Freie Universität Berlin

HAW HAMBURG

Reliable Firmware Updates for the Information-Centric Internet of Things

ACM ICN 2021, Virtual Event, Paris, France

Cenk Gündoğan[1] Christian Amsüss

Thomas C. Schmidt[1] Matthias Wählisch[2]

[1]HAW Hamburg [2]Freie Universität Berlin

cenk.guendogan@haw-hamburg.de christian@amsuess.com t.schmidt@haw-hamburg.de m.waehlisch@fu-berlin.de

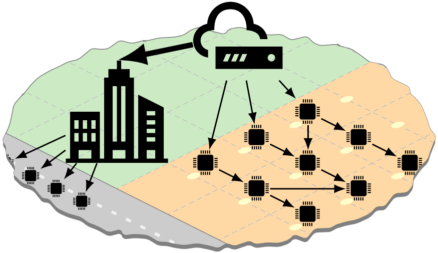
Common IoT Deployment

Device Characteristics

- Cloud services and gateways are powerful and resource-abundant
- IoT devices have low energy budgets and are resource-constrained

Network characteristics

- Connectivity between cloud services and gateways is **perpetual**
- Connectivity between gateways and IoT devices is **intermittent**



Lifecycle Management

- General purpose devices require software updates
- Increasing security demands require similar practices for IoT

A secure and reliable firmware propagation in low-power regimes is mandatory

Challenges of Firmware Propagation

- Updates are resource-consuming and show as peak loads in the Internet
- IoT firmware images are 1–2 orders of magnitude larger than sensor values
- Bandwidth limitation on constrained networks calls for new approaches

Update propagations can lead to DDoS and break security

Research Question

Can we leverage the benefits of NDN to perform secure and reliable firmware roll-outs at large-scale for the IoT?

Outline

Reliable Firmware Updates with NDN

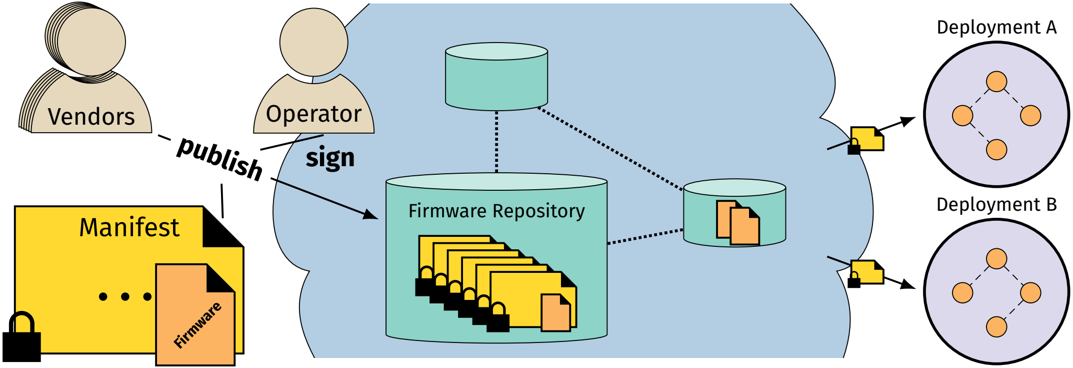
Protocol Performance Evaluation

Conclusion & Outlook

Reliable Firmware Updates with NDN

Building Blocks for Reliable Firmware Updates with NDN

- SUIT as blueprint and involvement of multiple stakeholders

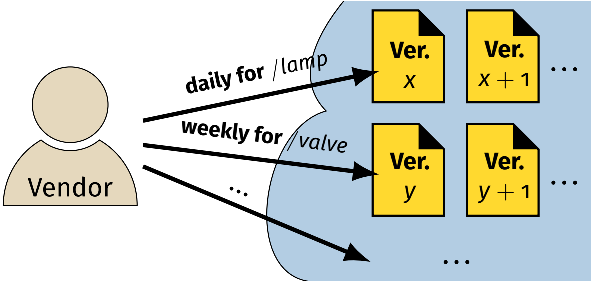


Naming Scheme and Firmware Versioning

- Hierarchy allows for FIB aggregation

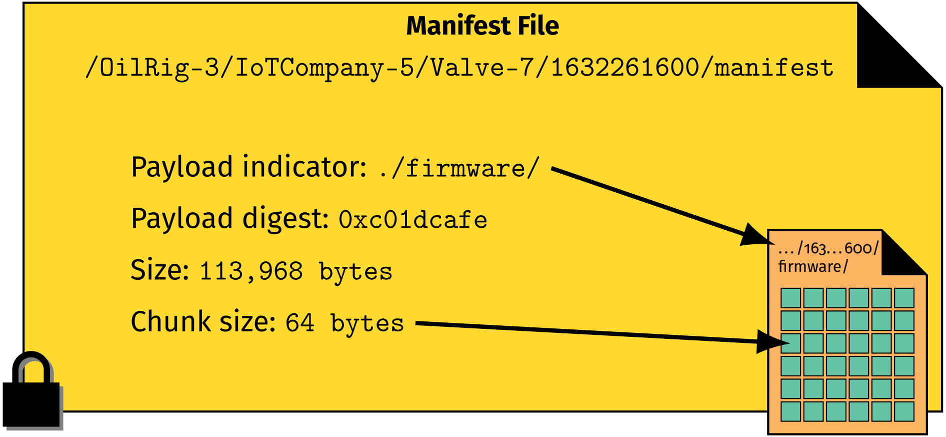
$$/\,\underbrace{\texttt{OilRig-3}}_{\text{Deployment}}\,/\,\underbrace{\texttt{IoTCompany-5}}_{\text{Vendor}}\,/\,\underbrace{\texttt{Valve-7}}_{\text{Device Class}}\,/\,\underbrace{\texttt{1632261600}}_{\text{Timestamp}}$$

- Version number is timestamp
- Release cycles per device class
- Devices request on schedule



SUIT-based Manifest

- Contains meta information on a specific firmware version
- References the actual firmware binary and chunks



Manifest File

`/OilRig-3/IoTCompany-5/Valve-7/1632261600/manifest`

Payload indicator: `./firmware/`

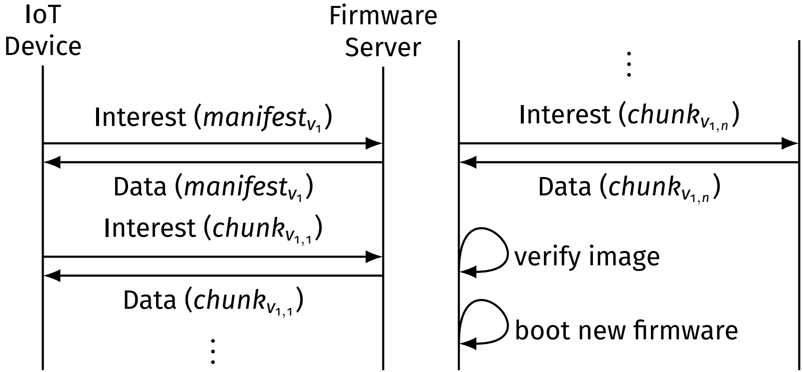
Payload digest: `0xc01dcafe`

Size: `113,968 bytes`

Chunk size: `64 bytes`

Firmware Retrieval

- Successful version discovery triggers firmware retrieval
- Complete images verify against message digest in signed manifest



Retrieval Strategies

Concurrent Retrievals

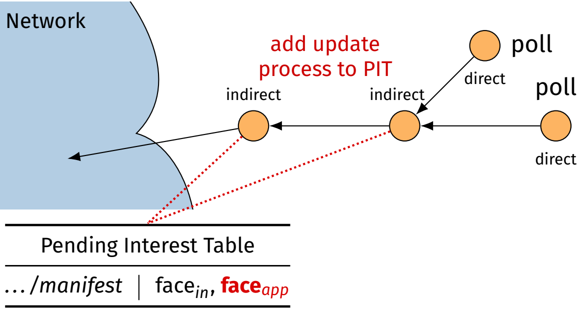
- Nodes retrieve missing chunks and also forward to downstream nodes
- Multiple nodes on a path perform update concurrently

Cascading Retrievals

- Nodes block downstream chunk requests while local retrieval is running
- Single node on a path performs update at a time

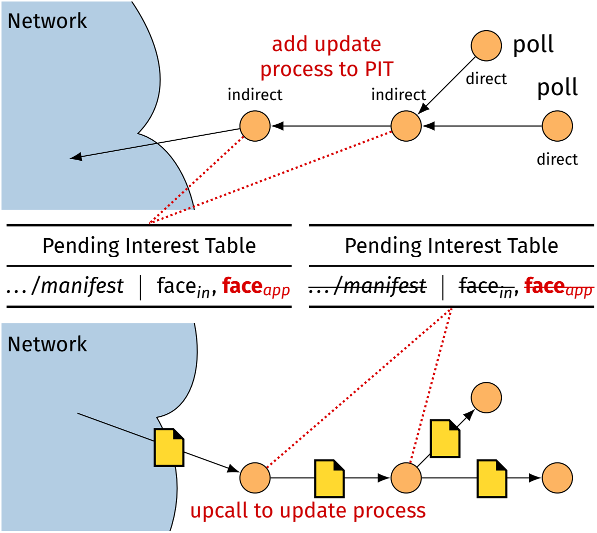
Indirect Version Discovery

- Forwarders detect valid version request and implicitly add face_{app} to PIT


Network
add update process to PIT
poll
poll
indirect
indirect
direct
direct
Pending Interest Table
$\ldots/manifest$ | face_{in}, $\textbf{face}_{app}$

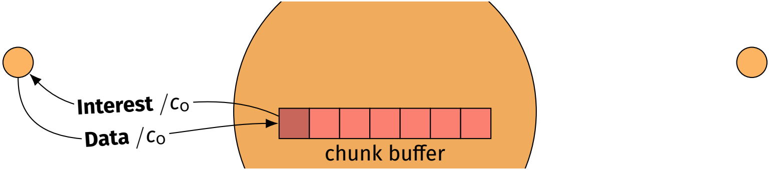
Indirect Version Discovery

- Forwarders detect valid version request and implicitly add face$_{app}$ to PIT
- Returning manifest triggers upcall to update process and also propagates downstream to face$_{in}$



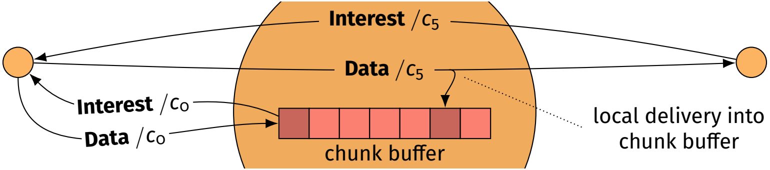
Local Buffer Management

- Chunks reside in persistent memory (e.g., embedded flash, SD card, . . .)
- NDN serves cache hits from same buffer to minimize RAM usage



Local Buffer Management

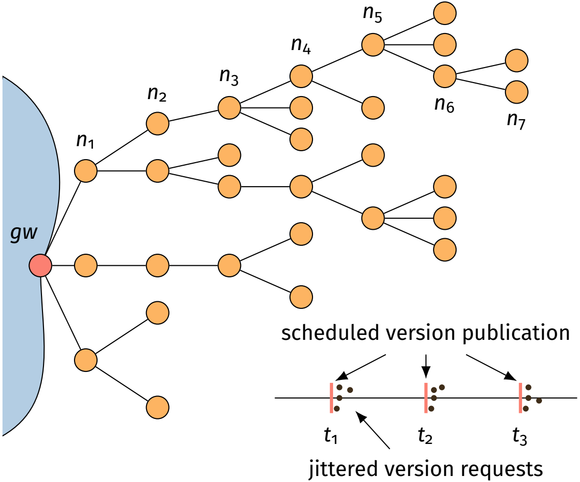
- Chunks reside in persistent memory (e.g., embedded flash, SD card, ...)
- NDN serves cache hits from same buffer to minimize RAM usage
- Forwarder locally delivers chunks from overlapping update processes



Protocol Performance Evaluation

Experiment Setup

Hardware	M3 node in IoT Lab testbed, IEEE 802.15.4
Software	RIOT
Topology	30 devices, 1 gateway
Scenario	Devices request new firmware version

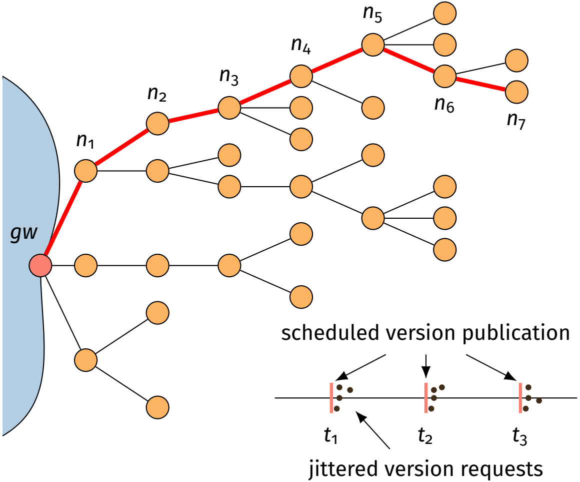


scheduled version publication

jittered version requests

Experiment Setup

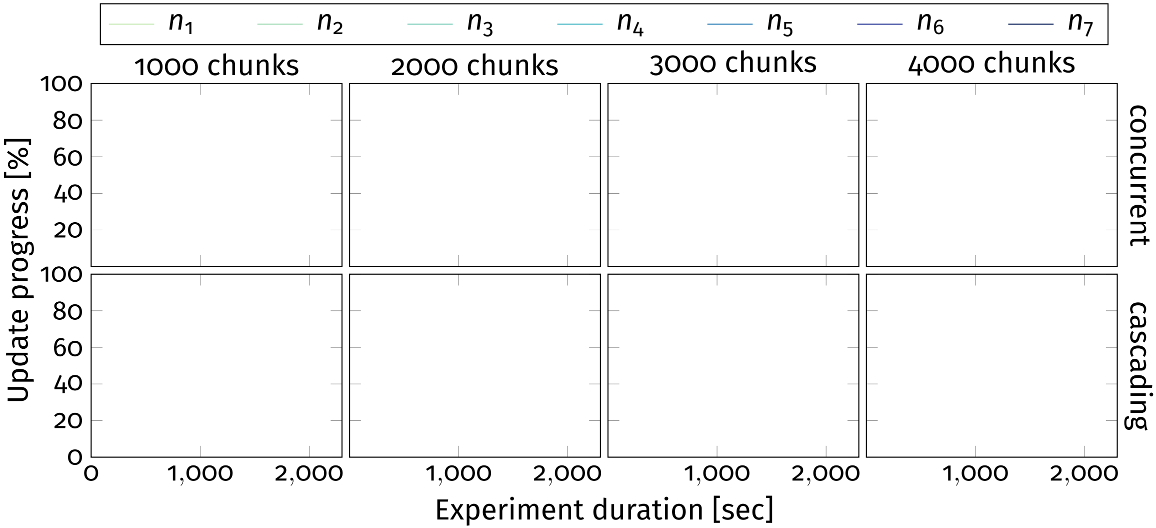
Hardware	M3 node in IoT Lab testbed, IEEE 802.15.4
Software	RIOT
Topology	30 devices, 1 gateway
Scenario	Devices request new firmware version



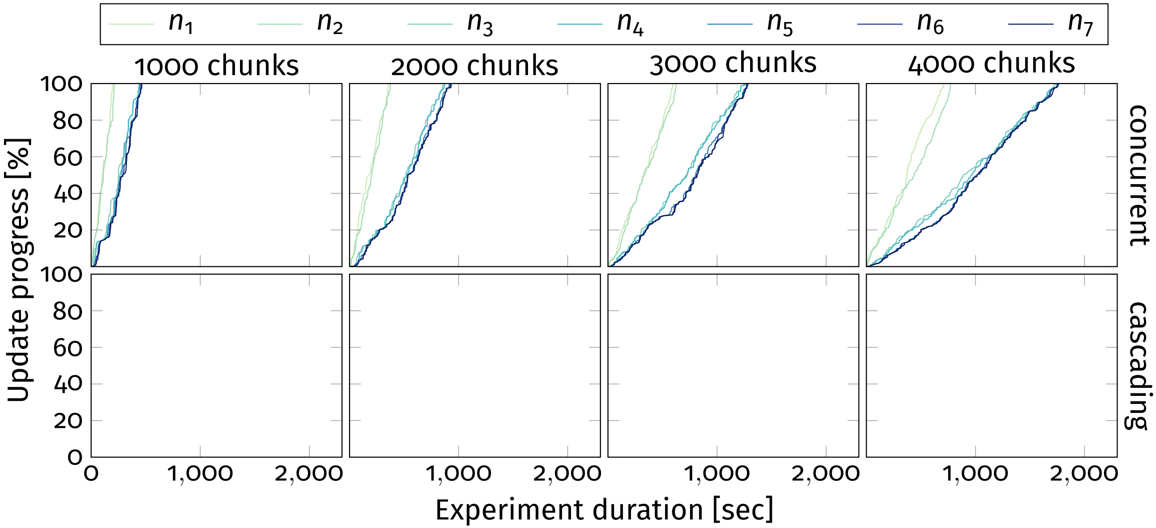
scheduled version publication

jittered version requests

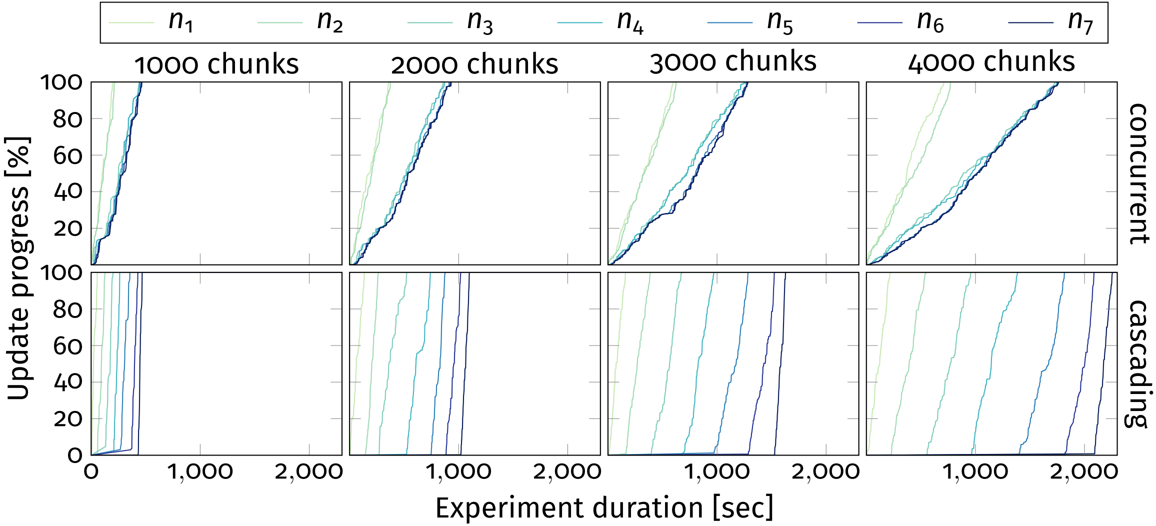
Firmware Update Progress



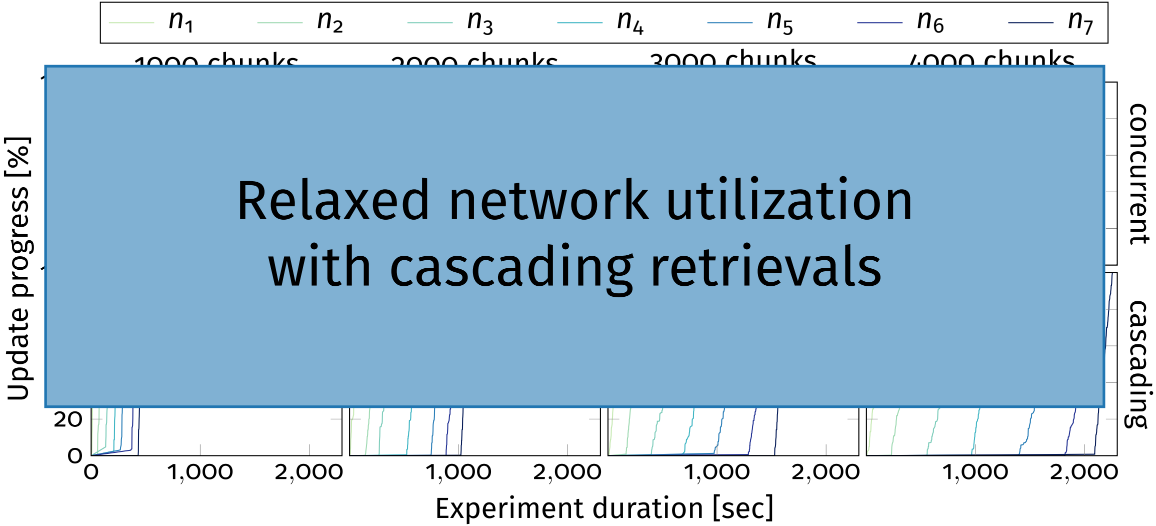
Firmware Update Progress



Firmware Update Progress

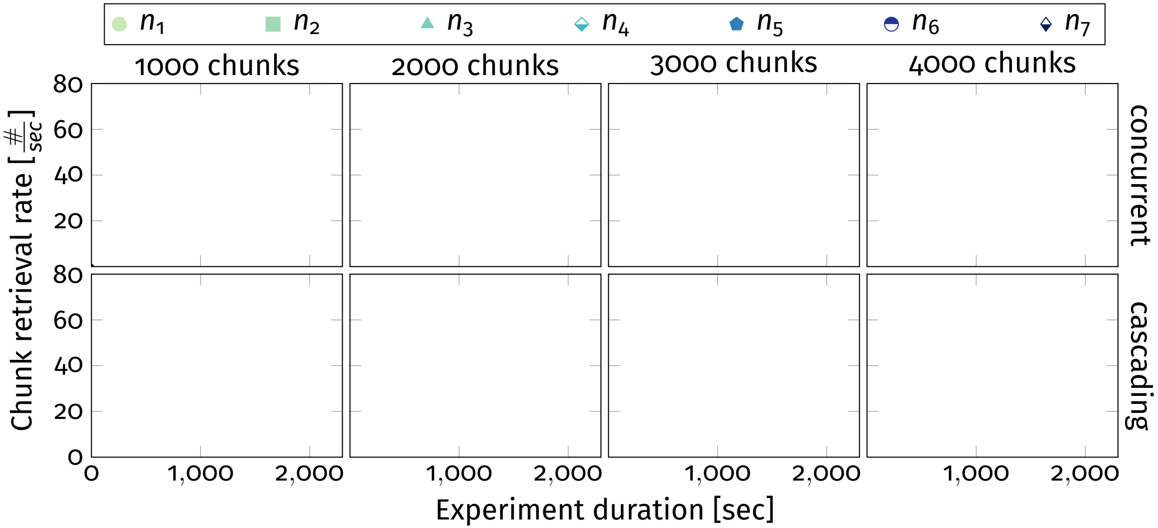


Firmware Update Progress

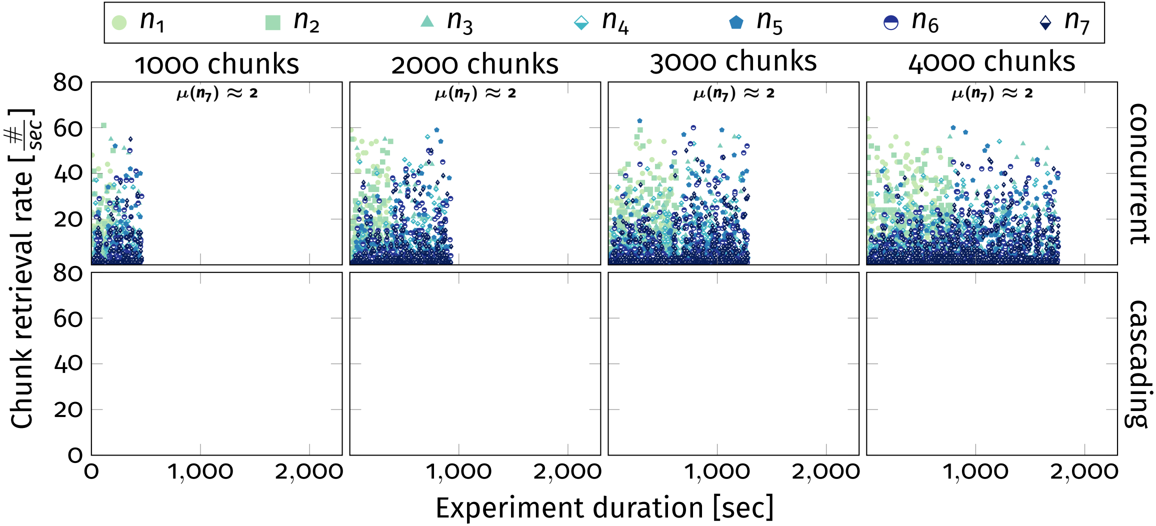


Relaxed network utilization
with cascading retrievals

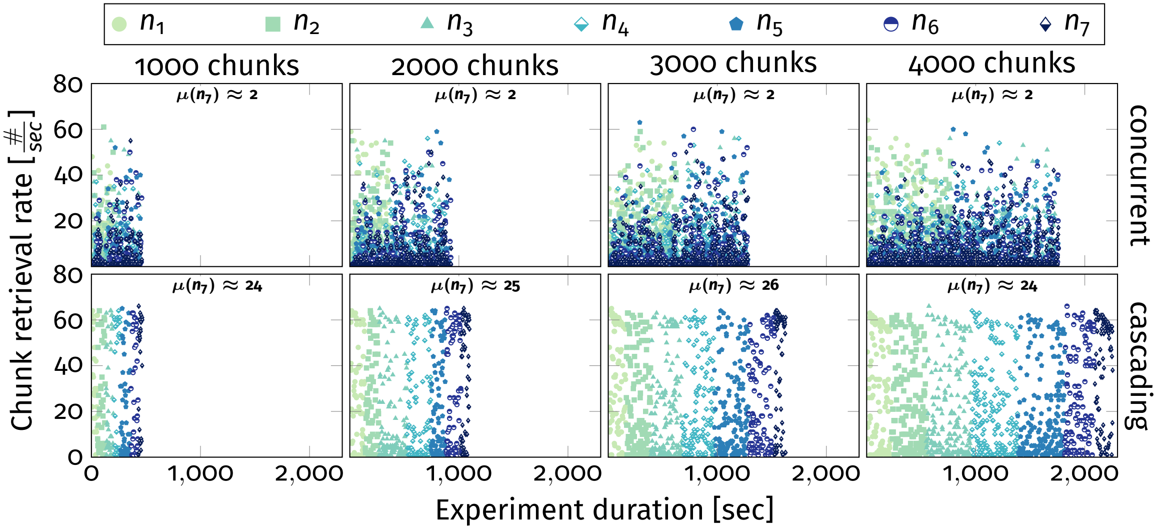
Goodput Analysis



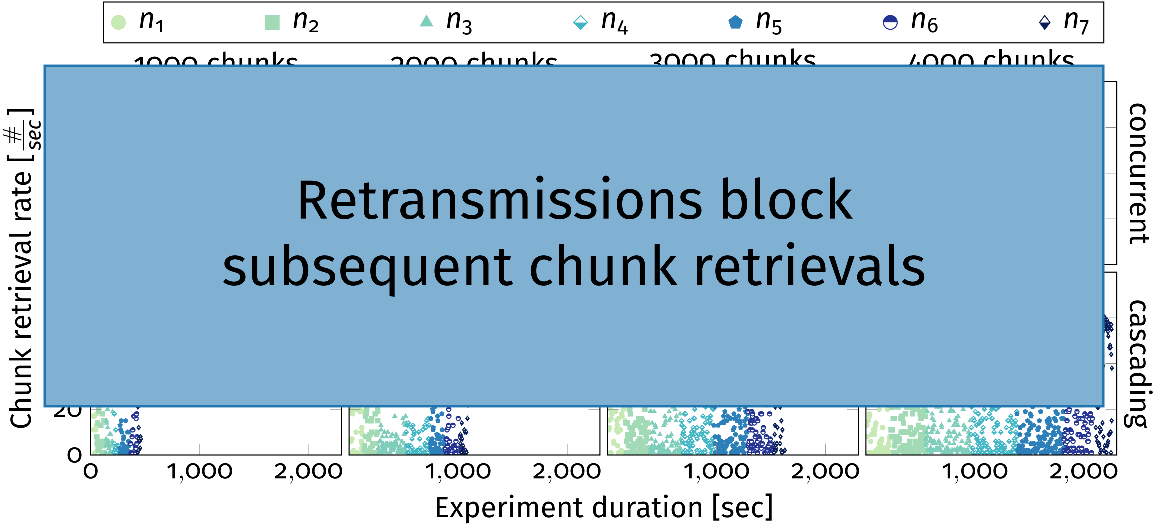
Goodput Analysis



Goodput Analysis

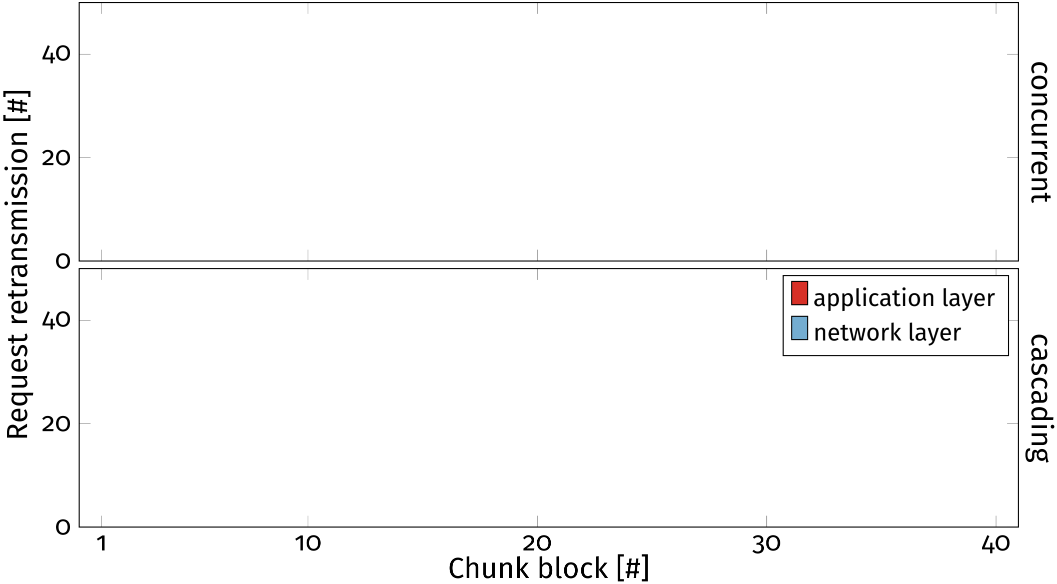


Goodput Analysis

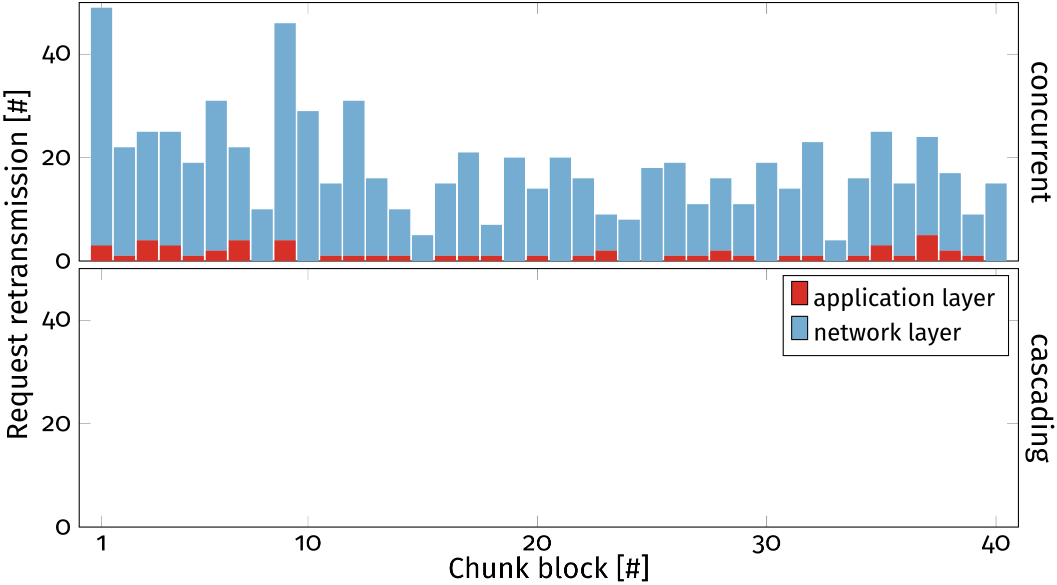


Retransmissions block subsequent chunk retrievals

Link Stress

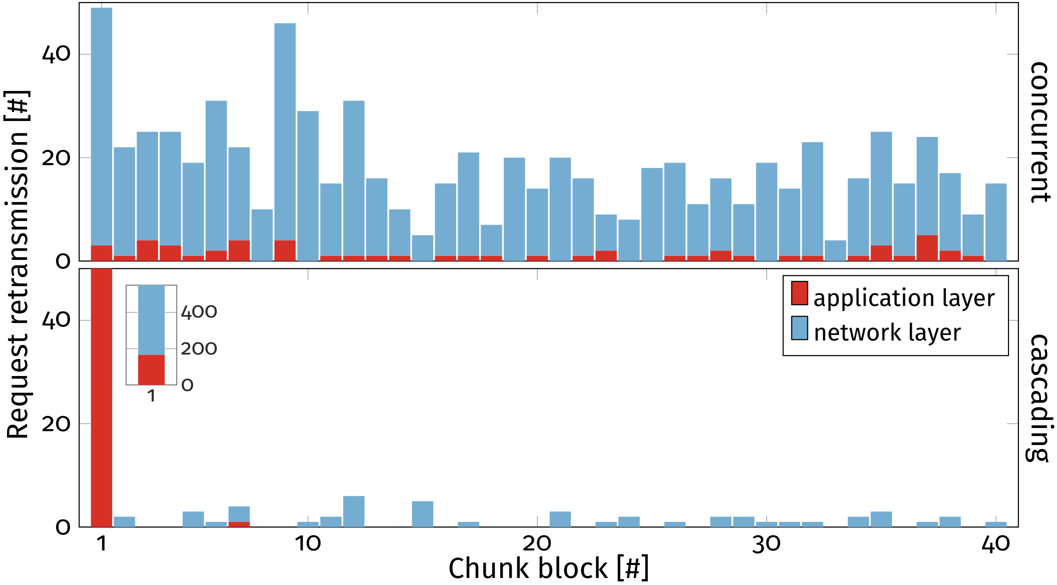

Request retransmission [#]
40
20
0
40
20
0
concurrent
cascading
🟥 application layer
🟦 network layer
1
10
20
30
40
Chunk block [#]

Link Stress

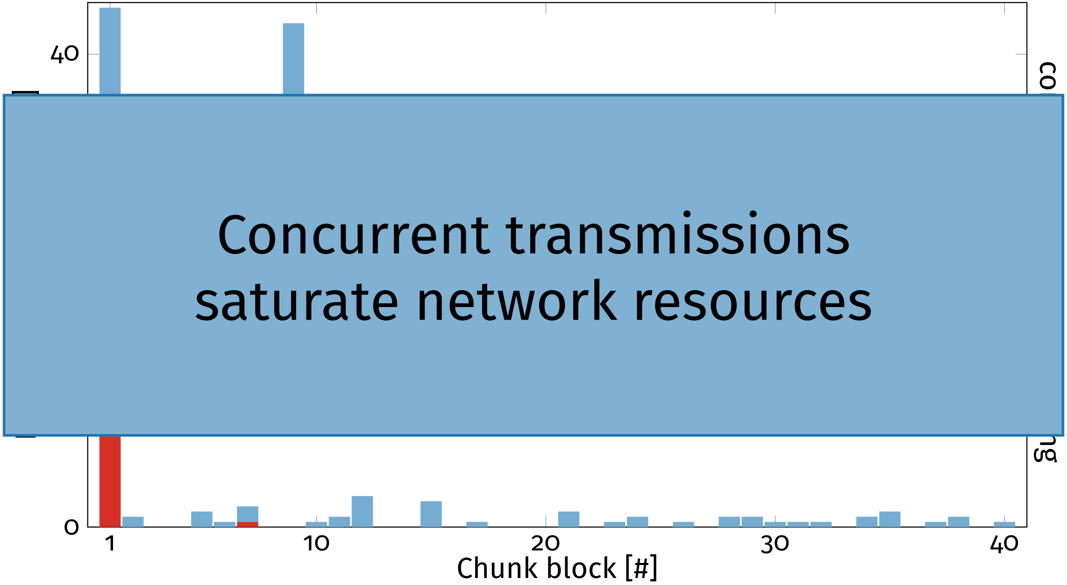


Request retransmission [#]
0
20
40
0
20
40
concurrent
cascading
application layer
network layer
1
10
20
30
40
Chunk block [#]

Link Stress

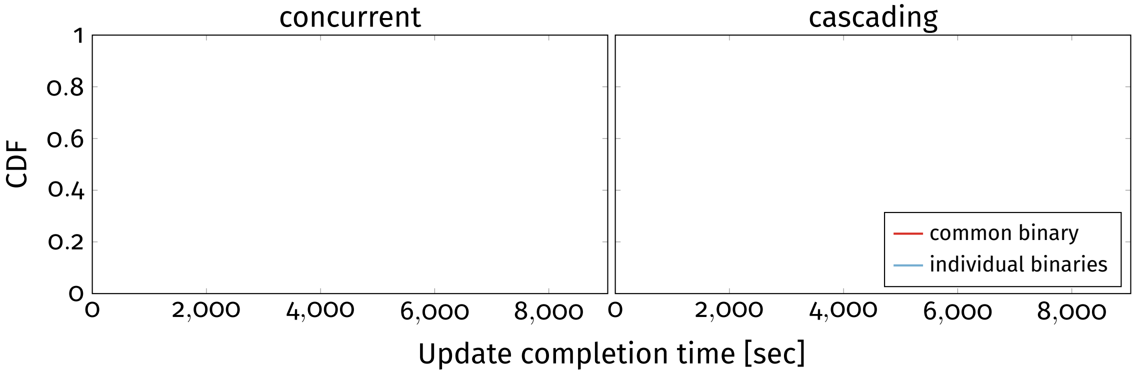

Request retransmission [#]
40
20
0
concurrent
400
200
0
1
40
20
0
application layer
network layer
cascading
1
10
20
30
40
Chunk block [#]

Link Stress

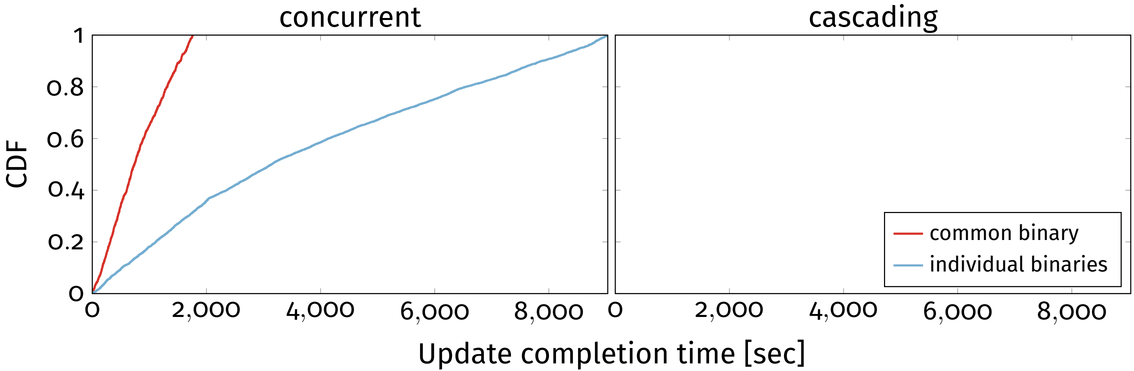


Concurrent transmissions saturate network resources

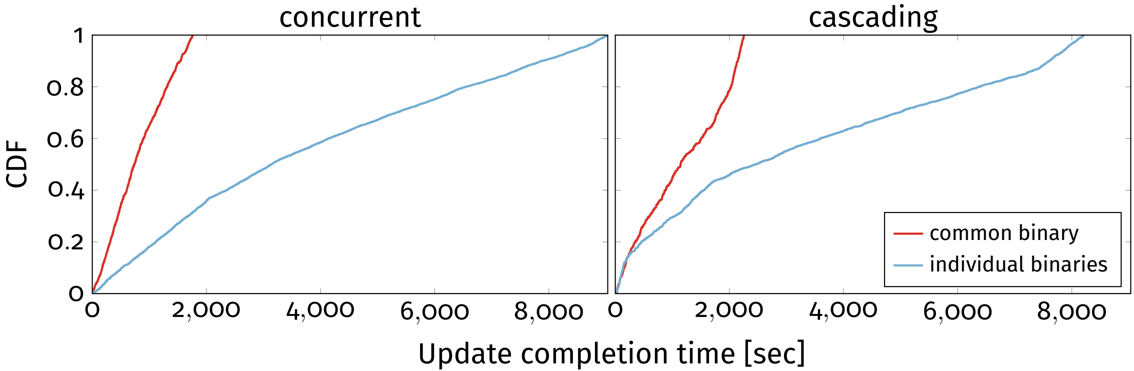
Multiparty Assessement



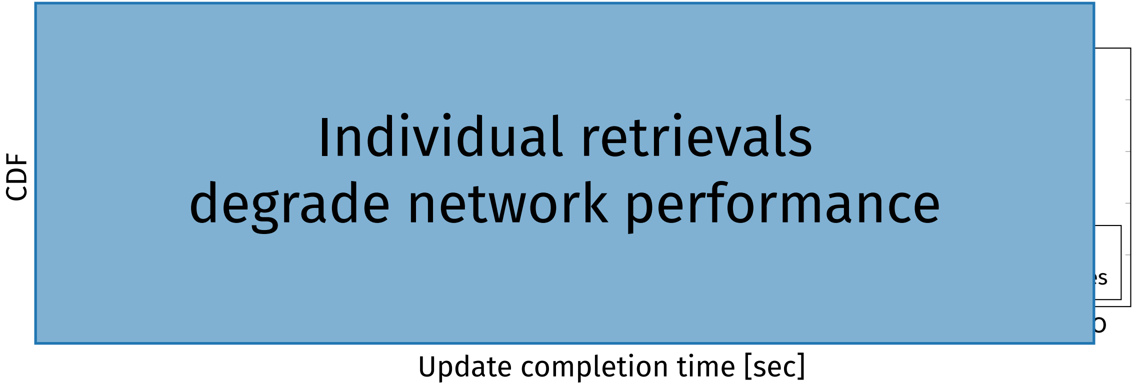
Multiparty Assessement



Multiparty Assessement



Multiparty Assessement



Individual retrievals
degrade network performance

Conclusion & Outlook

Conclusion & Outlook

Takeaways

- Inf.-centric content replication fosters efficient, reliable chunk dissemination
- Concurrent, uncoordinated distribution results in high link stress
- Cascading delivery relaxes strain on network resources
- Deployments with common binaries benefit from in-network caching

Next Steps

- Evaluate distribution of partial firmware updates
- Explore enhanced security measures to ease voluminous data transfers

Thank You!

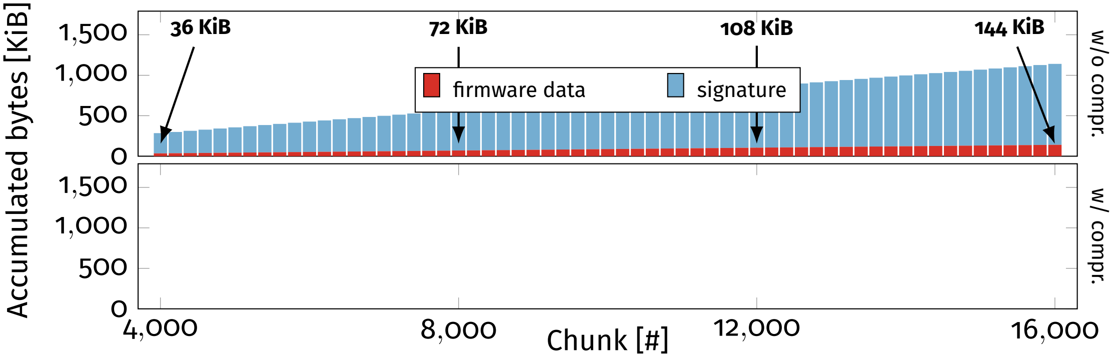
We support reproducible research.

`https://github.com/inetrg/ACM-ICN-2021-FWUPDATE`

Backup

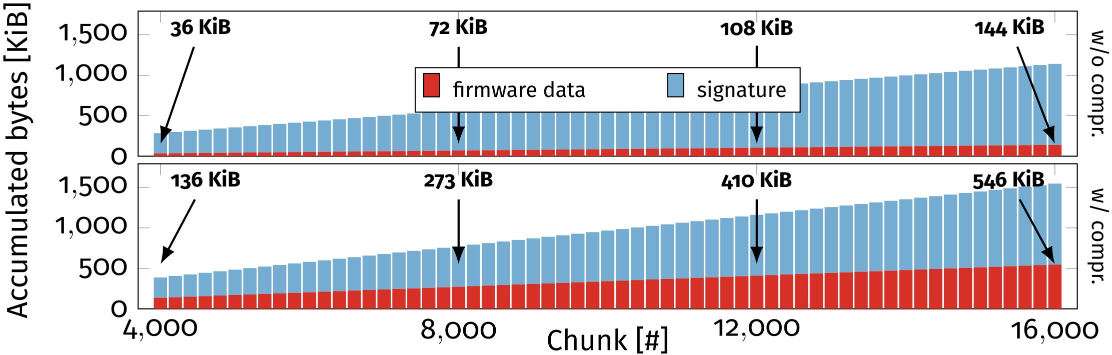
Overhead of Chunk-wise Signatures

- IEEE 802.15.4 MTU of 127 bytes with 23 MAC header overhead
- EdDSA with 64-byte signatures, NDN name and header with 31 bytes
- Payload size: 9 bytes (w/o ICNLoWPAN)



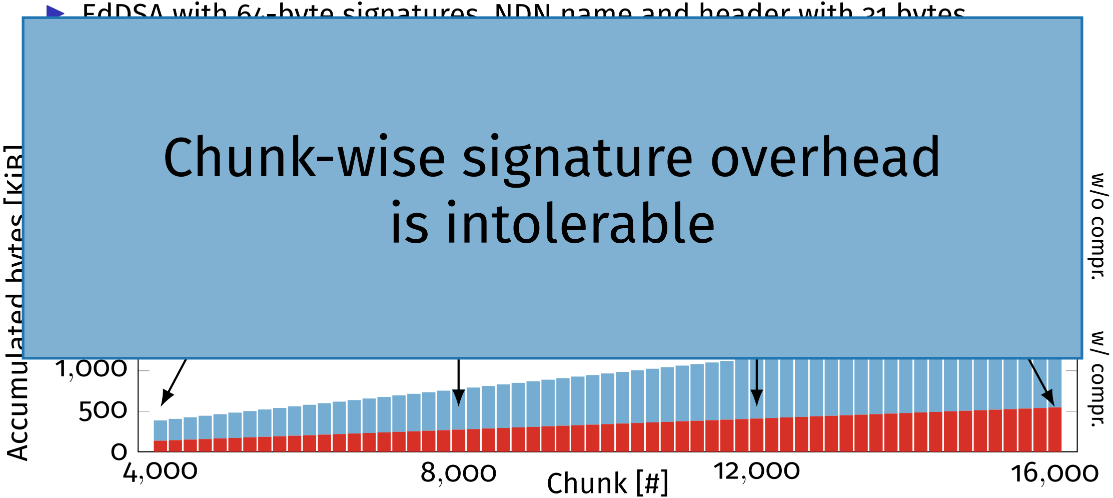
Overhead of Chunk-wise Signatures

- IEEE 802.15.4 MTU of 127 bytes with 23 MAC header overhead
- EdDSA with 64-byte signatures, NDN name and header with 31 bytes
- Payload size: 9 bytes (w/o ICNLoWPAN), 35 bytes (w/ ICNLoWPAN)



Overhead of Chunk-wise Signatures

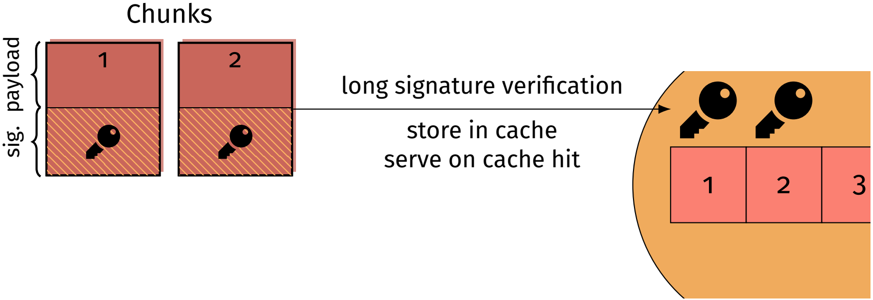
- IEEE 802.15.4 MTU of 127 bytes with 23 MAC header overhead
- EdDSA with 64-byte signatures, NDN name and header with 31 bytes



Chunk-wise signature overhead is intolerable

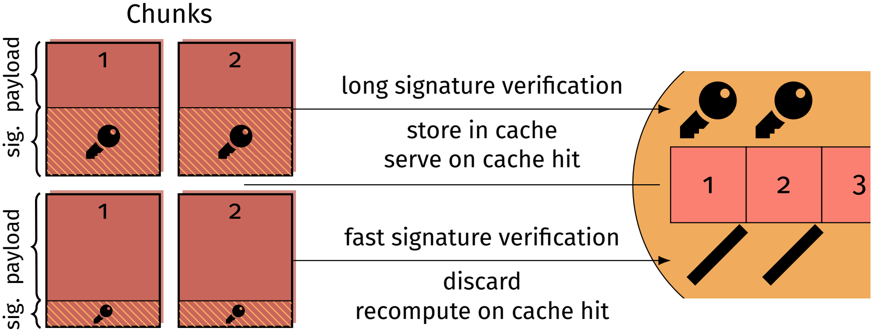
Enhanced Integrity Verification

- Asymmetric crypto is slow and energy-exhaustive (w/o crypto processor)
- Signatures reside in content store to serve cache hits for protected data



Enhanced Integrity Verification

- HMAC with pre-shared key is fast, but requires out-of-band channel
- Signatures are reproducible and can be discarded on reception


Chunks
payload
sig.
1
2
payload
sig.
1
2
long signature verification
store in cache
serve on cache hit
fast signature verification
discard
recompute on cache hit
1
2
3