 ⇒ $W_{\text{inc}}$小 ⇒ Fusion
減少、Renoほとんど変わらず (inefficient)

$D_{\text{min}}$大 ⇒ $\alpha$大 ⇒ Fusion増加、Reno減少 (unfriendly)

$W_{\text{inc}}$の決め方は要検討