Energy-, Latency- and Resilience-Aware Networking (e.LARN)
SPP 1914: “Cyber-Physical Networking”
http://larn.systems

Pablo Gil Pereira, Andreas Schmidt, Thorsten Herfet
Telecommunications Lab
Saarland Informatics Campus - Saarbrücken

Stefan Reif, Timo Hönig, Wolfgang Schröder-Preikschat
Department of Computer Science 4 (Distributed Systems and Operating Systems)
Friedrich-Alexander-Universität Erlangen-Nürnberg

November 15th, 2022
Recapitulation: Goals

- Tailor PRRT to control applications
- Improve PRRT’s performance
- OS for predictable communication
- Tailor network to control applications
**Outcome (since November 2018)**

<table>
<thead>
<tr>
<th>Software, Hardware &amp; Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PRRT</strong></td>
</tr>
<tr>
<td>- Cross-Layer Pacing + Rate Control</td>
</tr>
<tr>
<td>- Improved timing precision and accuracy</td>
</tr>
</tbody>
</table>
## Software, Hardware & Algorithms

<table>
<thead>
<tr>
<th><strong>PRRT</strong></th>
<th>Predictably Reliable Real-time Transport protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Cross-Layer Pacing + Rate Control</td>
</tr>
<tr>
<td></td>
<td>- Improved timing precision and accuracy</td>
</tr>
<tr>
<td><strong>X-LEEP</strong></td>
<td>Cross-Layer Pacing for Energy Efficiency</td>
</tr>
<tr>
<td></td>
<td>- Adjust CPU voltage and frequency</td>
</tr>
</tbody>
</table>
Outcome (since November 2018)

### Software, Hardware & Algorithms

<table>
<thead>
<tr>
<th><strong>PRRT</strong></th>
<th>Predictably Reliable Real-time Transport protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Cross-Layer Pacing + Rate Control</td>
</tr>
<tr>
<td></td>
<td>- Improved timing precision and accuracy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>X-LEEP</strong></th>
<th>Cross-Layer Pacing for Energy Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Adjust CPU voltage and frequency</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DeepHEC</strong></th>
<th>Deep Learning for Hybrid Error Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Adaptive error control</td>
</tr>
<tr>
<td></td>
<td>- Provide predictable adaptation time</td>
</tr>
</tbody>
</table>
Outline

Status

Cross Layer Pacing

X-LEEP

DeepHEC

Conclusion
<table>
<thead>
<tr>
<th>Bufferbloat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim Gettys; Kathleen Nichols: <strong>Bufferbloat: Dark Buffers in the Internet</strong>, CACM, 2012</td>
</tr>
</tbody>
</table>
Challenge

Bufferbloat

Jim Gettys; Kathleen Nichols: Bufferbloat: Dark Buffers in the Internet, CACM, 2012

Unpaced
Challenge

Bufferbloat

Jim Gettys; Kathleen Nichols: *Bufferbloat: Dark Buffers in the Internet*, CACM, 2012

Unpaced

Paced
Cross-Layer Pacing & X-Pace

Sender

Application $P^{(send)}$

Send Queue

PRRT $P^{(transmit)}$

Network $R_{btl}, PLR, D_{prop}, D_{queue}, D_{trans}$

Receiver

Application $P^{(deliver)}$

Recv Queue

PRRT $P^{(receive)}$

implicit pacing
Cross-Layer Pacing & X-Pace

Sender

Application \( P^{(send)} \)

Send Queue

PRRT \( P^{(transmit)} \)

Receiver

Application \( P^{(deliver)} \)

Recv Queue

PRRT \( P^{(receive)} \)

Network

\( R_{btl}, PLR, D_{prop}, D_{queue}, D_{trans} \)

\[ \max(P^{(deliver)}, P^{(receive)}, P^{(transmit)}, P^{(nw)}) \]

\[ \max(P^{(deliver)}, P^{(receive)}) \]

\[ P^{(nw)} \]

\[ P^{(deliver)} \]
Cross-Layer Pacing & X-Pace

Sender

Application $P^{(send)}$

Send Queue

PRRT $P^{(transmit)}$

Network $R_{btl}$, PLR, $D_{prop}$, $D_{queue}$, $D_{trans}$

Receiver

Application $P^{(deliver)}$

Recv Queue

PRRT $P^{(receive)}$

$\max(P^{(deliver)}, P^{(receive)}, P^{(transmit)}, P^{(nw)})$

$\max(P^{(deliver)}, P^{(receive)}, P^{(transmit)}, P^{(nw)})$

Implicit pacing

Explicit backward pacing

Explicit forward pacing

$P^{(send)}$

$P^{(deliver)}$

$P^{(receive)}$

$P^{(nw)}$
Figure: PRRT with X-Pace vs. Optimized Low-Latency TCP Variants
Outline

Status

Cross Layer Pacing

X-LEEP

DeepHEC

Conclusion
Goal: Adapt application and operating system to run-time behaviour of the network layer

Approach: Adjust voltage and frequency of the CPU

\[
f_{\text{opt}} = f_{\text{cur}} \cdot t_{\text{execution}} / t_{\text{period}}
\]
Implementation

- Goal: Adapt application and operating system to run-time behaviour of the network layer
- Approach: Adjust voltage and frequency of the CPU
- Application interface to obtain:
  - Data sampling period \((t_{\text{period}})\)
  - Last execution time \((t_{\text{execution}})\)

\[
f_{\text{opt}} = f_{\text{cur}} \cdot \frac{t_{\text{execution}}}{t_{\text{period}}}
\]
Figure: Increase in deadline violations, normalised to the execution at 1.4 GHz fixed
Evaluation Results

**Figure:** Increase in deadline violations, normalised to the execution at 1.4 GHz fixed

**Figure:** Power efficiency comparison, normalised to the execution at 1.4 GHz fixed
Outline

Status

Cross Layer Pacing

XLEEP

DeepHEC

Conclusion
Goal: meet the **application loss rate** and **latency constraints** using the **minimum** possible **redundancy information**
DeepHEC for Error Control

Predictable Reliability

- Goal: meet the **application loss rate** and **latency constraints** using the **minimum possible redundancy information**
DeepHEC for Error Control

Predictable Reliability

▶ Goal: meet the application loss rate and latency constraints using the minimum possible redundancy information

Output: \((k \times N) = \text{arg min}_{k, N} R_I(k, N)\)

Problem: unpredictable reaction time since the search space depends on channel state and application requirements

FEC Data packet

Parity packet

12 / 16
## Predictable Reliability

- **Goal:** meet the *application loss rate* and *latency constraints* using the minimum possible *redundancy information*

- **Approach:** combine proactive and reactive redundancy with Hybrid Error Coding (HEC)

### Diagram

- **Data packet**
- **Parity packet**

<table>
<thead>
<tr>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEC</td>
</tr>
</tbody>
</table>

\[
(k \times N \times P) = \arg \min_{k, N, P} R_I(k, N, P)
\]
**DeepHEC for Error Control**

**Predictable Reliability**

- **Goal:** meet the application loss rate and latency constraints using the minimum possible redundancy information
- **Approach:** combine proactive and reactive redundancy with Hybrid Error Coding (HEC)

![Diagram showing data and parity packets with k=4, p=3, and Nc=1]

\[
(k \times N \times P) = \arg \min (k, N, P) \text{RI}(k, N, P)
\]
DeepHEC for Error Control

Predictable Reliability

- **Goal:** meet the **application loss rate** and **latency constraints** using the **minimum possible redundancy information**
- **Approach:** combine proactive and reactive redundancy with Hybrid Error Coding (HEC)
- **Output:** \((k^*, N_P^*) = \arg \min_{k,N_P} RI(k, N_P)\)

![Diagram showing data packets and parity packets with k=4, p=3, N_p = [1, 2], and N_c = 1]
DeepHEC for Error Control

Predictable Reliability

- Goal: meet the **application loss rate** and **latency constraints** using the **minimum** possible **redundancy information**
- Approach: combine proactive and reactive redundancy with Hybrid Error Coding (HEC)
- Output: \((k^*, N_P^*) = \arg \min_{k,N_P} RI(k, N_P)\)
- Problem: **unpredictable reaction time** since the search space depends on channel state and application requirements

Data packet
Parity packet

\[ k = 4 \]
\[ p = 3 \]
\[ N_P = [1, 2] \]
\[ N_c = 1 \]
DeepHEC Architecture and Results

Architecture

- Input: channel state information and application requirements
- 5 fully connected hidden layers with 250 neurons each
- NN output: \((k, p, N_C)\)
- Algorithmic output: \(N_P\)
DeepHEC Architecture and Results

Architecture

- Input: channel state information and application requirements
- 5 fully connected hidden layers with 250 neurons each
- NN output: \((k, p, N_C)\)
- Algorithmic output: \(N_P\)

**Figure:** Inference Time Cumulative Distribution Function (CDF)
DeepHEC Architecture and Results

**Architecture**

- Input: channel state information and application requirements
- 5 fully connected hidden layers with 250 neurons each
- NN output: \((k, p, N_C)\)
- Algorithmic output: \(N_P\)

**Figure:** Inference Time Cumulative Distribution Function (CDF)

**Figure:** Redundancy Information error Cumulative Distribution Function (CDF)
Outline

Status

Cross Layer Pacing

X-LEEP

DeepHEC

Conclusion
# Outcome (since November 2018)

## Summary

<table>
<thead>
<tr>
<th>Year</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>2023</td>
<td>2 publications (CCNC)</td>
</tr>
<tr>
<td>2022</td>
<td>3 publications (in CCNC, TECS, EDCC)</td>
</tr>
<tr>
<td>2021</td>
<td>7 publications (in CCNC, EuroSec, LCTES, e-Energy, WEEE, PLOS)</td>
</tr>
<tr>
<td>2020</td>
<td>5 publications (in WEEE, IFAC, DCOSS Wi-DroIT, ISORC, APSys)</td>
</tr>
<tr>
<td>2019</td>
<td>2 publications (in EuroSys EdgeSys Workshop, InfoCOM ULLWN Workshop)</td>
</tr>
</tbody>
</table>
### Accepted Publications

- **Vogelgesang, Kai; Gil Pereira, Pablo; Herfet, Thorsten**: SHARQ: Scheduled HARQ for Time- and Loss-Rate-Sensitive Networks, CCNC, 2023
- **Vogelgesang, Kai; Herfet, Thorsten**: On Multihop vs. End-to-End Transport, CCNC, 2023
- **Pereira, Pablo Gil; Schmidt, Andreas; Herfet, Thorsten**: DeepHEC: Hybrid Error Coding using Deep Learning, EDCC, 2022
- **B. Herzog, S. Reif, J. Hemp, T. Höning, and W. Schröder-Preikschat**: Resource-demand estimation for edge tensor processing units, ACM TECS, 2022
- **Gil Pereira, Pablo; Herfet, Thorsten**: Polar Coding for Efficient Transport Layer in Multicast, CCNC, 2022
- **L. Gerhorst, B. Herzog, S. Reif, W. Schröder-Preikschat, and T. Höning**: AnyCall: Fast and flexible system-call aggregation, PLOS, 2021
- **B. Herzog, F. Hügel, S. Reif, T. Höning, and W. Schröder-Preikschat**: Automated selection of energy-efficient operating system configurations, WEEE, 2021
- **S. Reif, B. Herzog, J. Hemp, T. Höning, and W. Schröder-Preikschat**: AI waste prevention: Time and power estimation for edge tensor processing units, e-Energy 2021, 2021
- **S. Schuster, P. Wägemann, P. Ulbrich, and W. Schröder-Preikschat**: Annotate once — analyze anywhere: Context-aware WCET analysis by user-defined abstractions, LCTES’21, 2021
Accepted Publications

- S. Reif, B. Herzog, F. Hügel, T. Hönig, and W. Schröder-Preikschat: *Nearly symmetric multi-core processors*, APSys, 2020
- S. Reif and W. Schröder-Preikschat: *Precisely timed task execution*, ISORC, 2020
- Böhmer, Marlene; Schmidt, Andreas; Pereira, Pablo Gil and Herfet, Thorsten: *Latency-aware and-predictable Communication with Open Protocol Stacks for Remote Drone Control*, DCOSS Wi-DroIT, 2020
- Schmidt, Andreas; Pereira, Pablo Gil; Herfet, Thorsten: *Predictably Reliable Real-time Transport Services for Wireless Cyber-Physical Systems*, IFAC World Congress, 2020
- Reif, Stefan; Herzog, Benedict; Pereira, Pablo Gil; Schmidt, Andreas; Büttner, Tobias; Hönig, Timo; Schröder-Preikschat, Wolfgang; Herfet, Thorsten: *X-Leep: Leveraging Cross-Layer Pacing for Energy-Efficient Edge Systems*, WEEE, 2020
- Schmidt, Andreas; Reif, Stefan; Gil Pereira, Pablo; Hönig, Timo; Herfet, Thorsten; Schröder-Preikschat, Wolfgang *Cross-layer Pacing for Predictably Low Latency*, ULLWN, 2019
- 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*, EdgeSys, 2019
Conclusion

Network
✓ PRRT – Predictable protocol
✓ DeepHEC – Error Control

Operating System
✓ X-LEEP – Energy efficiency
✓ Precise task execution

Reliable Network Atom (RNA)
(Embedded Edge-Server, Car, Drone, ...)

Thank you for your attention. Questions?
Conclusion

Network
✓ PRRT – Predictable protocol
✓ DeepHEC – Error Control

Operating System
✓ X-LEEP – Energy efficiency
✓ Precise task execution

Integration
Optimisation
Cross-layer pacing

Reliable Network Atom (RNA)
(Embedded Edge-Server, Car, Drone, ...)

Thank you for your attention. Questions?
Conclusion

Network
- PRRT – Predictable protocol
- DeepHEC – Error Control

Operating System
- X-LEEP – Energy efficiency
- Precise task execution

Integration
- Optimisation
- Cross-layer pacing

Reliable Network Atom (RNA)
(Embedded Edge-Server, Car, Drone, ...)

Thank you for your attention. Questions?