CROSS-LAYER LATENCY-AWARE AND -PREDICTABLE DATA COMMUNICATION

Andreas Schmidt

Apr 07th, 2020

Anywhere on Earth

Doctoral Colloquium
CYBER-PHYSICAL SYSTEMS
DISTRIBUTED CYBER-PHYSICAL SYSTEMS
A REAL-WORLD DISTRIBUTED CPS

Proprietary Solution
Enhanced Shock Burst (ESB)

Our Open Solution
Wi-Fi, IP, PRRT

Publication-in-Progress [BSGPH20]
Cyber Physical Systems: Design Challenges [Lee08]

"Passing of time is inexorable"

Most abstractions insufficiently capture timing

Edward A. Lee
UC Berkeley
LATENCY-*

**AWARENESS**
- Know what latency you **require**
- Know what latency you **cause**
- **Share** this with others

**PREDICTABILITY**
- Built upon **awareness**
- Improve **confidence**
- **Share** confidence with others
STATE OF AFFAIRS IN NETWORKING

The Internet
- **most dominant** interconnecting communication systems
- **best effort**, i.e. no predictable timing (by design)

- 📶 **Link**: Transmission latencies known, but not exposed
- ⚇ **IP**: No timing at all

- 🃏 **Transport Layer**:
  - TCP: ✗
  - UDP: ✗
  - RTP: ✗

💡 Let's go back to **proprietary, closed** networks? **No!** Why?
Internet Architecture [Cla18] provides INTEROPERABILITY
RESEARCH QUESTION

CAN WE MAINTAIN INTEROPERABILITY AND AT THE SAME TIME ACHIEVE LATENCY-AWARENESS AND -PREDICTABILITY?

* as you might assume, this is a question too big for a PhD thesis
TODAY'S RESEARCH QUESTION

CAN WE BUILD TRANSPORT LAYERS WITH LATENCY-AWARENESS AND -PREDICTABILITY?
PRRT

[SH16a, GGG+19]
WHAT IS THE PROBLEM?

Full reliability $\Rightarrow$ infinite time [Sha48, Fei54]

Most applications...

... have (some) time constraint

... can tolerate (some) missing data

⚡ (Today's) transport protocols do not consider that
FLAVOURS OF RELIABILITY

**Full**

- $send(P_1)$ at $t=0$
- $send(P_2)$ at $t=1$
- $send(P_3)$ at $t=2$
- $send(P_4)$ at $t=3$
- $retransmit(P_2)$ at $t=4$

- $P_{1,E=2}$
- $P_{2,E=3}$
- $P_{3,E=4}$
- $P_{4,E=5}$

- $deliver(P_1)$
- $buffer(P_3)$
- $buffer(P_4)$

- $P_{2,E=3}$ expires

**Partial**

- $send(P_1)$ at $t=0$
- $send(P_2)$ at $t=1$
- $send(P_3)$ at $t=2$
- $send(P_4)$ at $t=3$
- $retransmit(P_2)$ at $t=4$

- $P_{1,E=2}$
- $P_{2,E=3}$
- $P_{3,E=4}$
- $P_{4,E=5}$

- $deliver(P_1)$
- $buffer(P_3)$
- $buffer(P_4)$

- $ignore(P_2)$ as $E = 3 < 4 = t$

- $deliver(P_4)$
PRRT PROTOCOL

- Predictably Reliable Real-time Transport
- Fundamental work in [Gor12]
- Redesign for control applications since 2015

LARN
Latency- and Resilience-Aware Networking
# PRRT's Place in the ISO Model

## Default

<table>
<thead>
<tr>
<th>Control/Multimedia App</th>
<th>PRRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>Any MAC</td>
<td></td>
</tr>
</tbody>
</table>

## In General

<table>
<thead>
<tr>
<th>Control/Multimedia App</th>
<th>PRRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any Layer with Process-to-Process Datagram Delivery</td>
<td></td>
</tr>
</tbody>
</table>
### PRRT CHANNEL MEASUREMENT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round-trip Time</td>
<td>NTP Algorithm [MMBK10]</td>
</tr>
<tr>
<td>Delivery Rate</td>
<td>IETF Draft [CCYJ17]</td>
</tr>
<tr>
<td>Loss Rate &amp; Correlation</td>
<td>Statistics over success-failure sequences</td>
</tr>
</tbody>
</table>

- Selective ACKs with feedback
- Measurements exposed via API
PRRT'S RECEIVE MODES

send() 0

PE=4

w = 1

recv_asap() returns P

recv_ordered() returns P

P expired
PRRT'S FUNCTIONS

- Error Control
  - Adaptive Hybrid Error Coding [Gor12]

- Congestion Control
  - Based on BBR ideas [CCG+16]
  - Utilization & measurement
  - Latency-avoidance

- Rate and Flow Control
  - Later: X-Pace
EVALUATION

\[ m_5^{PC} - m_1^{PC} \]

\[ m_6^{PC} - m_1^{PC} \]

One-way delay [ms]

Experiment time [s]

[GGG+19]
USABILITY

Policy

- Publicly available at prrt.larn.systems
- MIT licensed

Technology

- C implementation & API
- Python API (via Cython)
- Rust API (via bindgen)
WHAT IS THE PROBLEM?

- CPS demand predictable latency and timing
- Cross-layer, intra-host profiling is required
- *Microsecond regime* is increasingly relevant [BMPR17]
STAMPS

- **Wall-Time**
  - expensive to capture ($\approx 70$ns)

- **Processor Cycles**
  - cheap to capture ($\approx 10$ns)
  - non-trivial relation to wall-time
WHAT IS X-LAP?

X = Cross, Lap = time to do one round

X-Lap is a cross-layer, intra-host, timing & latency analysis tool

download at xlap.larn.systems
STAMPS IN X-LAP

e.g.: ChannelTransmit

$t = 5 \text{ ms} \quad c = 7\,156\,294$
X-LAP IN ACTION
## TRACES DATA FORMAT

<table>
<thead>
<tr>
<th>SN</th>
<th>...</th>
<th>ChannelTransmit_T</th>
<th>ChannelTransmit_C</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>...</td>
<td>5.012ms</td>
<td>7 156 294</td>
<td>...</td>
</tr>
<tr>
<td>43</td>
<td>...</td>
<td>5.029ms</td>
<td>47 961 303</td>
<td>...</td>
</tr>
</tbody>
</table>

... ... ... .... ...
```c
void send_packet(PrrtPacket* pkt) {
  pace(); // delays sending of data using cross-layer pacing
  wait_for_free_congestion_window_space();
  compute_next_send_time(); // for cross-layer pacing
  char[] bytes = serialize_packet_to_bytes(pkt);
  XlapStamp_Cycle(LinkTransmitStart, pkt->seqno);
  struct timespec timestamp;
  uint64_t cyclestamp;
  send(sock_fd, bytes, &timestamp, &cyclestamp);
  XlapStamp_TimeValue(ChannelTransmit, timestamp, pkt->seqno);
  XlapStamp_CycleValue(ChannelTransmit, cyclestamp, pkt->seqno);
  track_outstanding_packet(pkt);
  XlapStamp_Cycle(LinkTransmitEnd, pkt->seqno);
}
```
PACKET TRACE

The diagram above illustrates various packet trace events over time, with the X-axis representing time in microseconds (us). The events include:

- Decoding
- Feedback
- HandlePacket
- ReceiverIPC
- Receiver
- SenderEnqueued
- SenderIPC
- Enqueue
- Submit
- LinkTransmit
- PrrtTransmit
- Send
- Sender
- EndToEnd

Each event is represented by a line, with different colors indicating different processes:
- Red: Sender
- Blue: Receiver
- Black: EndToEnd
TRACE JITTER
MULTISERIES CORRELATION

CDF

Sender [us]

Send [us]

PrrtTransmit [us]

LinkTransmit [us]

Submit [us]

Receiver [us]

ReceiverIPC [us]

Feedback [us]
X-PACE

[SRGP+19]
WHAT IS THE PROBLEM?

 BUFFEROBLOAT [GN12]
PACE

\[ P, [P] = \frac{time}{unit \ of \ work} \]

PACING

intentionally delaying a transmission
CROSS-LAYER PACING

Every step $i \in [0 : n - 1]$ considers and adapts its pace $P^{(i)}$ to the bottleneck pace $P^{(btl)}$.

PACED SYSTEMS

$S$ is paced iff

$$\forall i, j \in [0 : n - 1]:
\begin{align*}
&i < j \Rightarrow P^{(i)} \geq P^{(j)}
\end{align*}$$
PROPAGATE PACES

MUST slow down to $P^{(btl)}$

MAY slow down to $P^{(i+1)} \in [P^{(i+1)} : P^{(btl)}]$
REDUCE 無駄*

* muda or waste, originally described by Toyota Production System [Ohn88]

📦 **Inventory:** only a delivered message is useful

⏰ **Waiting:** a message loses value over time

🚚 **Overprocessing:** as-fast-as-possible causes losses
X-PACE IN PRRT

\[ \max(P_{\text{deliver}}, P_{\text{receive}}, P_{\text{transmit}}, P_{\text{nw}}) \]

Sender

Application
\[ P_{\text{send}} \]

Send Queue

PRRT
\[ P_{\text{transmit}} \]

Receiver

Application
\[ P_{\text{deliver}} \]

Recv Queue

PRRT
\[ P_{\text{receive}} \]

Network

explicit backward pacing

max
\[ \max\left(\frac{P_{\text{deliver}}}{P_{\text{receive}}}\right) \]

max
\[ \max\left(\frac{P_{\text{send}}}{P_{\text{transmit}}}\right) \]

\[ P_{\text{nw}} \]
CROSS-LAYER API

send_sync()  socket.btl_pace
EVALUATION APPROACH

PRRT with X-Pace

VS.

Optimized TCP Variants (CUBIC, BBR)
traceroute to 79.199.28.123 (79.199.28.123), 30 hops max, 60 byte packets
1  vlan-herfet-neu.nt.uni-saarland.de (134.96.86.1) 0.400 ms 0.358 ms 0.420 ms
2  c65eb36-win.net.uni-saarland.de (134.96.6.54) 0.328 ms 0.314 ms 0.384 ms
3  cr-dui1-te0-5-0-2-4.x-win.dfn.de (188.1.241.185) 8.687 ms 8.688 ms 8.680 ms
4  cr-fra2-be16.x-win.dfn.de (188.1.144.178) 9.618 ms 9.732 ms 9.727 ms
5  ffm-b12-link.telia.net (213.248.97.40) 9.226 ms 9.563 ms 9.191 ms
6  ffm-bb3-link.telia.net (62.115.142.46) 10.069 ms 9.761 ms 9.704 ms
7  ffm-b4-link.telia.net (62.115.120.6) 9.802 ms 9.792 ms 10.024 ms
8  dtag-ic-319284-ffmpeg-b4.c.telia.net (213.248.93.187) 10.374 ms 10.175 ms 10.17
9  91.23.246.213 (91.23.246.213) 13.230 ms 13.252 ms 13.234 ms
10  * * *
... 
30  * * *
RESULTS

CDF

2 × 10^1 3 × 10^1 4 × 10^1

E2E Delivery Time [ms]

TCP-CUBIC
TCP-BBR
PRRT
$DT_{opt}$
TRANSPARENT TRANSMISSION
SEGMENTATION

[SH16b, SH17b]
WHAT IS A PROBLEM?

⚠️ Small receiver buffer ⇒ flow-limited communication
TRANSPARENT TRANSMISSION SEGMENTATION

Relay: Transport-Layer Performance-Enhancing-Proxy [RFC3135]
TTS IN ACTION

E2E

Sender → Node → Receiver

TCP RCV Buffer

TTS

Sender → Node with Relay → Receiver

Relay TCP RCV Buffer

TCP RCV Buffer

Legend

- Used Link Capacity
- Unused Link Capacity
- Buffer Capacity
IDEAL RECEIVER BUFFER

derived from a theoretical flow control model

\[ B_{\text{relay}} = \frac{RTT_1}{RTT_2} \cdot B_{\text{recv}} \]

\[ \Lambda = \frac{R_{\text{eff},TTS}}{R_{\text{eff},E2E}} = \frac{RTT_1}{RTT_2} + 1 \]
EVALUATION

\[ B_{\text{recv}} = 4 \text{ KiB}, \quad RTT_1 = 20 \text{ ms}, \quad RTT_2 = 5 \text{ ms} \]

\[ \Rightarrow \Lambda = 5, \quad B_{\text{relay}} = 4 \cdot 4 \text{KiB} \]
BENEFITS OF TTS

- Lossy Last Mile
- High Reordering due to Jitter
OUTLOOK

e.LARN, [SGPH20]
FUTURE WORK

- Energy-Awareness
- Statistical Shaping
- Multicast Support
WRAP-UP

NETWORKED CPS

PRRT [SH16a, GGG+19]

X-LAP [RSH+17, RSH+18]

X-PACE [SRGP+19]

TTS [SH16b, SH17b]

OUTLOOK
e.LARN, [SGPH20]
QUESTIONS
APPENDIX

[SH16b] Schmidt, Andreas; Lange, Tobias; Herfet, Thorsten: “Low-Latency Multimedia Streaming using Open Networking Environments“, International Conference on Computer and Communications (ICCC), Chengdu, China, October 2016


[RSH+17] Reif, Stefan; Schmidt, Andreas; Hönig, Timo; Herfet, Thorsten; Schröder-Preikschat, Wolfgang: “X-Lap: A Systems Approach for Cross-Layer Profiling and Latency Analysis for Cyber-Physical Networks”, 15th International Workshop on Real-Time Networks (ECRTS RTN), Dubrovnic, Croatia, June 2017

[RSH+18] Reif, Stefan; Schmidt, Andreas; Hönig, Timo; Herfet, Thorsten; Schröder-Preikschat, Wolfgang: “Delta: Differential Energy-Efficiency, Latency, and Timing Analysis for Real-Time Networks“, 16th International Workshop on Real-Time Networks (ECRTS RTN), Barcelona, Spain, July 2018

[GGG+19] Gallenmüller, Sebastian; Glebke, René; Günther, Stephan; Hauser, Eric; Leclaire, Maurice; Reif, Stefan; Rüth, Jan; Schmidt, Andreas; Carle, Georg; Herfet, Thorsten; Schröder-Preikschat, Wolfgang; Wehrle, Klaus: “Enabling wireless network support for gain scheduled control“. 2nd International Workshop on Edge Systems, Analytics and Networking (EdgeSys), Dresden, Germany, March 2019

[SRGP+19] Schmidt, Andreas; Reif, Stefan; Gil Pereira, Pablo; Hönig, Timo; Herfet, Thorsten; Schröder-Preikschat, Wolfgang: “Cross-layer Pacing for Predictably Low Latency”. 6th International IEEE Workshop on Ultra-Low Latency in Wireless Networks (ULLWN), Paris, France, April 2019

[SGPH20] Schmidt, Andreas; Gil Pereira, Pablo; Herfet, Thorsten: "Predictably Reliable Real-time Transport Services for Wireless Cyber-Physical Systems", IFAC World Congress, Berlin, Germany, July 2020
REFERENCES


IMAGE SOURCES

- Edward Lee