Analytic Performance Evaluation of the RED Algorithm

Stefan Köhler, Michael Menth, Norbert Vicari

- TCP Model
- RED Model
- TCP over RED
- Results
TCP

- Reliable transmission
- Closed loop flow control
- Elastic traffic without real-time requirements
- Major part of Internet traffic
- Interactive Applications
TCP’s slow start and congestion avoidance

The diagram illustrates the process of TCP's slow start and congestion avoidance. Initially, the window size is small, and the sender sends a sequence of ACKs, indicating the acknowledgement of received segments. As the congestion avoidance phase begins, the window size increases linearly, acknowledging ACKs for all outstanding segments. The graph shows the evolution of the window size and the transmission number, highlighting the transition from slow start to congestion avoidance.

ACK = all outstanding ACKs of the previous burst are acknowledged.
TCP’s fast retransmit algorithm

3*ACK = three duplicate ACKs are received for the segment sent immediately before the lost segment.

ACK = all outstanding ACKs of the previous burst are acknowledged.

window size

4

32

16

size

17

size

2

fast recovery

fast retransmit

congestion avoidance

transmission number

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 31 32

3*ACK = three duplicate ACKs are received for the segment sent immediately before the lost segment.

ACK = all outstanding ACKs of the previous burst are acknowledged.

window

advertised

window

University of Würzburg
Distributed Systems
Analytical Approach

- Semi-Markov process (SMP)
- Renewal points: Time instant when last packet that has seen the last renewal point has left the queue.
- „TCP rounds“

![Diagram showing Sender and Receiver connected by ACK packets with numbers 14 to 19]
Renewal Points
Discrete-Time Model for a Single TCP Connection

Assumptions

- TCP Reno
- FTP-source
- Independent packet losses (later based on RED)
- State variables:
  \[(W_n, S_n, M_n) := (\text{CWND}(t), \text{SSTRESH}(t), \text{Loss}(t))\]
- Observation: TCP-Round
A Single TCP Connection

**Input:** model state \((W_n, S_n, M_n)\), model factor \((L(W_n))\)

\[
\begin{align*}
\text{if } (M_n = 0) & \text{ then } \{\text{no loss last round}\} \\
S_{n+1} & := S_n \\
\text{if } (W_n = W_{\text{max}}) & \text{ then } \{\text{full window possible}\} \\
W_{n+1} & := W_n \\
\text{else} \\
\text{if } (W_n < S_n) & \text{ then } \{\text{slow start}\} \\
W_{n+1} & := 2W_n \\
\text{else} & \{\text{congestion avoidance}\} \\
W_{n+1} & := W_n + 1 \\
\text{endif} \\
\text{endif} \\
\text{else}
\end{align*}
\]
else
  if \( (M_n = 1) \) then \{one loss last round\}
    \[ S_{n+1} := \max(W_n/2, 2) \]
    \[ W_{n+1} := S_{n+1} \]
  else \{more than one loss last round\}
    \[ S_{n+1} := \max(W_n/2, 2) \]
    \[ W_{n+1} := 1 \]
  endif
endif

\[ M_{n+1} := \min(L(W_n), 2) \]

**Output:** model state\((W_{n+1}, S_{n+1}, M_{n+1})\)
Simulation ⇔ Analysis (I)

\[ P(\text{CWND}) \]

- simulation
- analysis
- \( p=0.01 \)
Simulation $\Leftrightarrow$ Analysis (II)

$p = 0.1$

$P(CWND)$ vs. CWND

- Simulation
- Analysis

University of Würzburg
Distributed Systems
Random Early Discard (RED) Queue

Drop Probability:
\[ P(AQL(t)) \]

\[ p_{\text{max}} \]

\[ r_{\text{min}} \]

\[ r_{\text{max}} \]

Average Queue Length

\[ p(A = i) = \begin{cases} 0 & 0 \leq i < r_{\text{min}} \\ \left( \frac{i - r_{\text{min}}}{r_{\text{max}} - r_{\text{min}}} \right) \cdot p_{\text{max}} & r_{\text{min}} \leq i \leq r_{\text{max}} \\ 1 & r_{\text{max}} < i \leq \infty \end{cases} \]
RED Queue Mechanism

**Input:** model state \((A_n)\), model factor \((B, L(B,A_n))\)

\[
Q := B - L(B, A_n)
\]

\[
A_{n+1} := w_q \cdot Q + (1-w_q) \cdot A_n
\]

**Output:** model state \((A_{n+1})\)

The probability of \(k\) losses within a batch \(B\) of \(j\) packets follows a binomial distribution:

\[
l(A = i, B = j)[k] = \binom{j}{k} p(A = i)^k \cdot (1 - p(A = i))^{j-k}
\]

B: batch of packets  
A: average queue size  
L: loss  
Q: actual queue size  
w_q: weighting factor
Compound Analysis

▷ h TCP Connections under RED:

**Input:** model state \( ((W_i^n, S_i^n, M_i^n), (A_n)) \), model factor \( L_i(A_n, W_i^n) \)

```plaintext
for \( i \in \{1, \ldots, h\} \) do
  \( (W_{i+1}^n, S_{i+1}^n, M_{i+1}^n) := TCP((W_i^n, S_i^n, M_i^n), L_i(A_n, W_i^n)) \)
end for

\( A_{n+1} := RED( (A_n), (\sum W_i^n, \sum L_i(A_n, W_i)) ) \)
```

**Output:** model state \( ((W_{i+1}^n, S_{i+1}^n, M_{i+1}^n), (Q_{n+1}, A_{n+1})) \)
Results
Parameters

If not stated differently

- $R_{\text{min}} = 9$
- $R_{\text{max}} = 18$
- $w_q = 0.3$
- 3 TCP sources
- CWND = 6
Influence of the Weighting Factor

$p_{\text{max}} = 0.5$

- Increased throughput, reduced variance
Influence of the Loss Function

- High loss probability reduces throughput.
- “Long Memory” provides better results.
Influence of the Buffer Size

Short (or congested) queues limit the congestion window size.

\( w_q = 0.3 \)

\( w_q = 0.3, R_{\text{max}} = 18 \)
\( w_q = 0.3, R_{\text{max}} = 12 \)
\( w_q = 1.0, R_{\text{max}} = 12 \)

\( p_{\text{max}} \) succeeded transmitted packets

\( w_q = 0.3, R_{\text{max}} = 18 \)
\( w_q = 0.3, R_{\text{max}} = 12 \)
\( w_q = 1.0, R_{\text{max}} = 12 \)

\( p_{\text{max}} \) mean CWND

\( w_q = 0.3 \)

\( w_q = 0.3 \leftrightarrow \text{RED} \)
“Long Memory” and small loss probabilities show larger congestion windows.
Conclusion and Outlook

Summary
- Discrete-Time Model of TCP and RED
  - Correlation of TCP sources
- Analytical Performance Evaluation
- Distributions for all TCP state variables

Results
- Sensitivity of TCP to
  - Weighting factor
  - Loss function
  - Congestion or short queues
- Good Performance: “Smoothed FIFO-Queues”

Outlook
- Comparison of different TCP implementations
- Influence of non-linear loss functions
- Fairness studies
THE END