Dynamic Traffic Engineering for Future IP Networks

Ivan Gojmerac, Thomas Ziegler, Fabio Ricciato, Peter Reichl
Telecommunications Research Center Vienna (ftw.)

4. Würzburger Workshop
"IP Netzmanagement, IP Netzplanung und Optimierung"

Würzburg, 27 July 2004
Outline

- Introduction to traffic engineering
- Adaptive Multi-Path (AMP) algorithm
- Performance evaluation and results
- Summary and outlook
What is “Traffic Engineering” (TE)?

- Traffic engineering is defined as performance optimization of operational networks (IETF)
  - Consider the traffic at the macroscopic level
  - Consider the network as a set of *limited* resources
    - Transmission bandwidth, switching throughput

- Traffic engineering tries to optimally match *traffic demands* with the available *network* resources by acting on *routing*
Traffic Engineering in IP Networks

- Traffic engineering methods for IP networks:
  - Link weight optimization in native IP networks
  - Optimization of Multi-Protocol Label Switched (MPLS) networks
  - Algorithmic approaches (dynamic routing in the ARPAnet, OMP)
Example of Connection-Less TE: Link Weight Optimization
Example of Connection-Oriented TE: Explicit-Routing Optimization

Traffic Demands

Network

Optimization...

Set of Explicit Routes for Virtual Pipes
Traffic Engineering in IP Networks

- Existing traffic engineering methods have important disadvantages:
  - MPLS and link weight optimization require additional network management
  - Unpredictable signaling overhead with Optimized Multi-Path (OMP)

- Our objective:
  - Autonomous and continuous load distribution in the network
  - Low overhead in terms of memory and bandwidth consumption

- Proposal: Adaptive Multi-Path Algorithm (AMP)
Current IP Routing

Node A — Node B — Node D — Node F

Node B — Node C — Node E — Node G

Congestion

Equal-Cost Multi-Path (ECMP)
AMP – Basic Operation

Backpressure Messages
AMP – Signaling

Node $Y_0$ → $X$ → $Y_1$ → $X$ → $Y_2$ → $X$ → $Y_3$→ $X$

Upstream BM
Downstream traffic
AMP – Signaling

- **Upstream BM**
- **Downstream traffic**

![Diagram]

- Node \(Y_0\)
- Node \(X\)
- Node \(Y_1\)
- Node \(Y_2\)
- Node \(Y_3\)

Edges:
- \(BM_{X \rightarrow Y_0}\)
- \(BM_{Y_1 \rightarrow X}\)
- \(BM_{Y_2 \rightarrow X}\)
- \(BM_{Y_3 \rightarrow X}\)

Upstream BM
Downstream traffic

Load
AMP – Signaling

\[ BM_{X \rightarrow Y_0} = f(Load_{XY_1}, \ldots, Load_{XY_n}, BM_{Y_1 \rightarrow X}, \ldots, BM_{Y_n \rightarrow X}) \]

Quasi-recursive structure of backpressure messages

\[\Rightarrow\]

GLOBAL PROPAGATION OF LOAD INFORMATION THROUGH LOCAL EXCHANGE OF SIGNALING MESSAGES
AMP – Signaling

\[ BM_{X \rightarrow Y_0} = f ( \text{Load}_{XY_1}, \ldots, \text{Load}_{XY_n}, BM_{Y_1 \rightarrow X}, \ldots, BM_{Y_n \rightarrow X} ) \]

Summarization of the number of parameters
AMP – Signaling

One parameter per link: \( g_i = \max (Load_{XY_i}, BM_{Y_i \rightarrow X}) \)
AMP – Signaling

\[ BM_{X \rightarrow Y_0} = f(\text{Load}_{XY_1}, \ldots, \text{Load}_{XY_n}, BM_{Y_1 \rightarrow X}, \ldots, BM_{Y_n \rightarrow X}) \]

Reduction of the number of parameters

\[ g_i = \max(\text{Load}_{XY_i}, BM_{Y_i \rightarrow X}) \]

\[ BM_{X \rightarrow Y_0} = f(g_1, g_2, \ldots, g_n) \]
AMP – Signaling

In X

Out X

Node $Y_0$  

Node $X$  

Node $Y_1$  

Node $Y_2$  

Node $Y_3$  

Upstream BM

Downstream traffic

In/Out Matrix in $X$

<table>
<thead>
<tr>
<th>In $X$</th>
<th>$\rightarrow Y_0$</th>
<th>$\rightarrow Y_1$</th>
<th>$\rightarrow Y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_0 \rightarrow$</td>
<td>$x$</td>
<td>160</td>
<td>120</td>
</tr>
<tr>
<td>$Y_1 \rightarrow$</td>
<td>490</td>
<td>$x$</td>
<td>230</td>
</tr>
<tr>
<td>$Y_2 \rightarrow$</td>
<td>830</td>
<td>120</td>
<td>$x$</td>
</tr>
</tbody>
</table>
AMP – Signaling

\[ BM_{X \rightarrow Y_0} = f\left(\text{Load}_{XY_1}, \ldots, \text{Load}_{XY_n}, BM_{Y_1 \rightarrow X}, \ldots, BM_{Y_n \rightarrow X}\right) \]

Reduction of the number of parameters

\[ g_i = \max\left(\text{Load}_{XY_i}, BM_{Y_i \rightarrow X}\right) \]

\[ BM_{X \rightarrow Y_0} = f\left(g_1, g_2, \ldots, g_n\right) \]

\[ = \sum_{Y_i \in \Omega_X \setminus Y_0} \frac{\beta_{Y_0 XY_i}}{\beta_{XY_i}} \cdot g_i \]

weights for congestion contributions
AMP Performance Evaluation

- Implementation of AMP in Network Simulator (ns-2)
- Simulated topology:
  - AT&T-US Network of 27 nodes and 47 links
  - Link capacities of 2.4 and 9.6 Gbit/s (scaled down to 15 and 60 Mbit/s in our simulations)
- Simulated traffic:
  - Web traffic according SURGE model
  - Traffic distribution according to the gravity model
  - Linear scaling of the number of Web users
AMP Performance Evaluation – Average Web Page Response Time

- Web page response time most important metric from the user’s perspective
- Significant reductions in Web page response times throughout investigated scenarios (up to 43%)

- SPR – Shortest Path Routing
- ECMP – Equal-Cost Multi-Path Routing
AMP Performance Evaluation – Total TCP Goodput

- Improved efficiency of resource utilization
- Total TCP goodput consistently higher with AMP compared to SPR and ECMP in our simulations (improvements of up to 28%)
AMP Performance Evaluation – Average CoVs of Link Load

- Similar average Coefficient of Variation (CoVs) of all link loads for the three routing strategies

⇒ stability of AMP load balancing
AMP Performance Evaluation
AMP Performance Evaluation – Average Web Page Response Time

Number of Web Users

- SPR
- ECMP
- AMP
AMP Performance Evaluation – Total TCP Goodput
Summary & Outlook

- **AMP Summary:**
  - Load balancing within the framework of routing
  - No management overhead, minimal signaling overhead
  - Implementation in Network Simulator (ns-2)
  - Significant performance improvements

- **Future research:**
  - AMP and network resilience
  - AMP fluid simulation
Thank you for your attention!

gojmerac@ftw.at
The goal of the load balancing mechanism in every node is to equalize the values of $g$ on all output links.
AMP – Load Balancing

- In order to avoid packet disordering:
  - => the unit for load balancing is a microflow aggregate
  - => packets are assigned to an aggregate by applying a CRC-16 hash-function on their source and destination IP addresses

- The CRC-16 solution space [0, 65535] is divided among the viable next hops

\[
\begin{align*}
161.53.101.8 & \quad 173.42.78.55 \\
\end{align*}
\]

\[\text{CRC-16} \quad 13217\]
AMP – Load Balancing

- Example routing table in Node B – the hash-space boundaries are defined for every reachable destination

<table>
<thead>
<tr>
<th>Destinations (in Node B)</th>
<th>Next hop: Node A</th>
<th>Next hop: Node D</th>
<th>Next hop: Node E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node A</td>
<td>[0 – 65535] (ALL PACKETS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node C</td>
<td>[0 – 23723]</td>
<td></td>
<td>[23724 – 65535]</td>
</tr>
<tr>
<td>Node D</td>
<td></td>
<td>[0 – 65535] (ALL PACKETS)</td>
<td></td>
</tr>
<tr>
<td>Node E</td>
<td></td>
<td></td>
<td>[0 – 65535] (ALL PACKETS)</td>
</tr>
<tr>
<td>Node F</td>
<td></td>
<td>[0 – 34447]</td>
<td>[34448 – 65535]</td>
</tr>
<tr>
<td>Node G</td>
<td></td>
<td>[0 – 52142]</td>
<td>[52143 – 65535]</td>
</tr>
</tbody>
</table>
AMP – Load Balancing

- Conservative load balancing mechanism – the size of load adjustment steps is changed dynamically.
Publications
