Relaxing constraints in stateful network data plane design

Carmelo Cascone*, Roberto Bifulco*, Salvatore Pontarelli+, Antonio Capone*  
*Politecnico di Milano (Italy), +NEC Laboratories Europe (Germany), +Univ. Roma Tor Vergata (Italy)

1. INTRODUCTION
Program and run at line rate algorithms that read and modify data plane’s state
   - E.g.: Stateful firewall, dynamic NAT, flowlet load balancing, AQM, measurement, etc.

2. PROBLEM STATEMENT
   - Pipelining is the way to scale for high throughput (Tb/s)
   - When pipelining, accessing state at different stages of the pipeline can cause data hazards

   Example: A packet counter. For each packet increase the value of X.

3. OBSERVATIONS
   1. Pipeline’s header processing rate depends on packet size
      - Packets are read from input ports in chunks, e.g. 80 bytes in RMT
      - 80 bytes * 1 Ghz (chip clock freq.) = 640 Gb/s line rate
      - Larger packets cause inter-packet idle cycles
        - Minimize risk of data hazards

   2. Distinguish between per-flow and global state
      - Global: shared among all packets
      - Per-flow: shared by packets of the same flow
        - Different flows can be processed in parallel
      - E.g., a stateful firewall needs only per-flow state, a DNAT both

4. MOTIVATING EXPERIMENTS
Evaluate the risk of data hazards using simulations with real traffic traces

<table>
<thead>
<tr>
<th>Tracer</th>
<th>Tracer</th>
<th>Description</th>
<th>Date</th>
<th>Num. pkts</th>
<th>Num flows per 1M pkts</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRDA</td>
<td>CRDA</td>
<td>SIDR's backbone link in Chicago, load condition</td>
<td>Feb 19, 2013</td>
<td>3.5M</td>
<td>100.6k</td>
</tr>
<tr>
<td>jg-0.5</td>
<td>jg-0.5</td>
<td>SIDR's backbone link in San Jose, unusually high number of 5 tuples</td>
<td>Nov 15, 2012</td>
<td>3.6k</td>
<td>240k</td>
</tr>
<tr>
<td>mawi-15</td>
<td>mawi-15</td>
<td>SIDR's backbone link in Japan, high volume of anomalous traffic</td>
<td>Jul 21, 2015</td>
<td>1.2M</td>
<td>40.9k</td>
</tr>
<tr>
<td>fb-web</td>
<td>Facebook</td>
<td>Packet sample from 10 most active 5 tuples in a web cluster</td>
<td>2015</td>
<td>447M</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Simulation results:
Fraction of data hazards (FDH) per batch of 100k packets
- Pipeline’s read chunk = 80 bytes
- All packets back-to-back
  i.e. 100% pipeline utilization

5. APPROACH: MEMORY LOCKING
If two packets of the same flow arrive back-to-back, processing is paused for the second packet until the first one has left the stage pipeline.

How it works:
1. Each packet is associated with a flow key (FK)
2. A dispatcher enqueues packets by hashing on the FK
3. A round-robin scheduler decides if a queue can be served by looking at the head-of-line’s FK and comparing it to what is currently in the stage’s pipeline.
4. Comparison is performed by reducing the space of the FK to few bits (W)

Simulation results:
- Trace sj-12 (worst case)
- Queue size = 10 headers, W = 4 bits

Clock cycle budget (and latency) for all traces:
Maximum number of clock cycles (limited to 30) per processing function, to sustain a given throughput.
W = 4 bits. Latency values are given for 1 Gzh clock frequency, i.e. 1 clock cycle = 1 ns.