Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services

Zhi-Li Zhang, Zhenhai Duan, Lixin Gao, and YiWei Thomas Hou

Virtual Time Reference System: A Unifying Scheduling Framework for Scalable Support of Guaranteed Services

Zhi-Li Zhang, Zhenhai Duan, and YiWei Thomas Hou

Speaker: Wei-Ming Yin
Instructor: Ying-Dar Lin

Nov. 24th, 2000
Agenda

• Motivation

• Virtual Time Reference System
  • Core stateless framework
  • End-to-end delay bound

• Bandwidth Broker Architecture

• Admission Control for Per-Flow Guaranteed Services
  • All rate-based vs. Mixed rate- and delay-based schedulers

• Admission Control for Class-Based Guaranteed Services
  • Dynamic flow aggregation under all rate-based schedulers

• Simulation Investigation

• Conclusion and Future works
Motivation

• **Hop-by-hop admission control approach**
  • Maintain per-flow or class-based QoS states at core routers
  • Perform local admission control and resource reservation
  • Maintain consistency of soft QoS states among all core routers
  • High communication overhead, less scalability, complicated design of core routers

• **Path-oriented admission control approach**
  • Relive core routers of QoS functions
  • Scale to both per-flow and class-based guaranteed services
  • Enable sophisticated QoS provisioning and admission control
  • No or minimal configuration of core routers
Virtual Time Reference System

- A core stateless framework
- A unifying scheduling framework
  - Core routers only perform forwarding and scheduling
- Three logic components
  - Packet state (on packet)
  - Edge traffic conditioning (edge)
  - Virtual time reference/update mechanism (core)
- Characterize per-hop behavior and end-to-end delay bound
Virtual Time Reference System
System Overview

Management information bases (MIBs)
- topology information base
- policy information base
- flow information base
- path QoS state information base
- node QoS state information base

Service modules
- admission control module
- QoS routing module
- policy control module

Control plane

Step 2: admission control process

Step 3: decision (accept/reject)

Step 1: new flow, service request

new flow arrival

Edge conditioner

Data plane

A network domain

Core router

$S_1$, $S_2$, $S_i$, $S_h$
Dynamic packet state

- State types:
  - The $k$th packet of flow $j$ at core router $i$.
  - The rate-delay parameter pair $(r^j, d^j)$. ← admission control
  - The virtual time stamp $w_{i,j,k}$. ← edge
  - The virtual time adjustment term $\delta_{j,k}$. ← edge

- Carried in packet header, initialized and inserted at edge, referenced (scheduling module) and updated (forwarding module) at core.
Edge Traffic Conditioning

- Regulate packets injection rate not exceeding reserved rate

\[
a_1^{j,k+1} - a_1^{j,k} \geq \frac{L^{j,k+1}}{r_j},
\]

where \( a_1^{j,k} \) denotes the arrival time of \( k \)th packet of flow \( j \), and \( L^{j,k} \) denotes the size of that packet.
Edge Traffic Conditioning
Virtual Time Reference/Update

- Per-hop behavior

Virtual delay: \( d_{i,j,k} \) = \( \frac{L_{j,k}}{r_{j}} + \delta_{j,k} \), rate-based scheduler

Delay-based scheduler: \( d_{j} \)

Virtual finish time: \( v_{i,j,k} \) = \( w_{i,j,k} + d_{i,j,k} \) ← referenced

Error term of core router \( i \) : \( \Psi_{i} \)

Actual finish time: \( f_{i,j,k} \leq v_{i,j,k} + \Psi_{i} \) ← per-hop behavior

Propagation delay to next hop of core router \( i \) : \( \pi_{i} \)

\( w_{i+1,j,k} = v_{i,j,k} + \Psi_{i} + \pi_{i} \) ← updated
Virtual Time Reference System

Suppose total $h$ hops, of which $q$ hop are rate-based scheduler, and $h-q$ hops are delay-based schedulers. The traffic profile of flow $j$ is $(s_j, r_j, P_j, L_{j, \text{max}})$.

$$f_{h,j,k} - a_{1,j,k} \leq d_{\text{core}}^j = q \frac{L_{j,\text{max}}^j}{r_j} + (h-q)d_j^j + \sum_{i \in P} (\Psi_i + \pi_i)$$

$$d_{\text{edge}}^j = \frac{\sigma^j - L_{j,\text{max}}^j}{P_j - \rho^j} \frac{P_j - r_j}{r_j} + \frac{L_{j,\text{max}}^j}{r_j} = T_{\text{on}}^j \frac{P_j - r_j}{r_j} + \frac{L_{j,\text{max}}^j}{r_j}$$

end-to-end delay bound

$$\Rightarrow d_{e2e}^j = d_{\text{edge}}^j + d_{\text{core}}^j$$

$$= T_{\text{on}}^j \frac{P_j - r_j}{r_j} + (q+1) \frac{L_{j,\text{max}}^j}{r_j} + (h-q)d_j^j + \sum_{i \in P} (\Psi_i + \pi_i)$$
Bandwidth Broker Architecture

Control plane

Step 1: new flow/service request
Step 2: admission control process
Step 3: decision (accept/reject)

Management information bases (MIBs)
- topology information base
- policy information base
- flow information base
- path QoS state information base
- node QoS state information base

Service modules
- admission control module
- QoS routing module
- policy control module

Data plane

A network domain
QoS State Information Bases

- Flow information base
  - Flow id
  - Traffic profile: $\sigma_j, \rho_j, P_j, L_j, max$
  - Service profile: $D_{j, req}$
  - QoS reservation: $(r, d)$

- Node QoS state information base
  - Bandwidth, buffer capacity, scheduler type, error term

- Path QoS state information base
  - Hop number, sum of error terms and propagation delays, minimal residual bandwidth along the path
Whether there is a feasible rate or not, with which delay requirement is less than or equals to e2e delay bound.
Admission Control

- For per-flow guaranteed services
  - Pure rate-based schedulers
  - Mixed rate- and delay-based schedulers
  - Scalability?
- Dynamic flow aggregation
- For class-based guaranteed services
  - Pure rate-based schedulers
  - Mixed rate- and delay-based schedulers
Per-flow: Path with Only Rate-based Schedulers

- Parameters

\[ P : \text{path} \]
\[ v : \text{flow} \]
\[ r^v : \text{reserved rate of flow } v \]
\[ d^v : \text{delay parameter of flow } v \]
\[ S_i : \text{core router } i \]
\[ F_i : \text{set of flows currently traversing } S_i \]
\[ C_i : \text{total bandwidth of } S_i \]
\[ C_{res}^{S_i} : \text{residual bandwidth at } S_i \]
\[ C_{res}^P : \text{minimal residual bandwidth, i.e. } C_{res}^P = \min_{i \in P} C_{res}^{S_i} \]
\[ (\sigma^v, \rho^v, P^v, L^v, \text{max}) : \text{traffic profile of flow } v \]
\[ D_{req}^v : \text{end-to-end delay requirement} \]
Per-flow: Path with Only Rate-based Schedulers

- **Fundamental inequalities**

\[ \rho^v \leq r^v \leq P^v \quad \text{and} \quad r^v \leq C^P_{\text{res}} \]

\[ T^v_{\text{on}} \frac{P^v - r^v}{r^v} + (h + 1) \frac{L^{v,\text{max}}}{r^v} + \sum_{i \in P} (\Psi_i + \pi_i) \leq D^{v,\text{req}} \]

- **Feasible rate range derivation**

Let \( r_{\text{min}}^v \) be the smallest \( r^v \)

\[ \Rightarrow r_{\text{min}}^v = \left[ T^v_{\text{on}} P^v + (h + 1) L^{v,\text{max}} \right] / \left[ D^{v,\text{req}} - \sum_{i \in P} (\Psi_i + \pi_i) + T^v_{\text{on}} \right] \]

Therefore, feasible rate range, \( R^*_\text{fea} \), is defined as

\[ [r^{\text{low}}_{\text{fea}}, r^{\text{up}}_{\text{fea}}] = \left[ \max \left\{ \rho^v, r_{\text{min}}^v \right\}, \min \left\{ P^v, C^P_{\text{res}} \right\} \right] \]

- The flow is admissible if the feasible rate range is non-empty, \( d^v \) is not necessary to be determined.

- The admissibility test can be done in \( O(1) \).
Fundamental inequalities

\[ \rho^v \leq r^v \leq P^v, \]
\[ r^v \leq C^P_{\text{res}}, \]
\[ T^v_{\text{on}} \frac{P^v - r^v}{r^v} + (q + 1) \frac{L^v_{\text{max}}}{r^v} + (h - q) d^v + \sum_{i \in P} (\Psi_i + \pi_i) \leq D^v_{\text{req}}, \]

and

\[ \sum_{\{j \in F_i : d^j_i \leq d^k_i\}} \left[ r^j_i (d^k_i - d^j_i) + L^j_{\text{max}} \right] + \left[ r^v_i (d^k_i - d^v_i) + L^v_{\text{max}} \right] \leq C^i d^k_i. \]
Efficient algorithm:

0. \( t^\nu = \frac{1}{n-q}[D^\nu,req - D^p_{tot} + T^\nu_{on}] \)
1. Let \( m^* \) such that \( d^{m^*-1} < t^\nu \leq d^{m^*} \)
2. for \( m = m^*, m^* - 1, \ldots, 2, 1 \)
3. \( R^m_{fea} \leftarrow [r^m_{l, fea}, r^m_{r, fea}] \)
4. \( R^m_{del} \leftarrow [r^m_{l, del}, r^m_{r, del}] \)
5. if \( (R^m_{fea} \cap R^m_{del} = \emptyset) \)
6. \( \text{if } (R^m_{fea} = \emptyset) || R^m_{del} = \emptyset || r^m_{l, fea} < r^m_{l, del} \)
7. break with \( d^\nu = d^m \)
8. else /*\( R^m_{fea} \cap R^m_{del} \neq \emptyset */
9. \( \text{if } (r^m_{l, fea} < r^m_{l, del}) \)
10. \( r^\nu \leftarrow r^m_{l, del}, d^\nu \leftarrow t^\nu - \frac{r^\nu}{r^\nu} \)
11. break with \( d^\nu \)
12. if \( (d^\nu > t^\nu) \) no feasible value found
13. else return \( d^\nu \)

Time complexity:

\( O(M), \text{ where } M \leq \sum_{S_i \text{ id delay-based}} |F_i| \).
Class-based Guaranteed Service Model

- Enhance the scalability of proposed BB architecture
- Service Model

- Dynamic flow aggregation has not been identified nor addressed
Dynamic Flow Aggregation (1/5)

- Impact on e2e delay (macroflow $\alpha \rightarrow \alpha'$)
- All rate-based schedulers

$$d'_{\text{edge}} = T_{on} \frac{P' - r'\alpha'}{r'\alpha'} + \frac{L'_{\text{max}}}{r'\alpha'}$$

Some packets from new macroflow may experience a worse-case delay in the network core by $d^{\alpha}_{\text{core}} = h \frac{L^{P,\text{max}}}{r^{\alpha'}} + \sum_{i \in P} (\Psi_i + \pi_i)$

Instead of $d^{\alpha'}_{\text{core}} = h \frac{L^{P,\text{max}}}{r^{\alpha'}} + \sum_{i \in P} (\Psi_i + \pi_i)$
Dynamic Flow Aggregation (2/5)

- **Edge delay bound**
  - Contingency bandwidth: to eliminate the lingering delay effect of the backlog packets
  - A new microflow $\nu$ aggregates or de-aggregates, the contingency bandwidth is $\Delta r^\nu$, and the contingency period is $\tau^\nu$.

\[ d_{\text{edge}}^{\text{new}} \leq \max \{ d_{\text{edge}}^\alpha, d_{\text{edge}}^{\alpha'} \} \]
Dynamic Flow Aggregation (3/5)

- **Edge delay bound**
  - The microflow $v$ is with $(\sigma^v, \rho^v, P^v, L^v_{\text{max}})$.
  - Sufficient conditions on $\Delta r^v$ and $\tau^v$:

\[
\begin{align*}
\Delta r^v & \geq P^v - r^v \quad \text{(microflow join)} \\
\Delta r^v & \geq r^v \quad \text{(microflow leave)}
\end{align*}
\]

and $\tau^v \geq \frac{Q(t^*)}{\Delta r^v}$,

where $Q(t^*) \leq d^\alpha_{\text{edge}}(r^\alpha + \Delta r^\alpha(t^*))$ is the backlog,

where $\Delta r^\alpha(t^*)$ is the total contingency bandwidth allocated to the macroflow $\alpha$ at time $t^*$.

$\rightarrow$ contingency period bounding
Dynamic Flow Aggregation (4/5)

- Core delay bound

\[
d_{\text{core}}^{\alpha'} = q \max \left\{ \frac{L_{P,\text{max}}}{r^\alpha}, \frac{L_{P,\text{max}}}{r^\alpha'} \right\} + (h - q)d^\alpha + \sum_{i \in P} (\Psi_i + \pi_i)
\]
Dynamic Flow Aggregation (5/5)

- Admission control: Microflow join

\[
d'_{e2e} = d'_{edge} + \max\{d'_{core}, d'_{core}\} \leq D_{\alpha,req}
\]

since \( r'_{\alpha} \geq r_{\alpha} \), hence, \( d_{\alpha} \leq d'_{core} \).

\[
\Rightarrow d'_{edge} \leq D_{\alpha,req} - d_{\alpha,core} \quad \text{..........(a)}
\]

also, \( \rho^v \leq r'_{\alpha} - r_{\alpha} \leq P^v \quad \text{..........(b)}
\]

\[
\Rightarrow r'_{\alpha} \text{ can be derived from (a) (b)}
\]
Simulation Investigation
Comparison

<table>
<thead>
<tr>
<th></th>
<th>Number of Calls admitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rate-Based Only</td>
</tr>
<tr>
<td><strong>Delay bounds</strong></td>
<td>2.44</td>
</tr>
<tr>
<td><strong>IntServ/GS</strong></td>
<td>30</td>
</tr>
<tr>
<td><strong>Per-flow BB/VTRS</strong></td>
<td>30</td>
</tr>
<tr>
<td><strong>Aggr BB/VTRS</strong></td>
<td>cd = 0.10</td>
</tr>
<tr>
<td></td>
<td>cd = 0.24</td>
</tr>
<tr>
<td></td>
<td>cd = 0.50</td>
</tr>
</tbody>
</table>
Mean Reserved Bandwidth
Flow Blocking Rate

![Graph showing Flow Blocking Rate]

- Per-flow BB/VTRS
- Contingency Period Feedback
- Contingency Period Bound

Y-axis: Blocking probabilities
X-axis: Offered loads
Conclusion

• Present a novel BB architecture based on VTRS
• Decouple the QoS control plane from data plane
• Propose path-oriented admission control approach
• Support per-flow and class-based guaranteed services
• No or minimal configuration of core routers
Future Works

- Distributed bandwidth broker architecture
- Inter-Domain QoS reservation and service level agreement