Aalto University
School of Electrical
Engineering

Performance of Software Switching

Based on papers in IEEE HPSR 2011 and IFIP/ACM Performance 2011

Nuutti Varis, Jukka Manner

Department of Communications and Networking (COMNET)

Aalto University
School of Electrical
Engineering

Agenda

- Motivation
- Performance Aspects
- Software Switching
- Evaluation Results
- Conclusion

Motivation

- Commodity hardware is emerging as an option for specialized networking devices
  - A cheap and flexible solution
  - Acceptable performance up to ~40Gbps and higher
  - Commercial products are available
- Excellent platform for network protocol research
  - Implement and experiment with new protocols in the ”real world”
  - Debugging in user space is easy
  - Performance with kernel module or specialized I/O engine

Motivation

- While moving from 1Gbps ports to 10Gbps ports, we wanted to also test the new Intel Sandy Bridge microarchitecture
- More specifically, we were interested in three questions

1. How do the improvements of the Sandy Bridge microarchitecture affect throughput?
2. What is the effect of specialized packet processing software on throughput?
3. What kind of throughput can be expected from a high-end single CPU setup with current hardware and software?

Performance Aspects

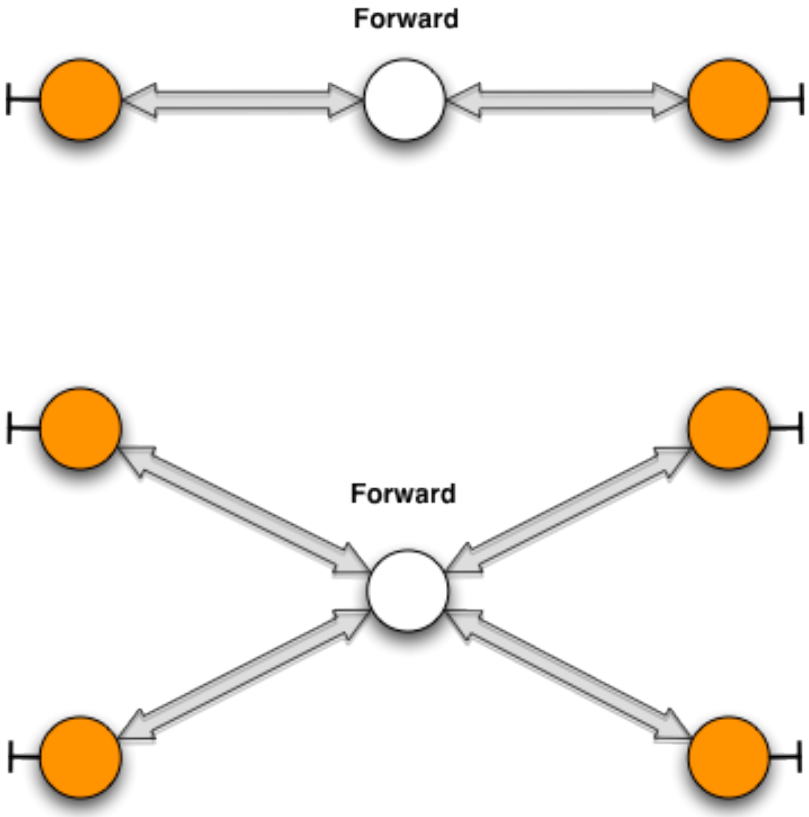
- Both hardware and software of the platform have key roles in the total device performance
- Hardware features
  - Direct Memory Access
  - Multi-core processors with on-die caches
  - Modern (point to point) I/O buses (QPI, PCI-E, …)
  - Multi-queue network interface cards
- Software-driven features
  - Interrupt scheme (NAPI, softirqs, polling, …)
  - Batching
  - Memory management (huge buffers, buffer pools, …)
  - Thread/process affinity
  - Data locality

Software Switching

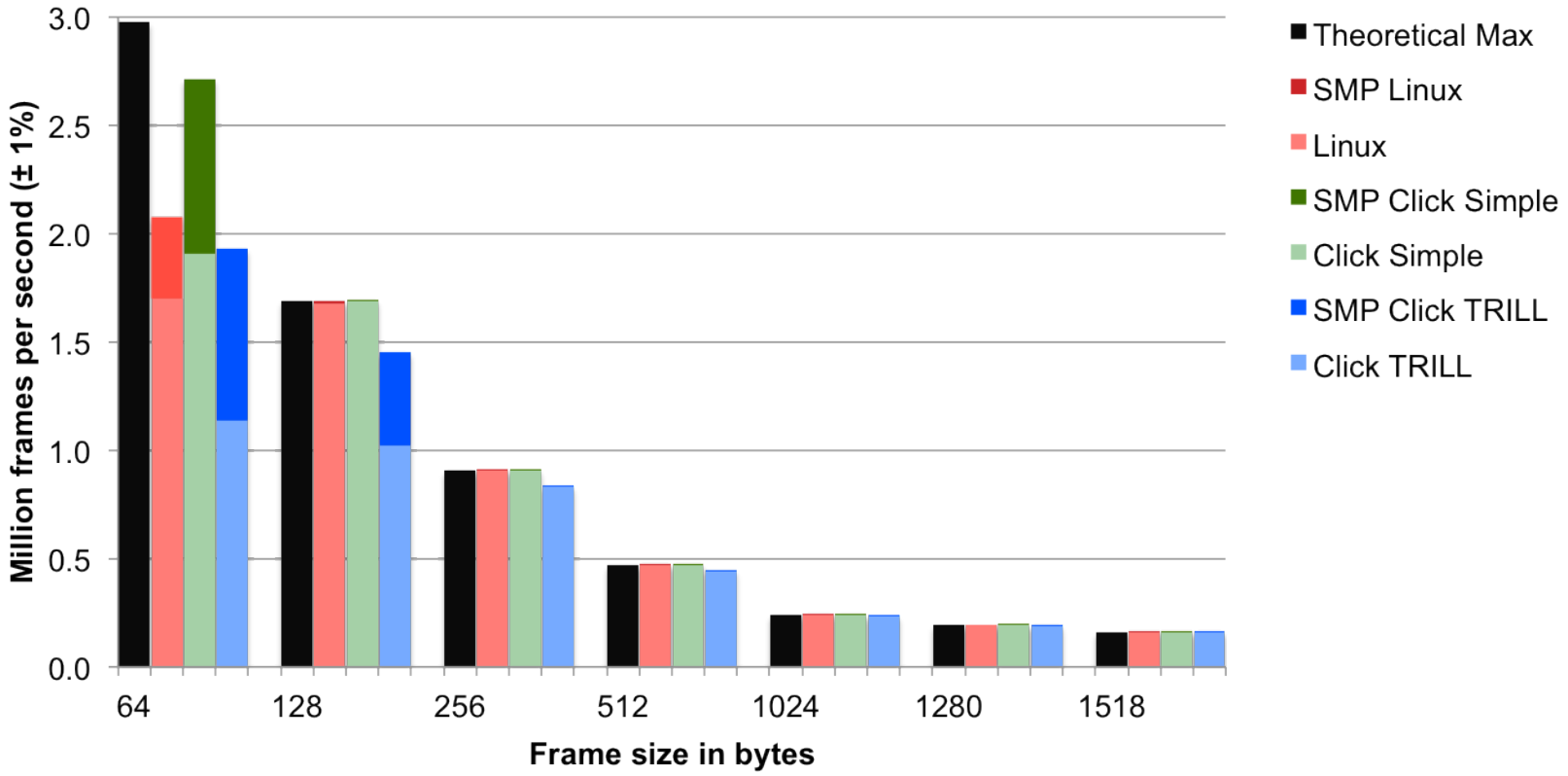
- Linux implements 802.3d as a kernel module
  - Network I/O using normal Linux network stack
- PacketShader has a Packet-IO concept that seeks to minimize memory copy when forwarding packets
- Click modular router is a framework for building network processing nodes
  - Functions in user space and as a (Linux) kernel module
- Click routers are composed of Elements
  - Network I/O, Processing, Monitoring, Classifying, …
- We have implemented TRILL as a set of Click Modular Router Elements
  - TRILL is a ”next generation” bridging protocol from IETF
    - Tunnels packets through the network
    - Brings features from layer three protocols to layer two, f.ex. hop counts and ECMP
  - Implementation emphasis on easy extendability and performance

Evaluation Setup

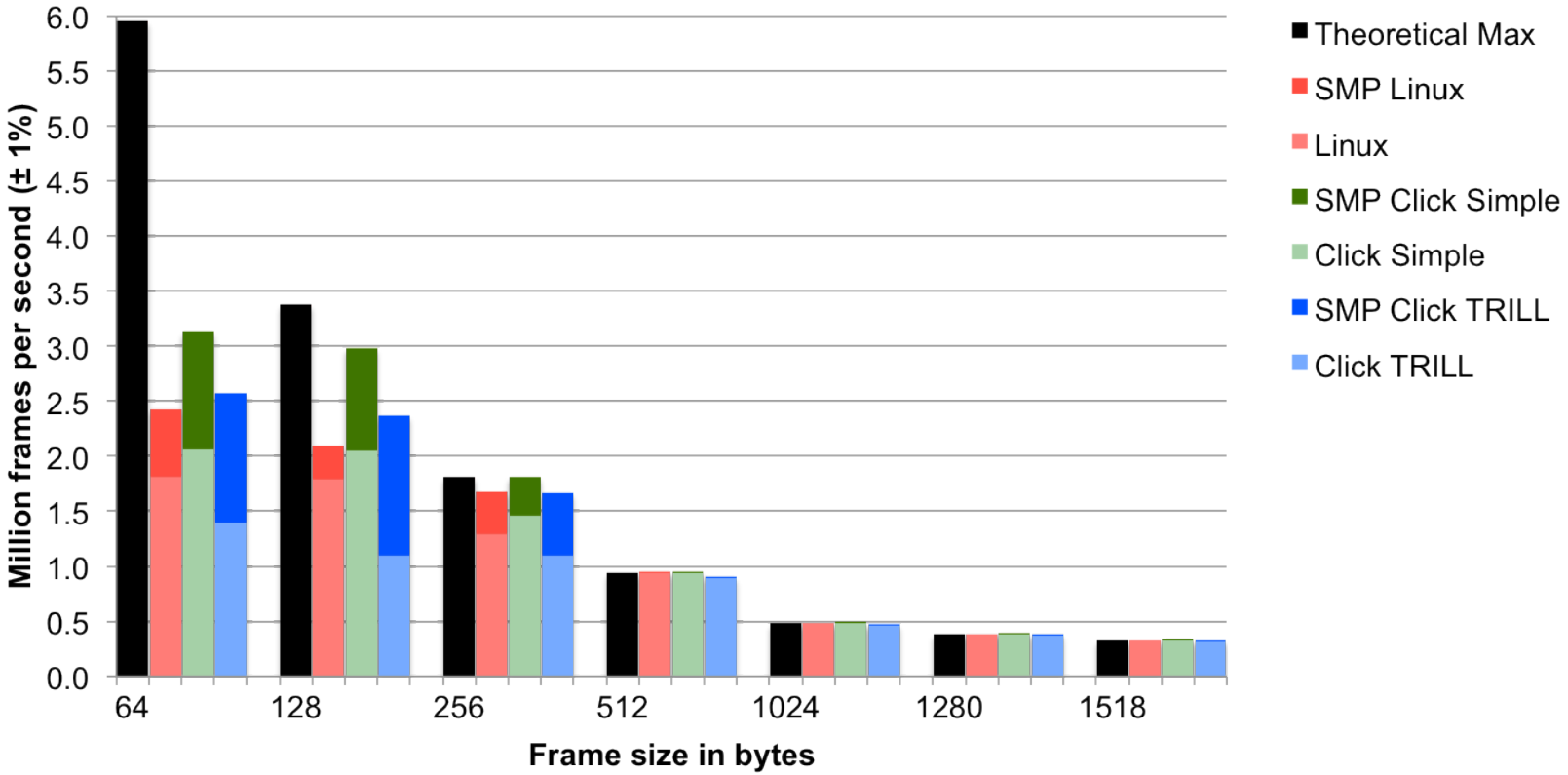
- Compare different software implementations
  - Linux Bridge
  - Click ”raw I/O”
  - PacketShader
  - Click with our TRILL implementation
- Quad core processors with either
  - 4 single queue 1Gbps ports
  - 4 10 Gbps ports
- Different topologies with 2 and 4 port of aggregated traffic
  - TRILL I/O performance evaluation requires both Edge- and Transit nodes
- Maximum lossless throughput in frames per second with various Ethernet payloads (RFC 2544)



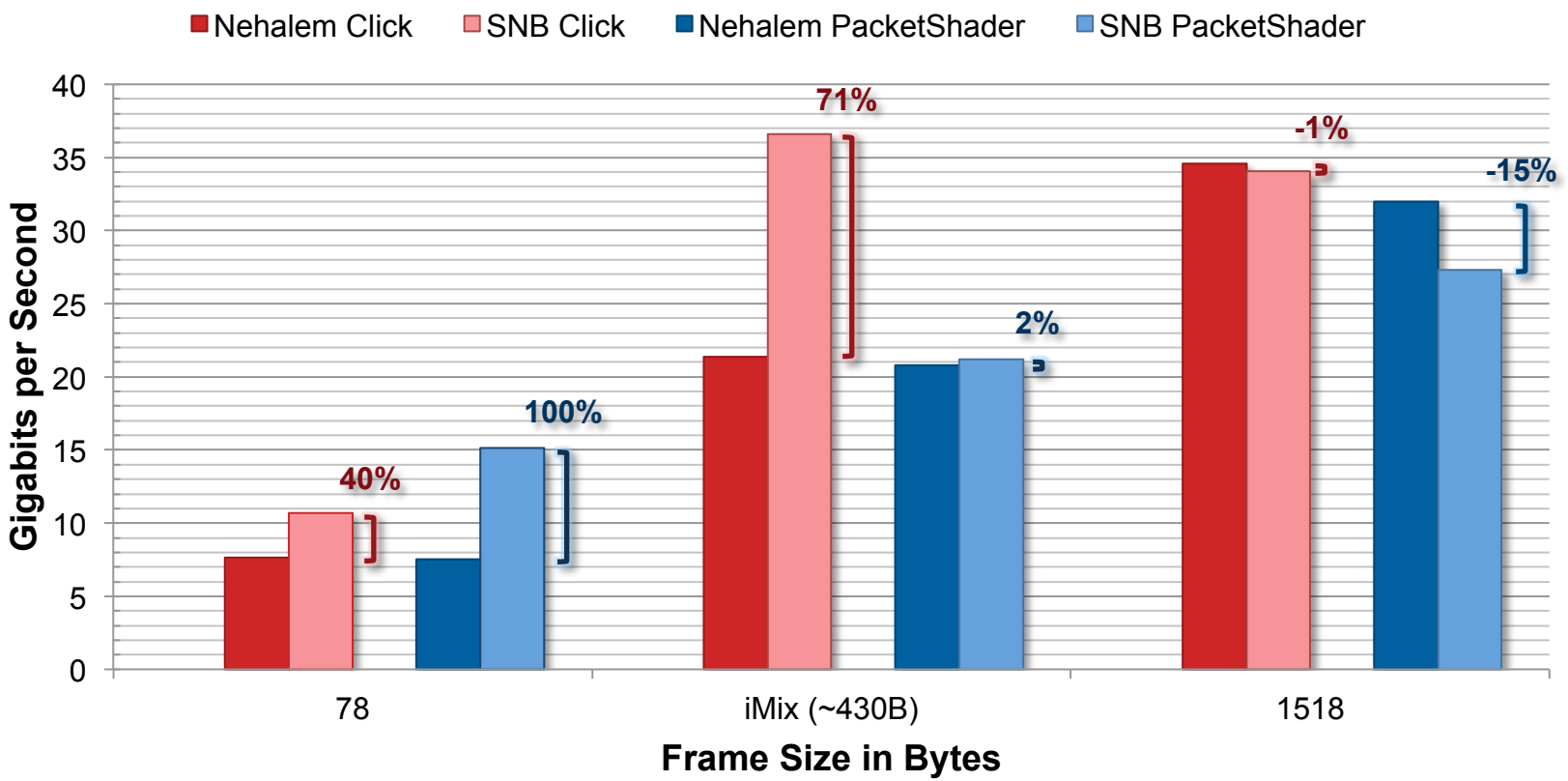
Measurement Results: One Bidirectional Flow Throughput (Nehalem, 1 Gbps)



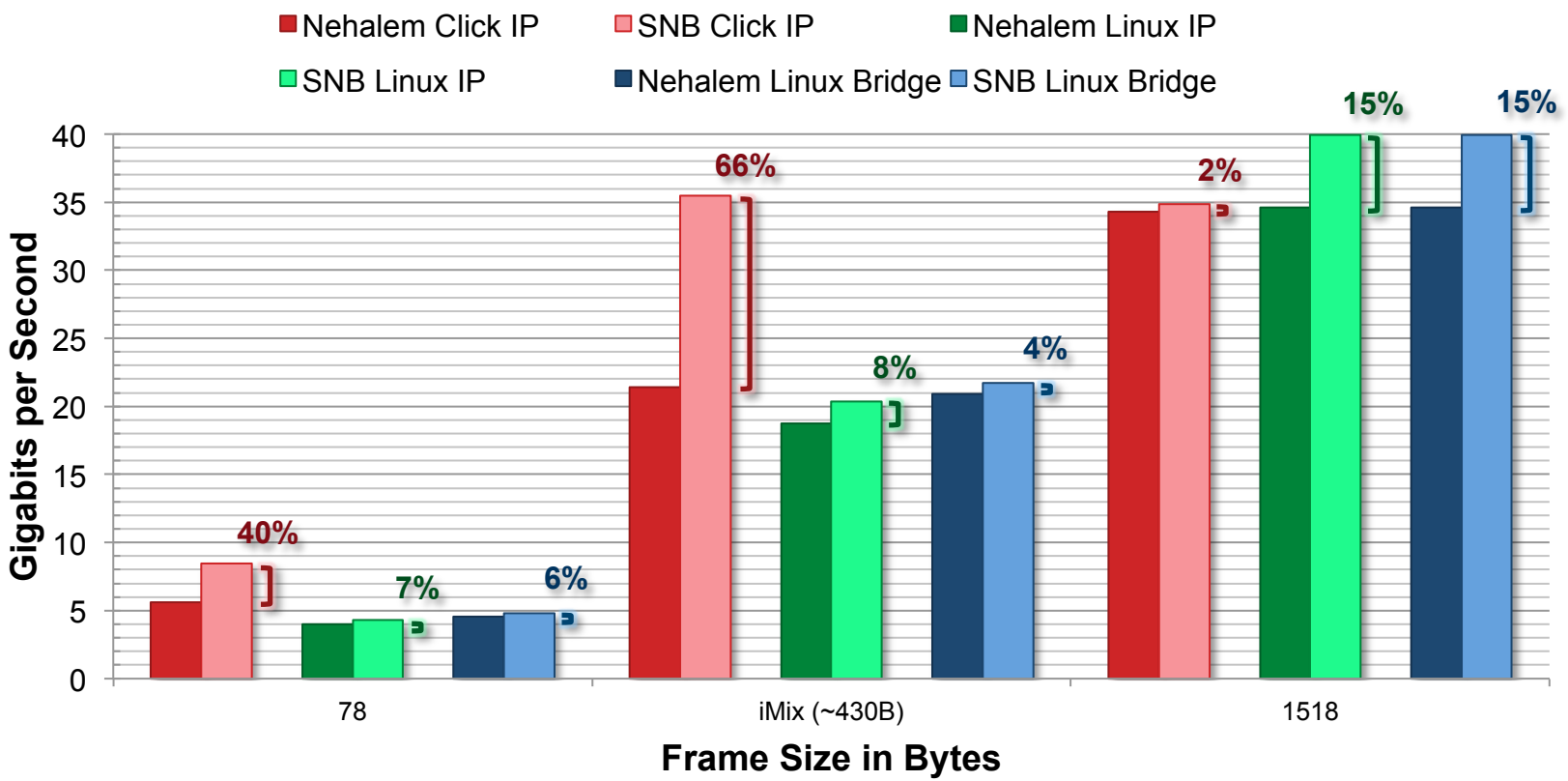
Measurement Results: Two Bidirectional Flows Throughput



Evaluation Results: Raw I/O Throughput



Evaluation Results: Processed I/O Throughput



Conclusion

- Thoughput does not scale linearly with small frame sizes in all techs
- Single queue network interface cards do not fully benefit from modern multi-core processors
- The Sandy Bridge platform excels especially with small frames when used with a specialized packet processing engine
- Linux network stack overhead eats most of the performance benefits of Sandy Bridge
- Our TRILL implementation performs similarly with Linux bridging component and Click raw I/O and benefits the most from parallelization
- The long term goal is to reach a point in performance and scaling, where the hardware limits are the definite bottleneck (but where?)