Augustus: a CCN router for programmable networks

ACM ICN 2016, Kyoto

Davide Kirchner\(^1\)*, Raihana Ferdous\(^2\)*, Renato Lo Cigno\(^3\), Leonardo Maccari\(^3\), Massimo Gallo\(^4\), Diego Perino\(^5\)*, and Lorenzo Saino\(^6\)

September 27, 2016

\(^1\)Google Inc., Dublin, Ireland; \(^2\)Create-Net, Trento, Italy; \(^3\)DISI – University of Trento, Italy; \(^4\)Bell Labs – Nokia, Paris, France; \(^5\)Telefonica Research, Spain; \(^6\)Fastly, London, UK

*This work was done while D. Kirchner and R. Ferdous were at the University of Trento, and D. Perino and L. Saino at Bell Labs.
1. Introduction

2. The Augustus CCN router

3. Performance evaluation

4. Conclusions and lessons learned
Introduction
Objectives

The main goal is to explore the possibilities offered by modern general-purpose hardware in the context of information-centric networking:

- Implement a CCN data plane forwarder fully in software
  - Run on a commodity x86_64 machine
  - Performance-oriented, open-source and extensible
- Analyze the performance in a worst-case scenario
Objectives

The main goal is to explore the possibilities offered by modern general-purpose hardware in the context of information-centric networking:

- Implement a CCN data plane forwarder fully in software
  - Run on a commodity x86_64 machine
  - Performance-oriented, open-source and extensible
- Analyze the performance in a worst-case scenario

Why software router? Flexibility:

- Quicker development/deployment cycle and (re)configuration
- Hardware can be dynamically allocated to network functions

Tools

- Off-the-shelf high-performance hardware
- High-speed packet I/O libraries [Int, Riz12]
- Software routing frameworks built on top [BSM15, KJL+15]
Forwarding flow

- Focus on the Content Centric Networking approach [JST+09]
- Interests hold full content name
  - Similar to CCNx (vs NDN)
- CS and PIT: exact match
- Longest-prefix match at FIB

Example: get /com/updates/sw/v4.2.5.tar.gz

Router R2:
Forwarding information base (FIB)

| /com/updates | eth0 |

Pending Interest Table (PIT)

Content Store (CS)
Forwarding flow

- Focus on the Content Centric Networking approach [JST⁺09]
- Interests hold full content name
  - Similar to CCNx (vs NDN)
- CS and PIT: exact match
- Longest-prefix match at FIB

Example: get /com/updates/sw/v4.2.5.tar.gz

Router R2:

Forwarding information base (FIB)
/com/updates/sw/v4.2.5.tar.gz  eth0

Pending Interest Table (PIT)
/com/updates/sw/v4.2.5.tar.gz  {eth1}

Content Store (CS)
Forwarding flow

- Focus on the Content Centric Networking approach [JST+09]
- Interests hold full content name
  - Similar to CCNx (vs NDN)
- CS and PIT: exact match
- Longest-prefix match at FIB

Example: get /com/updates/sw/v4.2.5.tar.gz

Router **R2**:

- Forwarding information base (FIB)
  - /com/updates  eth0

- Pending Interest Table (PIT)

- Content Store (CS)
  - /com/updates/sw/v4.2.5.tar.gz  (data...)

Forwarding flow

- Focus on the Content Centric Networking approach [JST\textsuperscript{+}09]
- Interests hold full content name
  - Similar to CCNx (vs NDN)
- CS and PIT: exact match
- Longest-prefix match at FIB

Example: get /com/updates/sw/v4.2.5.tar.gz

Router R2:

Forwarding information base (FIB)

| /com/updates                  | eth0 |

Pending Interest Table (PIT)

| /com/updates/sw/v4.2.5.tar.gz | (data...) |

Content Store (CS)

| /com/updates/sw/v4.2.5.tar.gz | (data...) |
The Augustus CCN router
Design principles

- Exploit parallelism at all possible levels:
  - Hardware multi-queue at NIC
  - DRAM memory channels
  - Multiple cores on chip
  - Multiple NUMA sockets
- Data structures designed to match the x86 cache system
- Shared read-only FIB, duplicated in all NUMA sockets
- Sharded, thread-private CS and PIT
  - Exploit NIC’s Receive Side Scaling capabilities to dispatch incoming packets to threads
- Zero-copy packet processing
  - Based on DPDK for fast packet I/O [Int]
- Explored two trade-offs: max performance or more flexibility
Low-level standalone C implementation:

- Based on low-level optimized APIs
- Pushes the platform to its limits
- Architecture based on *Caesar* [PVL+14]
Design - modular

- Based on (Fast)Click [KMC\textsuperscript{+}00, BSM15]
- Easy to extend, experiment with
- Same optimized data structures
- Can be deployed aside other routing components

I = Interest Packet
D = Data Packet

FromDPDKDevice(n)

InputMux

Check ICNHeader

ICN_CS

ICN_PIT

ICN_FIB

Discard

ToDPDKDevice(n)
Performance evaluation
Experimental setup

- Two twin machines, each with two 10Gbps Ethernet ports
- Measurements expressed in data packets per second
- Work in slight overload conditions

Worst-case assumptions:

- Every interest packet has a unique name: no CS hits, no PIT aggregation
- Minimal-sized packets, to stress the forwarding engine
Threads and core mapping

Threads are pinned to processing cores
Test servers: 2 sockets × 8 cores × 2 (hyperthreading)
Threads and core mapping

Threads are pinned to processing cores

Test servers: 2 sockets \( \times \) 8 cores \( \times \) 2 (hyperthreading)
Threads and core mapping

Threads are pinned to processing cores

Test servers: 2 sockets × 8 cores × 2 (hyperthreading)
Threads and core mapping

Threads are pinned to processing cores
Test servers: 2 sockets × 8 cores × 2 (hyperthreading)
Threads and core mapping

Threads are pinned to processing cores
Test servers: 2 sockets × 8 cores × 2 (hyperthreading)
Threads and core mapping

Threads are pinned to processing cores
Test servers: 2 sockets × 8 cores × 2 (hyperthreading)
Threads and core mapping

Threads are pinned to processing cores
Test servers: 2 sockets × 8 cores × 2 (hyperthreading)
Threads and core mapping

Threads are pinned to processing cores
Test servers: 2 sockets \times 8 \text{ cores} \times 2 \text{ (hyperthreading)}
Standalone performance

- 2 threads: large gap hyperthreaded vs physical cores
- Best performance: 4 threads (dual socket), 8 threads (single/dual)
Click module performance

- 1 thread: same cache miss ratio, half performance
- Best performance: 16 threads
FIB size scaling

Data throughput [Mpps]
- Standalone, 8 threads
- Standalone, 4 threads
- Click module, 16 threads
- Standalone, 1 thread
- Click module, 1 thread

Cache miss ratio

Number of FIB buckets
Conclusions and lessons learned
Conclusions and lessons learned

Present *Augustus*, a CCN software router which:

- Forwards packets at more than 10 millions data packets per second and supports a FIB with up to $2^{26}$ entries, and it is able to saturate the 10 Gbit/s link with Ethernet payloads as small as 87 bytes;
  - Tested with a thorough worst-case oriented performance evaluation
- Runs both as a stand-alone system, achieving the best performance, or as a set of elements in the Click modular router framework
- Is open source and can be used in software based networks for fast and incremental ICN deployment

Lessons learned:

- Manual configuration for best performance
- Abstraction hides critical low level properties
- Complex zero-copy in modular framework
Conclusions and lessons learned

Present *Augustus*, a CCN software router which:

- Forwards packets at more than 10 millions data packets per second and supports a FIB with up to $2^{26}$ entries, and it is able to saturate the 10 Gbit/s link with Ethernet payloads as small as 87 bytes;
  - Tested with a thorough worst-case oriented performance evaluation
- Runs both as a stand-alone system, achieving the best performance, or as a set of elements in the Click modular router framework
- Is open source and can be used in software based networks for fast and incremental ICN deployment

Lessons learned:

- Manual configuration for best performance
- Abstraction hides critical low level properties
- Complex zero-copy in modular framework
Augustus: a CCN router for programmable networks

ACM ICN 2016, Kyoto

September 27, 2016

Thanks for your attention

davkir@google.com
Bibliography
Fast userspace packet processing.  

[Int] Intel®.  
DPDK: Data plane development kit.  

Networking named content.  

**Nba (network balancing act): A high-performance packet processing framework for heterogeneous processors.**


**The Click modular router.**


**Caesar: A Content Router for High-speed Forwarding on Content Names.**
[Riz12] Luigi Rizzo.

netmap: A novel framework for fast packet I/O.