Deadline based Forwarding

draft-peng-detnet-deadline-based-forwarding-05

Shaofu Peng       ZTE
Peng Liu           China Mobile
Dong Yang          BJTU

IETF-116  March 2023, Yokohama
Updates

• Buffer size design.

• Give an illustration of schedulability conditions for leaky bucket arrival constraint function.
  Initially describe the delay resource reservation.

• Further describe the conditions for on-time mode.

• Admission control on the ingress.

• Overprovision analysis.
Motivations

• To find a potential queuing mechanism to match the requirements for scaling deterministic networks.

• Issues of existing mechanisms:
  – TSN CBS and ATS come with a high latency variance, as the minimum latency is not affected by them.
  – TSN CQF is quite challenging because it requires time synchronization.
  – TSN Multi-CQF only requires frequency synchronization, but with complex admission control and low bandwidth resource utilization.
  – The widely used priority based queuing scheme may give better average latency, but with worst case latency.

• This document propose a variants of EDF (Earliest Deadline Forwarding) scheduling, to dynamically rotate the priority of each aggregated FIFO queue and uniformly provide bounded delay/jitter.
Each deadline queue has **CT** (Count-down Time) that is decreased by **TI** (rotation timer interael), and **AT** (Authorization Time) that is for sending duration.

A packet with Allowable Queuing Delay (**Q**), computed by Planned Residence Time (**D**) and Accumulated Residence Variation (**E**), will put to a deadline queue, meeting **CT ≤ Q < CT+AT**.
Update-1: Buffer size design

- Each deadline queue is not bound to a fixed delay level \( (d_i) \), and it will actually store all levels of traffic during its CT decrement process.

  e.g:
  - At \( T_0 \), \( d_{100} \) traffic arrived and inserted to queue-A with \( CT = 100 \)
  - At \( T_0+10\text{us} \), \( d_{90} \) traffic arrived and inserted to the same queue-A with \( CT = 90 \)
  ... ...
  - At \( T_0+90\text{us} \), \( d_{10} \) traffic arrived and inserted to the same queue-A with \( CT = 10 \)
  
  That is, each level \( (d_i) \) of traffic arrived is \( r_i \times AT \), where \( r_i \) is the averaged bandwidth of level \( d_i \), \( AT \) is 10 us.

- Considering the stability condition \( \sum r_i \leq C \), then buffer size is designed to \( C \times AT - M \), where \( r_i \) is the bandwidth resource of level \( d_i \), \( C \) is the service rate of the deadline scheduler, \( M \) is the maximum interference packet size.

  The burst resource of any level must be less than the full-value, i.e., \( C \times AT - M \).
  However, the burst resource of each level will be more small if the bandwidth of other higher priority level cannot be negligible.
  When the concurrent burst of all levels are received, during the period of maximum level \( (d_n) \), all bursts can be sent one by one before their deadline.
Update-2: Conditions for Leaky Bucket Constraint

- For the case that \( n \) types of planned residence delay levels \((d_1, d_2, \ldots, d_n)\) is supported, and each level \( d_i \) has the leaky bucket arrival curve \( A_i(t) = b_i + r_i * t \), we have the following conditions:

\[
\begin{align*}
\sum_{i=1}^{n} b_i + r_1 * (d_n - d_1) + r_2 * (d_n - d_2) + \ldots + r_{n-1} * (d_n - d_{n-1}) + (r_2 + \ldots + r_n) * AT & \leq C * d_n - M \\
\end{align*}
\]

- For each level \( d_i \) of the link, the parameters \((b_i, r_i)\) is called its delay resource pool that can be reserved by the service.
Update-3: Conditions for On-time Mode

• The on-time mode does not cause the arrival curve to exceed the expected traffic constraint function, however, it is non-work-conserving and waste the opportunity to send packets, cause that some packets may exceed their deadline in the extreme case, e.g., each concurrent $b_i$ is full value (i.e., $C*AT - M$).
  • suggest in-time mode for low delay service.
  • suggest in-time or loose on-time mode applied on the transit nodes, and strict on-time on the egress, for low delay jitter service.

<table>
<thead>
<tr>
<th></th>
<th>in-time</th>
<th>loose on-time</th>
<th>strict on-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>when sum(all $b_i$) equals to $C*AT - M$ (traffic well-distributed)</td>
<td>before deadline</td>
<td>before deadline</td>
<td>near deadline, $\pm AT$</td>
</tr>
<tr>
<td>when each $b_i$ equals to $C*AT - M$</td>
<td>before deadline</td>
<td>before deadline</td>
<td>partially exceed deadline</td>
</tr>
<tr>
<td></td>
<td>before deadline</td>
<td>partially exceed deadline</td>
<td>exceed deadline, $AT \sim d_n$</td>
</tr>
</tbody>
</table>
Update-4: Admission Control on the Ingress

- Traffic regulation on the incoming port of the ingress node.
  - Leaky bucket depth is set to cover the reserved burst resource.
  - Leaky bucket rate is set to cover the reserved bandwidth resource.
- All path $j$ through the link allocate delay level $d_i$ resources, $\sum (b_{ij}, r_{ij})$, is less than the delay resource pool $(b_i, r_i)$ of that link's delay level $d_i$.
  - all service $k$ of delay level $d_i$ over path $j$ contribute the reserved resources $(b_{ij}, r_{ij})$ of that path..

The observed link:
- resource pool $(b_i, r_i)$ for each $d_i$ with initial, utilized, free amounts.
Update-5: Overprovision Analysis

- According to the schedulability condition, each delay level $d_i$ has its own resources pool $(b_i, r_i)$.
  - In the extreme case, each $b_i$ can be the full value, i.e., $C \times AT - M$, the scheduling procedure maybe like:
  - However, each level does not require the overprovision bandwidth $b_i/AT$. The bandwidth resource of each level is separate from the burst resource.
    - The requirece bandwidth of service is still according to the burst size per burst interval.
# Matching Evaluation of Requirements

- **Checklist**

<table>
<thead>
<tr>
<th>Requirement items</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1. Tolerate Time Asynchrony</td>
<td>Need no time synchronization.</td>
</tr>
<tr>
<td>3.2. Support Large Single-hop Propagation Latency</td>
<td>Not affected by link propagation delay.</td>
</tr>
<tr>
<td>3.3. Accommodate the Higher Link Speed</td>
<td>Each link sets AT independently according to its speed.</td>
</tr>
<tr>
<td>3.4. Be Scalable to The Large Number of Flows</td>
<td>No states per flow on the transit nodes. No overprovision issues.</td>
</tr>
<tr>
<td>3.5. Prevent Flow Fluctuation from Disrupting Service</td>
<td>Distinguish fluctuation flow by latency compensation.</td>
</tr>
<tr>
<td>3.6. Tolerate Failures of Links or Nodes and Topology Changes</td>
<td>No relationship with queueing mechanism...</td>
</tr>
<tr>
<td>3.7. Support Enhancement of Queuing Mechanisms</td>
<td>In-time mode for low latency, on-time mode for low jitter.</td>
</tr>
<tr>
<td>4.1. Support Aggregated Flow Identification</td>
<td>Defined delay level ($d_1$, $d_2$, ..., $d_n$)</td>
</tr>
<tr>
<td>4.2. Support Information used by Functions ensuring Deterministic Latency</td>
<td>Defined delay resource for each level, protocol extensions to advertise and reserve resource (TBD).</td>
</tr>
<tr>
<td>4.3. Support Redundancy Related Fields</td>
<td>No relationship with queueing mechanism...</td>
</tr>
<tr>
<td>4.4. Support Explicit Path Selection</td>
<td>No relationship with queueing mechanism...</td>
</tr>
</tbody>
</table>
Next step

• Any questions and comments?

Thank you!