SDN-based Application-Aware Networking on the Example of YouTube Video Streaming

Michael Jarschel, Florian Wamser, Thomas Höhn, Thomas Zinner, Phuoc Tran-Gia

www3.informatik.uni-wuerzburg.de
Motivation

Use application state information to optimize the user experience and resource management.
Scenario

- Leveraging SDN for network resource management
- Dynamic shift of application flows between available channels to enhance quality of critical applications
- Specific Example: Concurrent download and YouTube flows
  ➔ Maintaining a good YouTube quality
Pre-Buffered Playtime as QoE-Metric with YoMo

- YoMo (YouTube Monitor) estimates the pre-buffered playtime
- Used as:
  - Control input parameter for the application-aware approach
  - Quality indicator to evaluate the management algorithms
Reference Testbed

- Controller running “switch” application
  → only one link between the two OpenFlow switches usable
- Maximum throughput: 10 Mbit/s
# Resource Allocation Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Required Information</th>
<th>Source of Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round robin</td>
<td>Number of links</td>
<td>OF controller topology manager</td>
</tr>
<tr>
<td>Bandwidth-based</td>
<td>Used bandwidth of flows</td>
<td>OpenFlow statistics</td>
</tr>
<tr>
<td>Deep Packet Inspection</td>
<td>Type of traffic/application</td>
<td>DPI software</td>
</tr>
<tr>
<td>Application awareness</td>
<td>Application state</td>
<td>Application</td>
</tr>
</tbody>
</table>

Increasing Complexity
Experiment Procedure

- 0s: Begin of experiment; YouTube video is started
- 60s: Start of interfering traffic
- Iperf-generated TCP flows are used as interfering traffic
- 60s+x: Interfering traffic flows start with a 1s interval
- 420s: End of experiment
Reference Case with 5 TCP Flows

- Maximum throughput of 10 Mbit/s
- The YouTube flows do not have sufficient bandwidth available
  ➔ Stalling cannot be prevented
Bandwidth-Based Approach

- Maximum throughput: 50 Mbit/s
- Using flow statistics to calculate throughput
- Algorithm:
  - Select link based on throughput information
  - Switch flows with the highest throughput to another link
Flows are distributed among the different links
Influence of the interfering traffic is visible
The scheduler can maintain the pre-buffered playtime
Bandwidth-Based Allocation with 20 TCP Flows

- Flows are distributed among the different links
- Too many flows in the network
  ⇒ The YouTube-video stalls
Application-Awareness Approach

- Application notifies the controller about incoming application traffic
- Information about the YouTube-stream is obtained through YoMo

Algorithm:
- Classification of the flows into different priority classes
- The highest flow priority on a link determines the allowed number of flows
The maximum possible throughput is reached
Critical threshold $t_C$ at 20 s pre-buffered playtime
Regular threshold $t_R$ at 35 s pre-buffered playtime
Shift of interfering flows, if the buffer falls below the threshold
No stalling of the YouTube video
Efficiency of the Application-Awareness Approach

- Without interfering traffic the typical YouTube behavior is observed.
- With 25 TCP flows an influence is visible.
- With 50 TCP flows an intervention of the scheduler is necessary.
- For 75 and 100 interfering TCP flows a similar behavior is visible.
- Pre-buffered playtime can be maintained.
Conclusion

- Dynamic shift of application flows between available channels to enhance quality of critical applications

- Implemented SDN network resource management for YouTube using Northbound-API

- Application-aware approach outperforms conventional mechanisms in terms of QoE

- Future Work:
  - Trade-offs between multiple critical applications
  - Other QoS management mechanisms, e.g. OF 1.3 flow meters
  - More types and sources of application (state) information
  - Identification of a suitable standard Northbound-API realization
Questions and Comments?