Enabling Traffic Management without DPI

DPI Is Dead, Long Live Traffic Management

Mirja Kühlewind
Dirk Kutscher
Brian Trammell

Managing Radio Networks in an Encrypted World (MaRNEW) Workshop
Atlanta, September 24/25 2015
„Cooperative Traffic Management“

- Common denominator for many workshop contributions
- „Extend current connection-based encryption approaches by integrating middleboxes into the loop“

- Difficult to do right and to manage reliably
  - Trust?
  - Robustness?
  - Performance?
Previously

**UPCON – Solution outline**

1. **Detect user plane congestion** in Radio Access, Backhaul or Core Network entities

2. **Apply different traffic handling / QoS schemes** to user plane traffic, based on Subscriber profile, Application type, Content type

3. **Develop adequate traffic scheduling and traffic engineering mechanisms**, such as per-user or per-flow queuing, application-aware QoE scheduling, flow-based handover, media compression, etc.

4. **Enable policy-based control for operators to flexibly configure** the traffic the network behavior under handling mechanisms

- Operator-centric approach
- Interaction with transport protocols unclear
- Traffic engineering function requires massive DPI (in the presence of encryption)
Currently Proposed

**Throughput Guidance Solution Architecture**

- Application-provider-centric approach
- Conveying information about estimated current base station capacity to TCP senders
- Only works with TCP
- Implemented as TCP Option – interaction with middleboxes?
- Very specific – generality?
Thesis: Two Main Concerns

1. Meaningful Capacity Sharing
   - Enabling low-latency communication in the presence of high network utilization
   - Incentivize application/sender adaptivity

2. Reacting correctly to (wireless) link layer conditions
   - Distinguish from congestion events
Traffic Management Requirements

• **Application-independence**: permission-less innovation
  – No DPI required
  – Should work with all (future) application types
  – Should work with all (future) transport protocols

• **Efficiency and Effectiveness**
  – Should interact well with transport
  – ... Without complex management frameworks

• **Generality**
  – Should not be limited to specific systems or configurations

• **Privacy-friendly**
  – In-band cooperation tools should only expose essential traffic management information
Congestion Exposure Principle

TCP feedback

sender

network

receiver

congestion marking / loss

ConEx
Lessons Learned from ConEx

- Congestion exposure: means to incentivize application/sender adaptivity
- Mechanism vs policy
- Making current congestion visible to network and endpoints may not be enough
- IP not designed for in-band management
- Authentication needed
## Architecture

<table>
<thead>
<tr>
<th>App</th>
<th>User data. Definitely none of the path’s business.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport'</td>
<td>Stuff the path can’t see, but ensures the network doesn’t burn down</td>
</tr>
</tbody>
</table>
| SPUD    | Stuff the applications/endpoints are willing to share with the path  
          Stuff the applications/endpoints might be willing to hear from the path |
| UDP     | Stuff we can get across the the existing Internet |

Stuff app devs can’t change:
Socket APIs, kernel, middleboxes
Extensible and Efficient Traffic Management

• More flexible traffic management transport
  – Allow for generally encrypted traffic
  – SPUD prototype as a platform for experiments
  – Design for flexibility – without ignoring efficiency requirements
  – Finding minimum set of information to expose (PII issue)

• Re-think capacity sharing
  – Congestion accountability != TCP fairness
  – Incentivizing adaptability and immediate response to congestion
  – Support for low-latency: DCTCP-like
  – Simple QoS – distinguish interactive real-time from rest of traffic at bottlenecks
  – Additional signaling for non-congestion-induced events (wireless)
  – Hop-by-hop optimization and end-to-end control loops