Asynchronous Deterministic Networking (ADN) Framework for Large scale networks

draft-joung-detnet-asynch-detnet-framework-01
Jinoo Joung, Jeong-dong Ryoo, Tae-sik Cheung, Yizhou Li, Peng Liu

IETF Interim Dec. 2022
Overall framework

• Decouple the latency guarantee problem from the jitter guarantee problem

• Latency guarantee
  • Regulators or Metadata based forwarding

• Jitter guarantee
  • Latency guaranteed network & Time-stamping & Buffering
Latency guarantee framework with regulators

• ATS
  • At every node
  • FIFO system + IR
  • The FIFO system usually is a combination of an output queue & scheduler, and switching fabric up until an input port module
  • IR per input port
    • has only one queue, examines HoQ packet whether it is eligible to leave.
    • Transmits as soon as it becomes so (minimal IR).
    • Does not increase the worst latency of the FIFO system.
    • but still requires individual flow states

Implementation practice of ATS

IR architecture

Flow \(\pi\)-regular

Flows from a same input port remain FIFO in a node.

IR does not increase the worst latency of the FIFO system.

FIFO system \(S\)

Minimal IR

IR per input port

High priority FIFO queue

Output port module

Strict priority Scheduler

Low priority queue

Output port
Latency guarantee framework with regulators

- **FAIR** (Flow aggregate & IR) [FAIR]
  - At “aggregation domain (AD)” boundaries
  - FA is of flows with same path in AD
  - IR per FA
  - Generalized ATS
  - Shown to work better than ATS

**Implementation practice of FAIR at an AD ingress**

<table>
<thead>
<tr>
<th>IR locations</th>
<th>Scheduler</th>
<th>Flow-based</th>
<th>Based on FA with [input, output port] of a network</th>
<th>FIFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero IR</td>
<td>IntServ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR between networks</td>
<td></td>
<td>FAIR</td>
<td></td>
<td>ATS</td>
</tr>
<tr>
<td>IR at every node</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Generalized IR architecture**

- Regulated flows
- Minimal IR per FA
- Input port module
- Output port
While fixing $p$, $F$, $E$; from $F = n^* p^h$, $E = h^*d$, we get $d = E/h$, $n = F/p^h$.

- Latency bounds with varying $h$ with $p = 8$, $E = 4$, $F = 4096$, $r = 1$ Gbps, $\sigma = L = 10$ Kbit. $h$ can be 1, 2, or 4 in real systems.
- When $h = 2$, the latency bound of FAIR: 0.083 sec.
- The latency bounds of the IntServ & ATS: 0.164 & 0.328 sec.
Latency guarantee framework with regulators

- **PFAR** (Port-based FA regulation) [ADN]
  - At every node or at critical links to break the cycle
  - FA is of flows having same input/output port of a node
  - Regulate FA, not individual flow, with \( \Sigma B, \Sigma r \)
  - Best scalability: no need to maintain individual flow states
  - Shown to work almost as well as ATS.

Implementations practice of PFAR
Theoretical latency bounds of three schemes

They are functions of the utilization because of the cycles in the topology.

Symmetrical topology & identical flows; PFAR or IR at every node.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Number of microflows in the flow under</td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td>observation</td>
<td></td>
</tr>
<tr>
<td>$a_f$</td>
<td>Arrival rate of the flow under observation</td>
<td>n/480Kbps</td>
</tr>
<tr>
<td>$a_{CT}$</td>
<td>Arrival rate of the cross traffic</td>
<td>(~0)</td>
</tr>
<tr>
<td>$B_f$</td>
<td>Max burst of the flow under observation</td>
<td>n 2400bit</td>
</tr>
<tr>
<td>$B_{CT}$</td>
<td>Max burst of the cross traffic</td>
<td>= $B_f$</td>
</tr>
<tr>
<td>$L$</td>
<td>Maximum packet length</td>
<td>2400 bit</td>
</tr>
<tr>
<td>$C$</td>
<td>Link capacity</td>
<td>100 Mbps</td>
</tr>
</tbody>
</table>

FIGURE 6. Latency bounds of three regulation strategies in the case study with the network with nine nodes, as the link utilization varies from 0 to 1.
Latency guarantee framework with metadata

Global FT based forwarding framework (C-SCORE)

1) Obtain \( F_0(p) \) at the entrance node 0, as in the Virtual Clock [VC]:
\[
F_0(p) = \max\{F_0(p-1), A_0(p)\} + L(p)/r.
\]

2) In a core node, increment FT at the previous node by \( d_h(p) \):
\[
F_h(p) = F_{h-1}(p) + d_{h-1}(p).
\]

3) \( d_h(p) \) is a non-decreasing function of \( p \) within a node busy period & should be larger than or equal to the actual delay;
\[
d_h(p) \geq A_{h+1}(p) - A_h(p).
\]

4) In a core node, preserve the service order of packets from the same input port. (Since they are already sorted, almost.)

- The metadata to carry in a packet: \( F_h(p), d_h(p) \).
  - These are dynamic and need to be updated.
  - \( d_h(p) \) can be set to \( d_h \). Then metadata update is simpler.

- \( d_h \) can be obtained (theoretical or measured) in a distributed manner; or by a central network manager then distributed.
  - As an example \( d_h \) can be \( u_h \), the maximum latency in node \( h \) for any flow.

### Symbol Definition

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>An output port module of a switching device</td>
</tr>
<tr>
<td>( F_h(p) )</td>
<td>‘Finish time’ of packet ( p ) at node ( h )</td>
</tr>
<tr>
<td>( A_h(p) )</td>
<td>Arrival time of packet ( p ) at node ( h )</td>
</tr>
<tr>
<td>( L(p) )</td>
<td>Length of ( p )</td>
</tr>
<tr>
<td>( r )</td>
<td>Flow service rate</td>
</tr>
<tr>
<td>( d_h(p) )</td>
<td>FT increment factor of ( p ) at node ( h )</td>
</tr>
</tbody>
</table>
Simulation for max latency distribution

- Link capacity 1 Gbps
- All sources generate one flow for each destination (Total number of flows 36)
- Flow under observation: DstID 1 or 4, average input rate 110 Mbps, packet length 2K bit, max burst size 20K bit
- Flow type 2: DstID 0 or 5, average input rate 11 Mbps, packet length 10K bit, max burst size 200K bit
- Flow type 3: DstID 2 or 3, average input rate 110 Mbps, packet length 10K bit, max burst size 200K bit
- Bottleneck link utilization 70%
- The quantum values of DRR are proportional to the input rates.
Jitter guarantee framework \([BN, ADN]\)

- Basic idea: Jitter guarantee \(\approx\) Reproducing the inter-arrival process with the inter-departure process of a network.
- With a latency guaranteed network, timestamping, and buffering at the network boundary:
  - E2E jitter is upper bounded by \((U+g-m)^+\).
  - ‘E2E buffered latency’ \((c_i - a_i)\) is also upper bounded by \((U-W+m)\).
  - Moreover, we can control the jitter bound. We can even have zero jitter by setting \(m=U+g\), with E2E buffered latency bound \((2U+g-W) \approx 2U\).
  - Time stamping can be placed in the RTP timestamp field or the IP/MPLS header as metadata.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_n)</td>
<td>Arrival instance of the (n)th packet of the flow to the network</td>
</tr>
<tr>
<td>(b_n)</td>
<td>Departure instance of the (n)th packet of the flow from the network</td>
</tr>
<tr>
<td>(c_n)</td>
<td>Buffer-out instance of the (n)th packet of the flow</td>
</tr>
<tr>
<td>(b_n - a_n)</td>
<td>E2E latency of the (n)th packet</td>
</tr>
<tr>
<td>(c_n - a_n)</td>
<td>E2E buffered latency of the (n)th packet</td>
</tr>
<tr>
<td>(U)</td>
<td>E2E latency upper bound of the flow guaranteed by the network</td>
</tr>
<tr>
<td>(W)</td>
<td>E2E latency lower bound of the flow guaranteed by the network</td>
</tr>
<tr>
<td>(m)</td>
<td>Jitter control parameter of the framework, (W \leq m).</td>
</tr>
<tr>
<td>(g_n)</td>
<td>Processing delay within the buffer of the (n)th packet of the flow. It includes store/lookup/forward delay.</td>
</tr>
<tr>
<td>(g)</td>
<td>Maximum (g_n) over (n).</td>
</tr>
</tbody>
</table>

The jitter between packets \(i\) and \(j\) is defined as \(|(c_i - a_i) - (c_j - a_j)|\).
Experiment with embedded systems

- The clocks (100MHz) of the two systems drift from each other by 6 μs per second.
- The source sends a burst of 20 RTP packets, each having a fixed 250-byte length, every 5 msec.
- The network module produces random delays between 50 ~ 500 μs.
- U is conservatively set at 600us considering various processing times.
- The Ethernet is set to be 1Gbps.
- Measured jitter of less than 10 μs, with a clock drift compensation algorithm [BN].
Thank you

• Please take a look at


• Comments and Questions are welcome!


