TCP Hollywood: An Unordered, Time-Lined, TCP for Networked Multimedia Applications

Stephen McQuistin and Colin Perkins
University of Glasgow

Marwan Fayed
University of Stirling
Multimedia Applications

• Significant and growing percentage of all Internet traffic

• Sensitive to latency, as well as loss

• Must use either TCP or UDP, and TCP when UDP not available

Global Consumer Internet Traffic

- Internet video
- Other

Cisco Visual Networking Index 2014-2019
Multimedia Applications in TCP

Sending segments between a TCP sender and receiver, looking at interaction with applications

Sender

user

kernel

Network

kernel

user

Receiver

time
Multimedia Applications in TCP

- **Sender**
  - *user*
  - *kernel*

- **Network**
  - *buffers*
  - *time*

- **Receiver**
  - *kernel*
  - *user*

- $T_{framing}$
Multimedia Applications in TCP
Multimedia Applications in TCP
Multimedia Applications in TCP

Sender

Network

Receiver

user

kernel

kernel

user

time

7
Multimedia Applications in TCP

$T_{\text{playout}}$ reduces gaps in playback due to jitter.
Multimedia Applications in TCP

Sender

network

Receiver

network

user

kernel

time

Playout begins
Multimedia Applications in TCP

Sender

Network

Receiver

Segment lost
Multimedia Applications in TCP

Segment arrives, but can’t be delivered due to head-of-line blocking
Multimedia Applications in TCP
Multimedia Applications in TCP

\[ T_{\text{retransmit}} = 4 \times T_{\text{framing}} + T_{\text{rtt}} \]
Multimedia Applications in TCP

Standard TCP results in three segments being delivered too late to be played out.
Introducing TCP Hollywood

- Remove latency introduced by standard TCP

- Looks like standard TCP on the wire, to maintain deployability
# Delivery Models

<table>
<thead>
<tr>
<th>Standard TCP</th>
<th>TCP Hollywood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordered delivery</td>
<td>Unordered delivery</td>
</tr>
</tbody>
</table>
## Delivery Models

<table>
<thead>
<tr>
<th>Standard TCP</th>
<th>TCP Hollywood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordered delivery</td>
<td>Unordered delivery</td>
</tr>
<tr>
<td>Reliability</td>
<td>Partial reliability, based on timelines and dependencies</td>
</tr>
</tbody>
</table>

- **Standard TCP**
  - Ordered delivery
  - Reliability

- **TCP Hollywood**
  - Unordered delivery
  - Partial reliability, based on timelines and dependencies
## Delivery Models

<table>
<thead>
<tr>
<th>Standard TCP</th>
<th>TCP Hollywood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordered delivery</td>
<td>Unordered delivery</td>
</tr>
<tr>
<td>Reliability</td>
<td>Partial reliability, based on timelines and dependencies</td>
</tr>
<tr>
<td>Byte stream</td>
<td>Message-oriented, with sub-streams</td>
</tr>
</tbody>
</table>
# Delivery Models

<table>
<thead>
<tr>
<th>Standard TCP</th>
<th>TCP Hollywood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordered delivery</td>
<td>Unordered delivery</td>
</tr>
<tr>
<td>Reliability</td>
<td>Partial reliability, based on timelines and dependencies</td>
</tr>
<tr>
<td>Byte stream</td>
<td>Message-oriented, with sub-streams</td>
</tr>
<tr>
<td>Other services (e.g., congestion control)</td>
<td></td>
</tr>
</tbody>
</table>
Architecture

- Functionality split between user-space intermediary layer, and kernel extensions
- Intermediary layer works over either standard TCP or TCP Hollywood
- Supports partial deployments
Intermediary Layer

- Encodes and frames messages at the sender
- Decodes and reassembles messages at the receiver
Kernel Extensions

- Enables partial reliability at the sender

- Delivers segments in arrival order at the receiver
Unordered message delivery

- Nagle’s algorithm disabled

- Applications pass messages with optional metadata: deadlines and dependencies

- Message data encoded to escape zero bytes for use as framing markers — *transparent channel*

- Segments passed to intermediary layer as they arrive, with ACKs generated as under standard TCP
Partial reliability

• Messages might expire: estimated to arrive too late, or depend on an undelivered message

• Expired messages aren’t retransmitted - next live message sent instead as an *inconsistent retransmission*

• Same length and TCP sequence number as standard TCP — payload is different
When are inconsistent retransmissions useful?
Feasibility Region

$T_{\text{playout}}$: Duration of the playout delay at the receiver.
Feasibility Region

Network round-trip time
Feasibility Region

Plotting the region of feasible values of $T_{\text{playout}}$ across round-trip times
Feasibility Region

$T_{playout}$

$T_{framing}$

Duration of media in each message

$T_{rtt}$
Feasibility Region

Message needs to be decoded before being played out
Feasibility Region

\[ T_{\text{playout}} \]

\[ T_{\text{framing}} \]

\[ T_{\text{rtt}} / 2 \]

Application delay bound

\[ T_{\text{max}} - T_{\text{framing}} - T_{\text{rtt}} / 2 \]
Feasibility Region

\[ T_{\text{framing}} + \frac{T_{\text{rtt}}}{2} \leq T_{\text{framing}} \leq T_{\text{max}} - T_{\text{framing}} - \frac{T_{\text{rtt}}}{2} \]
Feasibility Region

\[ T_{\text{playout}} = T_{\text{rtt}} + 4 \cdot T_{\text{framing}} - T_{\text{max}} - T_{\text{framing}} - \frac{T_{\text{rtt}}}{2} \]

\[ T_{\text{rexmit}} \]
Feasibility Region

Standard TCP retransmissions are useful
Feasibility Region

Standard TCP retransmissions arrive too late to be used
Where TCP Hollywood Helps

TCP Hollywood sends inconsistent retransmissions instead

Standard TCP retransmissions arrive too late to be used
Example

IPTV application, using DASH for delivery

$T_{\text{max}} = 1 \text{ second}$
Standard TCP retransmissions effectively useless when $T_{rtt}$ higher than 110ms
Eliminating Head of Line Blocking

\[ T_{\text{rtt}} + 4 \cdot T_{\text{framing}} \]

\[ T_{\text{playout}} \]

\[ T_{\text{HoL}} \]

Duration of head-of-line blocked segments
Eliminating Head of Line Blocking

\[ T_{rtt} + 4 \cdot T_{framing} \]

Segments that arrive after a loss are queued
Eliminating Head of Line Blocking

\[ T_{rtt} + 4 \cdot T_{framing} \]

**Tplayout**

**T\textsubscript{HoL}**

Retransmission of lost segment won’t arrive in time to be played out
Eliminating Head of Line Blocking

\[ T_{rtt} + 4 \cdot T_{framing} \]

\[ T_{playout} \]

\[ T_{HoL} \]

Arrival time of retransmission determines head-of-line blocking delay
Eliminating Head of Line Blocking

\[ T_{\text{rtt}} + 4 \cdot T_{\text{framing}} \]

\[ T_{\text{playout}} \quad T_{\text{HoL}} \]

Head of line blocking is eliminated in TCP Hollywood
Deployability

- Inconsistent retransmissions are the only wire-visible modification vs. standard TCP — same TCP sequence number, different payload

- Middleboxes performing payload inspection may interpret the behaviour as an attack — man on the side
TCP Hollywood is deployable

- Experiments between TCP Hollywood server, and 14 clients in the UK
- 8 fixed-line residential ISPs, 4 mobile operators - all major UK ISPs
- Test on two ports to check if behaviour is different for HTTP traffic
TCP Hollywood is deployable

- Inconsistent retransmissions delivered successfully
- Segments cached by middlebox, so original delivered instead — performance no worse than standard TCP
- Safe failure mode for TCP Hollywood

<table>
<thead>
<tr>
<th>ISP</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed-line</td>
<td></td>
</tr>
<tr>
<td>Andrews &amp; Arnold</td>
<td>80</td>
</tr>
<tr>
<td>BT</td>
<td>4001</td>
</tr>
<tr>
<td>Demon</td>
<td></td>
</tr>
<tr>
<td>EE</td>
<td></td>
</tr>
<tr>
<td>Eclipse</td>
<td></td>
</tr>
<tr>
<td>Sky</td>
<td></td>
</tr>
<tr>
<td>TalkTalk</td>
<td></td>
</tr>
<tr>
<td>Virgin Media</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE</td>
</tr>
<tr>
<td>O2</td>
</tr>
<tr>
<td>Three</td>
</tr>
<tr>
<td>Vodafone</td>
</tr>
</tbody>
</table>
TCP Hollywood

- Unordered, partially reliable message-oriented transport protocol
- Wire-compatible with TCP
- Analysis shows when TCP Hollywood helps applications
- Deployable across all major UK fixed-line and cellular ISPs

“Hollywood Sign”, Katie Harbath - CC BY-NC-SA 2.0
flickr.com/photos/katieharbath/5529042403