Application-Network Integration: Possibilities, Challenges, and Possible Next Steps

Y. Richard Yang

IETF 106
November 20, 2019
Singapore
Application-Aware (AA) Networks
Can Have Diverse AA Capabilities,
Requiring Different Support

<table>
<thead>
<tr>
<th>Example Capability</th>
<th>Possible Support &amp; Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treat each packet the same (aka not AA)</td>
<td></td>
</tr>
<tr>
<td><strong>Aware at app-level granularity</strong></td>
<td></td>
</tr>
<tr>
<td>Create different networks/slices (e.g., voice vs data networks)</td>
<td>IP, access; SDN; scheduling</td>
</tr>
<tr>
<td>Identify packets by ports (e.g., ACL)</td>
<td>Packet header port; scheduling</td>
</tr>
<tr>
<td><strong>Aware of sub-app granularity</strong></td>
<td></td>
</tr>
<tr>
<td>Scheduling each packet according to app-level deadline (e.g., fastpass’14)</td>
<td>Custom packet header; scheduling</td>
</tr>
<tr>
<td>Distinguish application-level structures (e.g., I frame vs P frame)</td>
<td></td>
</tr>
<tr>
<td>Fancy: Co-flow scheduling (e.g., VARYS’14, AALO’15)</td>
<td>Network state; scheduling</td>
</tr>
<tr>
<td><strong>Aware of cross-app/protocol dependency</strong></td>
<td></td>
</tr>
<tr>
<td>Fancy: identify full dependency (e.g., application level dependency such as DNS-&gt;handshake-&gt;...)</td>
<td>Network state; scheduling</td>
</tr>
</tbody>
</table>
Network-Aware (NA) Applications Can Have Diverse NA Capabilities, Requiring Different Network Information/Support

<table>
<thead>
<tr>
<th>Example Capability</th>
<th>Support &amp; Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer time selection</td>
<td>Network state in time; can delay</td>
</tr>
<tr>
<td>Server direction</td>
<td>Path properties from client to potential servers; has multiple servers</td>
</tr>
<tr>
<td>Rate adaptation</td>
<td>None/None/ECN/INT</td>
</tr>
<tr>
<td>CC, reacting to loss/delay/ECN bit/INT (e.g., HPCC’19)</td>
<td></td>
</tr>
<tr>
<td>Adaptive streaming</td>
<td></td>
</tr>
<tr>
<td>Lower-than-best-effort (e.g., LEDBAT)</td>
<td></td>
</tr>
<tr>
<td>Multi-path TCP</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Despite Broad Capabilities, 
A Simple Gap Example

• Collaborative, distributed, exa-scale data sciences [1]
  – Applications: LHC, LIGO, LSST, EHT ...
  – Services
    • Time-Block-Maximum Bandwidth
      – Application asks for a specific time block and would like to know (or provision) the maximum bandwidth available for a specific time period.
    • Bandwidth-Sliding-Window
      – Application asks for a specific bandwidth and duration and provides an acceptable time window. For example, a request may be for 40 Gbps for a 10-hour time window, sometime in the next 3 days.
    • Time-Bandwidth-Product (TBP)
      – Application asks for “8 hours of transfer at 10Gbps” representing a TBP of 36 TBytes. The user also specifies an acceptable time window, and other options such as “prefer the highest bandwidth rate available”, or the lowest.
  – Protocol outcomes
    • Immediate provision
    • What is Possible?
    • Negotiation

Basic Challenge: Architecture

- Applications and networks can be designed with different objectives
  - Application: optimizes application’s utility
  - Network: optimizes network’s utility, enforces fairness, ...

- The end-to-end principle which mostly argues for the minimization of AA-networking
### A Simple Example

**Application objective:** optimize total throughput

- Using a fluid model*, we can derive that: optimizing throughput $\Rightarrow$ maximizing up/down link capacity usage

$$\begin{align*}
\max_i \sum_j t_{ij} \\
\text{s.t. } \forall i, \sum_{j \neq i} t_{ij} \leq u_i, \\
\forall i, \sum_{j \neq i} t_{ji} \leq d_i, \\
\forall i \neq j, t_{ij} \geq 0
\end{align*}$$

**Network objective:** minimize maximum link utilization (MLU)

- $b_e$: background traffic volume on link $e$
- $c_e$: capacity of link $e$
- $I_e(i,j) = 1$ if link $e$ is on the route from $i$ to $j$
- $t_k$: a traffic demand matrix $\{t_{ij}^k\}$ for each pair of nodes $(i,j)$

$$\begin{align*}
\min_{e \in E} \max_k \left( b_e + \sum_{i \neq j} t_{ij}^k I_e(i,j) / c_e \right)
\end{align*}$$
A Simple Example: System Formulation

• Combine the objectives of network and applications

\[
\min_{e \in E} \max \left( b_e + \sum_k \sum_{i \neq j} t_{ij}^k I_e(i, j) \right) / c_e
\]

s.t., for any \( k \),

\[
T^1_k \quad \text{and} \quad T^k
\]

\[
\max \sum_i \sum_{j \neq i} t_{ij}^k \\
\text{s.t.} \forall i, \sum_{j \neq i} t_{ij}^k \leq u_i^k, \\
\forall i, \sum_{j \neq i} t_{ji}^k \leq d_i^k, \\
\forall i \neq j, t_{ij}^k \geq 0
\]
Why Hard: Constraints Couple Net/App

\[
\min_{\forall k: t^k \in T^k} \max_{e \in E} \frac{b_e + \sum_{k} \sum_{i \neq j} t_{ij}^k I_e(i, j)}{c_e}
\]

Constraints couple network/applications together!
Solution Architecture:
Decouple Network and Application

\[
\min \max_{\forall k: t^k \in T^k} \max_{e \in E} (b_e + \sum_k \sum_{i \neq j} t^k_{ij} I_e(i, j)) / c_e
\]

- With dual variable \(p_e (\geq 0)\) for the inequality of each link \(e\), and dual finite condition
  \[
  \sum_e p_e c_e = 1
  \]

\[
D(\{p_e\}) = \min_{\forall k: t^k \in T^k} \sum_e p_e (b_e + \sum_k t^k_{ij}) = \sum_e p_e b_e + \sum_k \min_{i \neq j} \sum_{i \neq j} p_{ij} t^k_{ij}
\]
Bigger Picture

- The interface between applications and network is the dual variables \( \{ p_{ij} \} \)

\[
p_{ij} = \sum_{e \text{ on route } i \rightarrow j} p_e
\]
From Simple Example to Real Design

- At a high level, one may indeed consider both ALTO (estimation) and (per-packet) INT info as \( p_{ij} \)
- Multiple steps to make progress
  - A systematic study of existing NAA capabilities and their support requirements
  - A systematic study of existing AAN capabilities and their support requirements
  - Systematic design of network and application abstraction spaces