Path Awareness and Selection in the Socket Intents prototype

Theresa Enghardt
TU Berlin
theresa@inet.tu-berlin.de
Scenario: Multiple paths

Multiple paths via different access networks
- Laptop can use WiFi or cellular
- WiFi usually default, but not always better¹

¹ Deng et al.: “WiFi, LTE, or Both? Measuring Multi-Homed Wireless Internet Performance” (2014)

→ Pick the better one? Use both?
Socket API

Vanilla BSD sockets:
- Connections "look the same"
- No information about paths

→ use **default path** based on system policy
Socket Intents

Intents:

What the application \textit{wants}, \textit{knows}, \textit{prefers}, or \textit{assumes} about its traffic (connection or message)

$\rightarrow$ What to optimize for

See our \url{draft-tiesel-taps-socketintents}
Socket Intents

Intents:
• Traffic Category
• Size to be received
• Bitrate to send
• Timeliness
• Cost preferences

See our draft-tiesel-taps-socketintents
Socket Intents

Connection 1:

- Traffic Category: Query
- Size to be Received: 10 KB

See our draft-tiesel-taps-socketintents
Socket Intents

Path property estimates:
• Median Round Trip Time (RTT)
• Maximum bitrate
• WiFi utilization
• ...
Socket Intents

Path 1: LTE
- RTT = 100 ms
- Bandwidth = 20 Mbit/s

Path 2: WiFi
- RTT = 10 ms
- Bandwidth = 2 Mbit/s
Socket Intents

Path Selection Policy:

- “Use path with shorter completion time”
- LTE: \(\approx 200\) ms
- WiFi: \(\approx 20\) ms

→ Use WiFi
Socket Intents

Connection 2:
- Traffic Category: **Bulk**
- Size to be Received: **500 KB**

**LTE**
- 100ms, 20 Mbit/s

**WiFi**
- 10ms, 2 Mbit/s
Socket Intents

Path Selection Policy:
  - “Use path with shorter completion time”
  - LTE: ≈ 225 ms
  - WiFi: ≈ 450 ms

→ **Use LTE**

... or both (MPTCP)
Socket Intents Prototype

1. Connect Request With Intents

4. Bind to address A, connect

2. “Which path to use?”

3. “Use Path 1 for this connection, with local address A”

- **Socket Intents Library:** Augmented Socket API
- **Multi Access Manager:**
  - Standalone daemon with policy modules
  - Gathers current performance estimates
  - Chooses path and local address

See our draft-tiesel-taps-socketintents-bsdsockets and code https://github.com/fg-inet/socket-intents
Paths

Multi Access Manager:

- Detects locally configured interfaces with their prefixes and addresses
- Gathers statistics on them passively, based on current and past traffic

<table>
<thead>
<tr>
<th></th>
<th>Path 1</th>
<th>Path 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local prefix</td>
<td>a:a:a:a::a/64</td>
<td>1.2.3.4/24</td>
</tr>
<tr>
<td></td>
<td>b:b:b:b::b/64</td>
<td>5.6.7.8/24</td>
</tr>
<tr>
<td>Minimum RTT</td>
<td>95 ms</td>
<td>105 ms</td>
</tr>
<tr>
<td></td>
<td>8 ms</td>
<td>10.5 ms</td>
</tr>
<tr>
<td>Maximum Bitrate</td>
<td>18.9 Mbit/s</td>
<td>1.8 Mbit/s</td>
</tr>
<tr>
<td>Utilization</td>
<td>N/A</td>
<td>2%</td>
</tr>
</tbody>
</table>
Path Bitrate

- Bitrate per interface
  - Read interface counter every \( n \) ms\(^1\)
  - \( \text{bitrate}_{\text{current}} = \frac{\text{counter}_{\text{current}} - \text{counter}_{\text{prev}}}{n} \)
  - \( \text{bitrate}_{\text{max}} \): Maximum within the last \( m \) minutes\(^2\)
    - estimate of bandwidth of the path
- Assumes the bottleneck on each path is within the first few hops

\(^1\) 100 ms works for us
\(^2\) 5 minutes works for us
Path RTT

- RTT per prefix
  - Linux kernel keeps list of current TCP connections
  - Each TCP connection has a current Smoothed RTT (SRTT)
  - Query SRTTs of all connections over prefix every $n$ ms
  - Compute current mean, median or SRTTs
  - If no current values, retain values for up to $m$ minutes
  - Compute minimum of the last $m$ minutes
- We expect the first hop or first few hops to dominate latency
Radio properties on path

• For WiFi:
  - Current Received Signal Strength
  - Last observed modulation bitrates
  - Utilization: QBSS Information Element from Beacon frames
• Other wireless technologies possible, but hard
Current and Future Work

- Show page load time reduction for web browsing
- More path selection policies
- More path properties
  - RTT variation
  - Packet loss
  - Information from the network
Summary

- **Socket Intents:**
  - Application provides hints on Connection (or Message…)
  - We know what to optimize for
- **Path properties:**
  - Socket Intents prototype gathers them locally
  - Observed median SRTT, maximum bitrate, WiFi Utilization
- **Path selection:**
  - E.g. use Path with shortest expected completion time
  - Other optimization possible, e.g. for cost

See our [draft-tiesel-taps-socketintents-bsdsockets](https://github.com/fg-inet/socket-intents) and code [https://github.com/fg-inet/socket-intents](https://github.com/fg-inet/socket-intents)