TCP Congestion Control with a Misbehaving Receiver

ACM SIGCOMM CCR Oct 1999

Paper by: Stefan Savage, Neal Cardwell, David Wetherall, Tom Anderson

Discussion Leader: Christoph Jechlitschek

Presented by: Manfred Georg
Overview

- RFC 2581
- Attacks
  - ACK division
  - DupACK spoofing
  - Optimistic ACKing
- TCP Daytona
- Robust Protocol Design
- Conclusion
RFC 2581

- TCP Specifications
- Based on mutual trust
- Ambiguous
- Security holes
ACK division

- cwnd grows with each ACK received
- Attack:

  For each received packet, send many ACKs each for a piece of the packet.
ACK division

Sender

RTT

Data 1:1461

ACK 487

ACK 973

ACK 1461

Data 1461:2921

Data 2921:4381

Data 4381:5841

Data 5841:7301

Receiver
DupACK spoofing

- Fast Recovery
- One Packet sent for each DupACK
- Bounded by Receiver Window
DupACK spoofing
Optimistic ACKing

- Preemptively acknowledge packets
- Reduce RTT
- Conceal losses
- Possible data loss
Optimistic ACKing
TCP Daytona

- Modified Linux 2.2.10 TCP stack
- ACK division 24 lines
- DupACK spoofing 11 lines
- Optimistic ACKing 45 lines
TCP Daytona  ACK division

Sequence number (Bytes)

Data Segments
ACKs

Data Segments (normal)
ACKs (normal)

Time (sec)
TCP Daytona  Optimistic ACKing

Data Segments
ACKs

Data Segments (normal)
ACKs (normal)
## Vulnerable OS

<table>
<thead>
<tr>
<th></th>
<th>ACK Division</th>
<th>DupACK Spoofing</th>
<th>Optimistic Acks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 2.6</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Linux 2.0</td>
<td>Y</td>
<td>Y (N)</td>
<td>Y</td>
</tr>
<tr>
<td>Linux 2.2</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Windows NT4/95</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>FreeBSD 3.0</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DIGITAL Unix 4.0</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>IRIX 6.x</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>HP-UX 10.20</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>AIX 4.2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Robust Protocol Design

• Principle 2:

  The conditions for a message to be acted upon should be clearly set out.

• What is an ACK?

• Byte or segment granularity?
Robust Protocol Design

• Principle 1:
  
  Every message should say what it means: the interpretation of the message should depend only on its content.

• What is a DupACK?

• Meaning is context dependent

• Use nonce to establish context
Robust Protocol Design

• Principle 3:
  
  If sender identity is essential to the meaning of a message, explicitly include it in the message.

• Segment not required to create ACK

• Cumulative Nonce
Conclusion

- TCP grew in cooperation
- Internet uncooperative
- Security holes
- (Easy) fixes
  - Consistent granularity
  - Cumulative nonce
- Robust protocol design