On the Modeling of TCP Latency and Throughput

Dong Zheng

MASTER’S THESIS PRESENTATION

Department of Electrical and Computer Engineering
Mississippi State University
Outline

- Introduction and Motivation
- Background Information on TCP
- Building the Stochastic Models
- Model Validation by Simulation
- Conclusion and Future Work
What Is The Problem?
- From Practical View

- TCP’s performance dominates behavior of Internet traffic, inspiring tremendous research on stochastic TCP model

1. improve TCP performance by understanding the sensitivity of TCP performance to the network conditions
2. help design of active queue management
3. aid in the design of TCP-friendly transfer multicast protocols

— An accurate model of TCP performance is needed
What Is The Problem?
-From the Model’s View

- Most existing models doesn’t include the analysis of time-outs effects
- Models including the analysis of time-outs underestimate it
- None of the existing steady state model include the slow start phase
- Not accurate modeling of the delayed acknowledgment’s effect in the slow start phase

— New coupled models are needed
What Is This Research All About?

• Develop better and tractable model for slow start
• Develop complete steady state model including the slow start phase
• Develop accurate model for short-lived TCP flows
Why Include the Slow Start?

- Slow start phase begins whenever TCP recovers from time-out phase
- Empirical studies observed that slow start phase occurs often for long-lived TCP flows
- Models that ignored slow start overestimate TCP performance
  — Including slow start phase into steady state analysis results in accurate performance predictions
Why Need New Models for short-lived TCP connections?

- 85% of TCP traffic are short-lived flows
- Connections ends while in slow-start phase
  - never enter congestion avoidance
  - steady-state model doesn’t apply
TCP Features

• Connection oriented
  – Explicit and acknowledged connection establishment
• Reliable stream exchange
  – every packet has sequence number
  – acknowledging the receipt of the right packet (usually delayed)
  – set retransmission timer for every packet sent
• Congestion control
### Slow Start and Congestion Avoidance

<table>
<thead>
<tr>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>If current congestion window (cwnd) is less than slow start threshold (ssthresh)</td>
<td>If (cwnd &lt; ssthresh)</td>
</tr>
<tr>
<td>TCP is in slow start phase, and increase the cwnd exponentially</td>
<td>cwnd = cwnd + 1;</td>
</tr>
<tr>
<td>Otherwise in congestion avoidance mode, and cwnd increases linearly</td>
<td>Else</td>
</tr>
<tr>
<td></td>
<td>cwnd += 1/cwnd;</td>
</tr>
</tbody>
</table>
A Typical TCP Connection
(No Loss Happens)
Steady State Model
- Assumptions

- Based on TCP Reno release from Berkeley
- High link speed
- Fixed packet size
  - Congestion window alone determines the send rate
Steady State Model
- Assumptions (Continued)

- Modeling dynamics of TCP in terms of “rounds”
  — starts when a window of packets is sent and ends when one or more acknowledgments are received
- Delayed acknowledgment algorithm applied
- Packet losses in accordance with bursty loss model
  — Packet losses are correlated in each round but independent between rounds
Steady State Model

Typical congestion window evolution

- Slow start phase
- Congestion avoidance phase
- Time-out phase
Let $M_i$ be the number of packets sent during the total time $S_i$:

$$M_i = Y_i^{ss} + \sum_{j=1}^{n_i} Y_{ij} + R_i$$

$$S_i = Z_i^{ss} + \sum_{j=1}^{n_i} A_{ij} + Z_i^{TO}$$

Assuming $(M, S)$ to be sequences of i.i.d. random variables, the send rate is:

$$B = \frac{E[M]}{E[S]}$$
Steady State Model

Considering \( \eta_i \) to be i.i.d. random variables and independent of \( Y_{ij} \), we have:

\[
B = \frac{E[Y^{ss}] + E[n]E[Y] + E[R]}{E[Z^{ss}] + E[n]E[A] + E[Z^{TO}]}
\]
Slow Start Phase

- Congestion window growth pattern is: \( cwnd_i = \left[ \frac{cwnd_{i-1}}{2} \right] + cwnd_{i-1} \)
- The total number of packets sent in first \( n \) rounds: \( Y_{n ss} = \sum_{i=1}^{n} cwnd_i \)

<table>
<thead>
<tr>
<th>Number of packets sent</th>
<th>( E[Y_{ss}] = \frac{E[W_{TD}]}{2} g^2 - 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time duration</td>
<td>( E[Z_{ss}] = \log \left( \frac{E[W_{TD}]}{2C_1} \right) \ast RTT )</td>
</tr>
</tbody>
</table>
### Congestion Avoidance Phase

#### Table

<table>
<thead>
<tr>
<th>Number of packets sent</th>
<th>[ E[Y] = \frac{E[X]}{2} \left( \frac{E[W^{TD}]}{2} + E[W^{TD}] - 1 \right) + E[\beta] ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time duration</td>
<td>[ E[A] = RTT(E[X] + 1) ]</td>
</tr>
</tbody>
</table>
Congestion Avoidance Phase (continued)

- Expected congestion window size:
  \[ E[W_{TD}] = \frac{2(b-2p)}{3} + \sqrt{\frac{4(bp+2(1-p^2))}{3bp}} + \left(\frac{2b-4p}{3b}\right)^2 \]

- Number of packets sent in the fast retransmit:
  \[ E[\beta] = (E[W_{TD}] - 1)(1 - p) \]
- Number of rounds in TDP:
  \[ E[X] = b\left(\frac{E[W_{TD}]}{2} + 1\right) \]
Time-outs Phase (continued)

• Padhye’s steady-state model use:
  \[ \frac{1}{E[w]} = E[1/w] \]

• Not so good approximation:
  \[ E[\left(\frac{1}{\sqrt{W}}\right)^2 \leq E[\left(\frac{1}{\sqrt{W}}\right)^2] E[\left(\sqrt{W}\right)^2] \]
  \[ \Rightarrow \quad \frac{1}{E[W]} \leq E[\frac{1}{W}] \]

• Better approximation:
  \[ E[\frac{1}{W}] \approx \frac{1}{E[W]} \left(1 + \frac{Var(W)}{E[W]^2}\right) \]
Probability of Packet Loss Resulting in time-out

\[ Q^{TD} = E[Q^{TD}(w)] \]

\[ = E[\min(1, \frac{3}{w})] \]

\[ = \min(1,3E[\frac{1}{W^{TD}}]) \]

\[ \approx \min(1, \frac{3\sqrt{3}}{E[W^{TD}]}) \]
Send Rate and Throughput

- **Send Rate:**
  — is the number of packets sent per second
- **Throughput:**
  — is the number of packets received per second

From Padhye’s model:

| Number of TDPs in a congestion avoidance phase | \( E[n] = \frac{1}{QTD} \) |
| Number of packets sent in the time-out phase | \( E[R] = \frac{1}{1-p} \) |
| Time spent in the time-out phase | \( E[Z^{TO}] = T \frac{f(p)}{0_{1-p}} \) |
Send Rate

\[
B = \begin{cases} 
\frac{E[W^{TD}]g^2}{2} - 2 + \frac{1}{Q^{TD}(E[W^{TD}])} \left( \frac{1-p}{p} + E[W^{TD}] \right) + \frac{1}{1-p} \\
(\log g \left( \frac{2C_1}{E[W^{TD}]} \right) + \frac{1}{Q^{TD}(E[W^{TD}])} \left( \frac{bE[W^{TD}]}{2} + b + 1 \right)RTT + \frac{f(p)T_Q}{1-p} 
\end{cases}
\]

When \( E[W^{TD}] < W_m \)

\[
\frac{Wmg^2}{2} - 2 + \frac{1}{Q^{TD}(W_m)} \left( \frac{1-p}{p} + W_m \right) + \frac{1}{1-p} \\
(\log g \left( \frac{2C_1}{W_m} \right)RTT + \frac{1}{Q^{TD}(W_m)} \left( \frac{b}{8} W_m + \frac{1-p}{pW_m} + 2 \right) + 1)RTT + \frac{f(p)T_Q}{1-p} 
\]

When \( E[W^{TD}] \geq W_m \)
Throughput

To obtain throughput, changes are needed:

\[ E[Y] \rightarrow E[Y'] = E[\alpha] + E[\beta] - 1 \]

The number of packets that have been sent in a TDP

\[ E[R] \rightarrow E[R'] = 1 \]

The expected number of packets sent in the time out phase

The number of packets that have been received in a TDP

The expected number of packets received in the time out phase
Throughput
(continued)

\[
H = \begin{cases} 
\frac{E[W_{TD}]}{2} g^2 - 2 + \frac{1}{Q_{TD}(E[W_{TD}])} \left( \frac{1-p}{p} + (E[W_{TD}]/2 - 1)(1-p) \right) + 1 \\
\left( \log g\left( \frac{E[W_{TD}]}{2C_1} \right) + \frac{1}{Q_{TD}(E[W_{TD}])} \left( \frac{bE[W_{TD}]}{2} + b + 1 \right)RTT \right) + \frac{f(p)T}{1-p} 
\end{cases}
\]

When \( E[W_{TD}] < W_m \)

\[
\frac{W_{mg}}{2} - 2 + \frac{1}{Q_{TD}(W_m)} \left( \frac{1-p}{p} + (W_m - 1)(1-p) \right) + 1 \\
\left( \log g\left( \frac{W_m}{2C_1}RTT \right) + \frac{1}{Q_{TD}(W_m)} \left( \frac{bW_m}{8} + \frac{1-p}{pW_m} + 1 \right)RTT \right) + \frac{f(p)T}{1-p} 
\]

When \( E[W_{TD}] \geq W_m \)
Comparison Example

The conditions are: RTT=200ms, MSS=536Bytes, $w_1=1$ segment, $T_0=1$s, $W_m=20$ segments
Model Validation by Simulations
Short-lived TCP Connection Model

1. Initial three-way-handshake connection
   — modeled by Cardwell’s paper
2. Initial slow start
   — same model used in steady state model
3. First loss
   — same analysis used for time-out phase
4. Subsequent losses
   — good approximation: Steady-state model
Short-lived TCP connection Model

- Time spent in initial slow start part:
  \[ E[n] = \begin{cases} 
  ([\log g(\frac{W_m}{C_1})] + \frac{1}{W_m}(E[Y_{init}] - g^2W_m - 2)) & \text{When } E[W_{init}] > W_m \\
  ([\log g(\frac{E[Y_{init}] + 2}{C_1})] - 2) & \text{When } E[W_{init}] \leq W_m 
  \end{cases} \]

- Time spent in the first loss part:
  \[ T_{loss} = (1-(1-p)^d)(Q_{init}E[Z^{\alpha}]+(1-Q_{init})E[z_i]) \]

- Time spent in the rest part:
  \[ T_{rest} = \frac{d - E[Y_{init}]}{H} = \frac{dp - (1-(1-p)^d)(1-p)}{p*H} \]
Short-lived TCP Connection Latency

\[ T_{latency} = E[T_{twhs}] + E[n]RTT + T_{loss} + T_{rest} + T_{delay} - \frac{RTT}{2} \]

- \( T_{delay} \): caused by delayed acknowledgment for the first packet which is characterized by mean of 100ms
- Only half of a round is needed to send the last window of packets, so deduct the half round trip time from the total latency.
Short-lived TCP model

→ Steady state model
Latency of Short-lived TCP Connection
Throughput vs. Loss Rate
(File Size = 2KB)
Throughput vs. Loss Rate
(File Size = 6KB)
Throughput vs. Loss Rate
(File Size = 11KB)
Comparison of the Average Error

<table>
<thead>
<tr>
<th>Loss Rate</th>
<th>P = 0</th>
<th>3×10⁻³ ~ 10⁻¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Size</td>
<td>0.5~26KB</td>
<td>2KB</td>
</tr>
<tr>
<td>[CSA00]</td>
<td>9.40%</td>
<td>4.08%</td>
</tr>
<tr>
<td>Proposed</td>
<td>5.83%</td>
<td>0.59%</td>
</tr>
</tbody>
</table>
Conclusions

• Propose new model for the slow start phase
  – Based on discrete equation
  – Using results from Fibonacci sequence
• Develop complete steady state model
  – Integrate slow start phase
  – Accurate time-out analysis
• Develop accurate short-lived TCP model
  – Using same analysis of slow start
  – New estimate time-out analysis
1. Future Work

- Considering effect of fast recovery
  — will help building a more accurate model

- Analyze effects of different loss models to TCP’s performance
  — help design different queuing methods

- Find probability distribution of latency
  — better than the expected value