Let It Flow
Resilient Asymmetric Load Balancing with Flowlet Switching

Erico Vanini*, Rong Pan*, Mohammad Alizadeh†, Parvin Taheri*, Tom Edsall*
Load Balancing in Data Centers

- **Goal:** Avoid congestion hotspots
- Active research area
- Solved for symmetric topologies
- Still open question in asymmetric scenarios
Asymmetry Is Common in Practice

Link Failure
Asymmetry Is Common in Practice

Imbalanced striping: # of ports indivisible by # of switches in other tier

Dealing with Asymmetry

Erico Vanini – CISCO
Dealing with Asymmetry

Erico Vanini – CISCO
Dealing with Asymmetry

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Thrput</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECMP-WCMP (Local Stateless)</td>
<td>65G</td>
</tr>
<tr>
<td>Congestion-Aware</td>
<td></td>
</tr>
</tbody>
</table>
Handling asymmetry needs path congestion information, which varies dynamically with traffic.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Thrput</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECMP-WCMP (Local Stateless)</td>
<td>65G</td>
</tr>
<tr>
<td>Congestion-Aware</td>
<td>70G</td>
</tr>
</tbody>
</table>
Example: CONGA

1. Leaf switches (top-of-rack) track congestion to other leaves on different paths
2. Use this information to minimize bottleneck utilization

<table>
<thead>
<tr>
<th>Dest Leaf</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>5 3 7 2</td>
</tr>
<tr>
<td>L2</td>
<td>1 1 5 4</td>
</tr>
</tbody>
</table>

Congestion-To-Leaf Table @L0
Existing Load Balancing Schemes

Congestion-aware decisions: complex
- Measure and feed back congestion in real time
- CONGA, Hedera, HULA, MPTCP, FlowBender,…

Congestion-oblivious decisions: simple
- Random, round robin, hashing decision process
- ECMP, WCMP, Packet-Spray, Presto,…
Is there a simple load balancing scheme (with congestion-oblivious decisions) that can handle asymmetry?
LetFlow

Simple:
Randomly assign Flowlets to available paths

Flowlets:

“Flowlets are bursts from same flow separated by at least $\Delta$”

“the main origin of flowlets is the burstiness of TCP at RTT and sub-RTT scales.”

Simple Asymmetric Scenario

Detect and randomly assign Flowlets to available paths

Link Failure

Extremely simple!
- No measurements
- No feedback
- No congestion state
LetFlow

Traffic Load % (w/o link failure)

FCT (ms)

better
What’s Going On?

Link Failure

Force % of choices per path

Erico Vanini – CISCO
Flowlets are Robust

Performance is not sensitive to load balancing decisions

Traffic load: 90%
Flowlet Length

- avg flowlet length (packets)
- full capacity path
- half capacity path
- % of Choices Assigned to Green path

Ideal split
Flowlets are Elastic

- Flowlets change size based on congestion on the path
  - Uncongested path → larger flowlets
  - Congested path → smaller flowlets

→ Flowlet sizes *implicitly* encode path congestion information

... this determines the amount of traffic on each path – not just load balancing decisions

LetFlow *is* congestion-aware, despite simple random decisions
Why Are Flowlets Elastic?

- Because of congestion control (e.g., TCP)
  - A flowlet gap occurs on
    - Window cuts (Loss/ECN)
    - Latency spikes (ACK clocking)
- But, there’s a more basic reason, applicable to any congestion control protocol ...
LetFlow Analysis

- Assume flows transmit as Poisson processes

\[
\lambda_1 = \frac{C_1}{n_1} \\
\lambda_2 = \frac{C_2}{n_2}
\]
LetFlow Analysis

State transition probability $P_{n_1,n_2}^j \approx \frac{c_j}{2(c_1+c_2)} e^{-\lambda j \Delta}, j \in \{1,2\}$
LetFlow Analysis

Takeaways

1. Flows move from low rate paths (small $\lambda$) to high rate paths (large $\lambda$)
2. The flowlet timeout ($\Delta$) is important
   - Shouldn’t be too small or large

State transition probability $P_{n1,n2}^j \approx \frac{c_j}{2(c_1+c_2)} e^{-\lambda_j \Delta}, j \in \{1,2\}$
Experiments Summary

Different workloads: web search, data mining, enterprise

- Testbed experiments: ECMP, CONGA, LetFlow
  - 2 leaves 2 spines, 64 servers: symmetric & asymmetric topologies

- Simulations: ECMP, WCMP, Presto*, CONGA, LetFlow
  - Large topology: 6 leaves 6 spines, 288 servers
  - Complex asymmetric topologies: speed mismatch, combined workloads, multitier
  - Different protocols: TCP, DCTCP, DCQCN
LetFlow within 2X of CONGA; Both are much better than other schemes
Multi Destination Scenario

Traffic Load uniform to the 2 destinations
Multi Destination Scenario

Traffic Load uniform to the 2 destinations

FCT is similar between Conga and LetFlow

CONGA          LetFlow 90%          LetFlow 60%

% of destination traffic

Erico Vanini – CISCO
Other Transport Protocols

DCTCP

FCT (ms)

Load % (w/o link failure)

DCQCN

FCT (ms)

Load % (w/o link failure)
Conclusion

• Flowlet switching is a powerful technique for asymmetric load balancing

• LetFlow: a simple LB mechanism that handles asymmetry
  • Random decisions but implicitly congestion-aware
  • Suitable for standalone switches – does not need feedback

• Letflow is stochastic and reactive in nature
  • Cannot proactively prevent congestion / queue buildup like more sophisticated schemes
LET it FLOW!