Simulation-Based Evaluation of a Delay-Based Forwarding Concept

Mehmet Cakir
Dept. Computer Science, Hamburg University of Applied Sciences, Germany
mehmet.cakir@haw-hamburg.de
Outline

1. Introduction to Future Networking Applications and QoS
2. Delay-Based Forwarding Concept
3. Implementation with OMNeT++/INET
4. Simulation-Based Evaluation
5. Conclusion & Outlook
Introduction to Future Networking Applications and QoS
Introduction to Future Networking Applications and QoS

• Future networking applications have high demands in communication
  • Industry 4.0
  • Haptic applications
  • Virtual Reality (VR)

• Present QoS-Mechanisms can prioritize network traffic

• IntServ can prioritize particular traffic

• Clemm and Eckert propose the novel approach Delay-Based Forwarding (DBF) with using packet metadata

• DBF wants to provide fairness in the network

• We proof our simulation results against the paper results
Delay-Based Forwarding Concept
Delay-Based Forwarding Concept

Clemm and Eckert, High-Precision Latency Forwarding over Packet-Programmable Networks
Delay-Based Forwarding Concept

**SLO parameters:**
- \( D_{\text{min}} = 300\,\mu s \)
- \( D_{\text{max}} = 600\,\mu s \)
- \( e\text{Delay} = 0\,s \)

**Packets from Sender1**
- At ingress:
  - \( e\text{Delay} = 100\,\mu s \)
- Computed SLOs:
  - \( t_{\text{Min}} = \text{Trcv} + 100\,\mu s \)
  - \( t_{\text{Max}} = \text{Trcv} + 400\,\mu s \)
- At egress:
  - \( e\text{Delay} = 200\,\mu s \)

**Sender1**
- Latency: 100\,\mu s

**Packets from Sender2**
- At ingress:
  - \( e\text{Delay} = 100\,\mu s \)
- Computed SLOs:
  - \( t_{\text{Min}} = \text{Trcv} + 50\,\mu s \)
  - \( t_{\text{Max}} = \text{Trcv} + 100\,\mu s \)
- At egress:
  - \( e\text{Delay} = 150\,\mu s \)

**Sender2**
- Latency: 100\,\mu s

**Latency: 100\,\mu s**

**Receiver**

**Latency: 100\,\mu s**
Implementation with OMNeT++/INET
Implementation with OMNeT++/INET

Dependencies on INET:
- StandardHost
- Router
- Ipv4NetworkLayer
- ITrafficConditioner
  - ingressTC
  - egressTC
- PriorityQueue
  - PriorityClassifier
  - PacketQueue
  - PriorityScheduler
Implementation with OMNeT++/INET

Layer 3
- DBFComputer (Computes DBF Parameters)
- DBFIpv4 (Adds DBFIpv4Options)

Layer 2
- DBFClassifier (Classifies Packets)
- DBFPriorityQueue (Provides Priority Queues)
- DBFScheduler (Schedules and Dequeues Packets)
Simulation-Based Evaluation
Simulation-Based Evaluation
Simulation-Based Evaluation

![Graph showing Jitter in microseconds for various queue outputs, comparing Emulation and Simulation.]
Simulation-Based Evaluation

![Simulation-Based Evaluation Graph](image)

- **Queue output**
  - Queue 1
  - Queue 2
  - Queue 3
  - Queue 4

- **Jitter in microseconds**
  - Queue 1: [values]
  - Queue 2: [values]
  - Queue 3: [values]
  - Queue 4: [values]

- **SLO-lb/Dmin 700µsec**

- **Labels**:
  - Emulation
  - Simulation
Simulation-Based Evaluation

SLO-lb/Dmin 800μsec

Jitter in microseconds

Queue output

Queue 1
Queue 2
Queue 3
Queue 4

Emulation
Simulation
Simulation-Based Evaluation

SLO-lb/Dmin 900µsec

Queue 1  Queue 2  Queue 3  Queue 4

Queue output

Jitter in microseconds

- Simulation
- Emulation
Simulation-Based Evaluation

SLO-lb/Dmin 1000µsec

Queue output

Queue 1  Queue 2  Queue 3  Queue 4

Jitter in microseconds

- Simulation
- Emulation
Simulation-Based Evaluation

![Graph showing Jitter in microseconds for Queue 1, Queue 2, Queue 3, and Queue 4. The graph indicates a trend where Jitter decreases as Queue output increases. The SLO-lb/Dmin 1100μsec line is represented by a red line, while the Emulation line is represented by a blue line. The y-axis represents Jitter in microseconds, with values ranging from 0 to 300, and the x-axis represents Queue output, ranging from Queue 1 to Queue 4.]
Conclusion & Outlook
Conclusion & Outlook

Conclusion
• Transparency to higher Layers with using DBF at Layer 3 and 2
• Differences in results but similar behavior

Outlook
• Figuring out the differences between emulation and simulation
• Further investigations with TCP in a heterogeneous network
• Modification of DBF which increases priority using eDelay