Modifying existing applications for 100 Gigabit Ethernet

Jelte Fennema

University of Amsterdam

29th June 2016

Introduction

- 100 Gigabit Ethernet (100GbE) is becoming common
- Measuring the network speed is important
- iperf3 is unable to saturate a 100GbE link
    - Can only reach ~45Gbit/s
    - CPU core is being maxed out

DPDK as a possible solution

- The Linux networking stack is too slow
- Possible solution: Data Plane Development Kit (DPDK)
  - Developed by Intel for very fast network I/O
  - Includes special high performance drivers

Linux networking

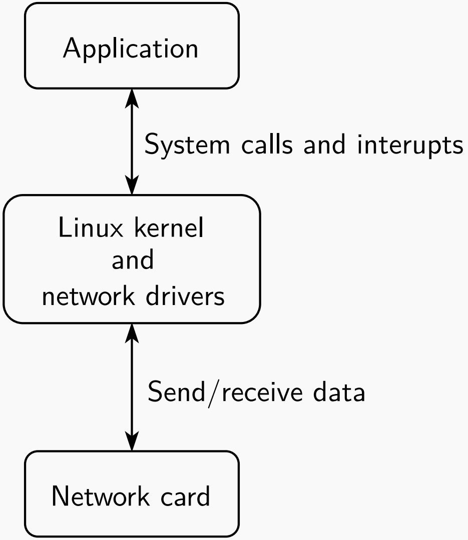

Figure 1: Normal Linux networking

DPDK networking

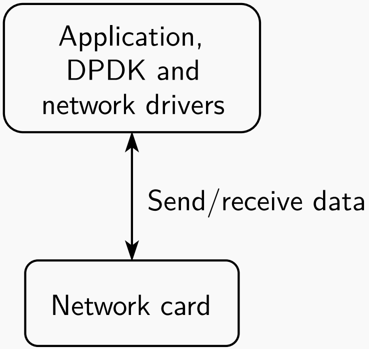

Figure 2: DPDK networking

Current DPDK packet generators

- Moongen
  - Achieved 120Gbit/s over multiple 10GbE interfaces
  - Doesn't support our Network Interface Card (NIC)
- Pktgen
  - Developed by Intel as official DPDK application
- Both have not been tested on 100GbE NICs

Research questions

1. Can current DPDK packet generators saturate a 100GbE link?
2. What is necessary to modify iperf3 to use DPDK?
3. What throughput improvements can be be achieved by modifying iperf3 to use DPDK?

Setup

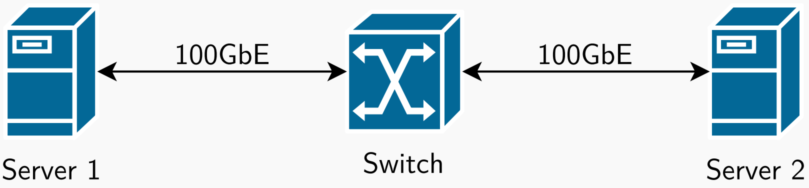

Figure 3: The test setup

Accelerated Network Stack

- iperf3 uses regular TCP connections
- DPDK itself can only be used for sending raw packets

Accelerated Network Stack

- iperf3 uses regular TCP connections
- DPDK itself can only be used for sending raw packets

- ANS is a FreeBSD networking stack modified for DPDK
  - Contains support for popular network protocols

New iperf3 versions

Two new iperf3 versions are created:

- One modified to use ANS
- A Linux version with comparable modifications

Focus

- TCP
- Single stream
    - Single core

Performance settings

- Setting CPU affinity
- `isolcpus`
- Disable hyperthreading

DPDK baseline

- Pktgen could reach 86Gbit/s
- This is for raw packets

iperf3 multi process baseline

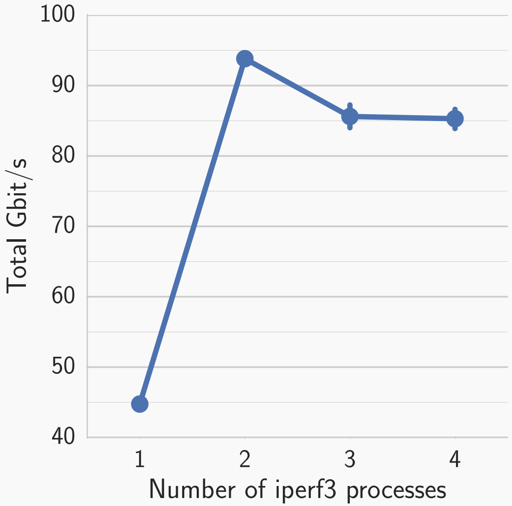

Figure 4: iperf3 speedtest with multiple processes

Modifications iperf3

- Event loop conversion from `select` to `epoll` style
- Removal of synchronous network I/O

Modifications to iperf3

Three iperf3 versions:

1. Regular
2. Epoll
3. ANS

Initial performance tests

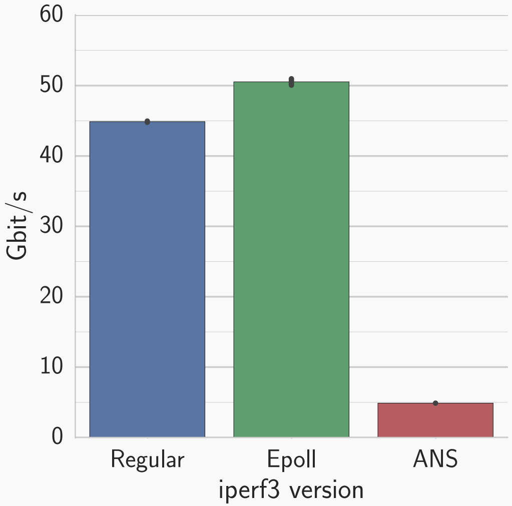

Figure 5: Initial performance comparison

Missing performance features

- TCP window scaling
- Jumbo frames are broken
- Offloading to the NIC

Performance with more streams

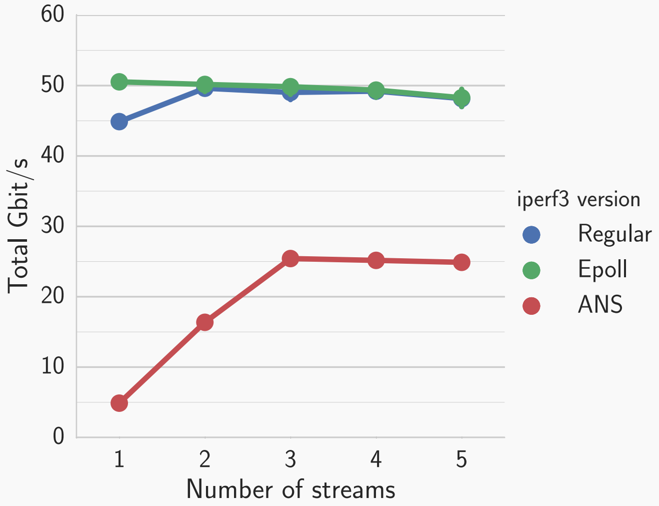

Figure 6: Performance with multiple TCP streams

Full impact of missing features

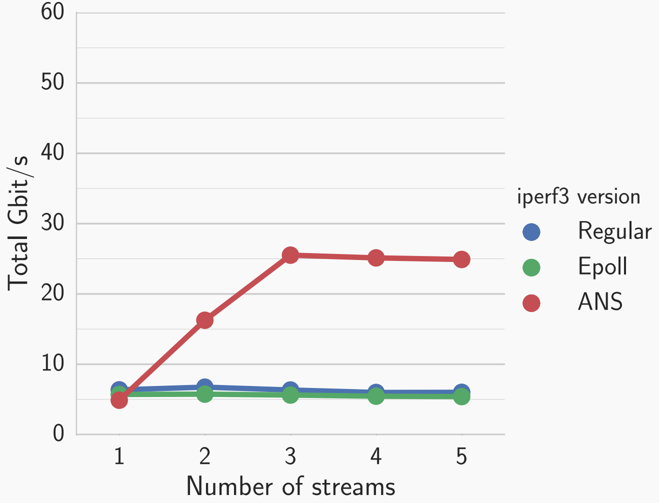

Figure 7: Performance comparison without missing ANS features

Final weird result

- Multiple streams improve single stream performance

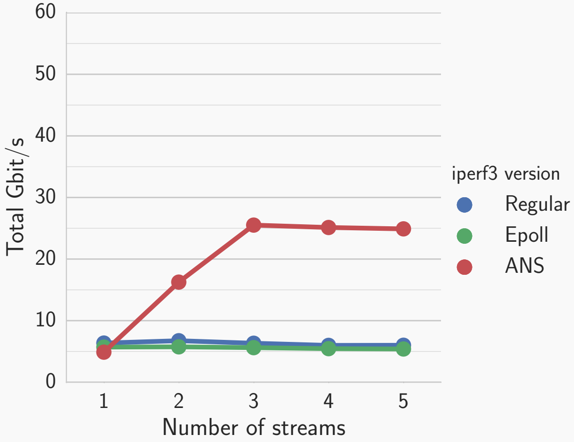

Figure 8: Performance comparison without missing ANS features

Modified transmit buffer length

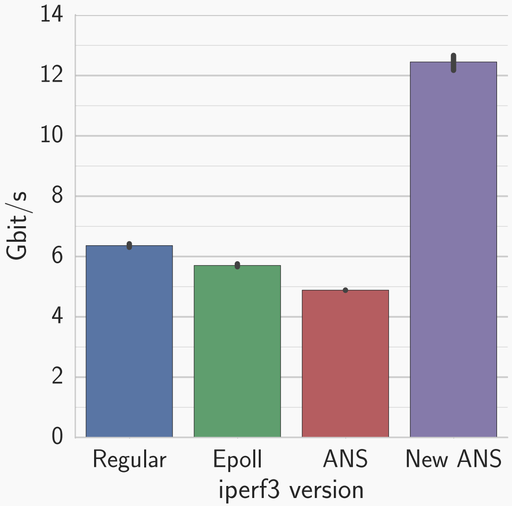

Figure 9: A single TCP stream with performance features disabled

Modified transmit buffer length

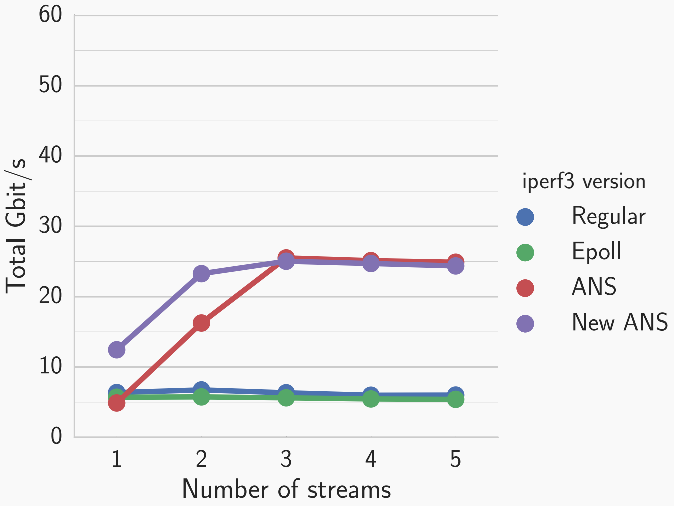

Figure 10: Multiple TCP streams with performance features disabled

Conclusion

- Pktgen was not able to fully fill the 100GbE link
    - But it was much faster than iperf3

Conclusion

- Pktgen was not able to fully fill the 100GbE link
    - But it was much faster than iperf3
- Modifying existing applications for DPDK is relatively easy by using ANS

Conclusion

- Pktgen was not able to fully fill the 100GbE link
  - But it was much faster than iperf3
- Modifying existing applications for DPDK is relatively easy by using ANS
- iperf3 speeds with ANS are currently slower than with Linux

Conclusion

- Pktgen was not able to fully fill the 100GbE link
    - But it was much faster than iperf3
- Modifying existing applications for DPDK is relatively easy by using ANS
- iperf3 speeds with ANS are currently slower than with Linux
- When missing ANS features are disabled for Linux ANS is faster

Conclusion

- Pktgen was not able to fully fill the 100GbE link
  - But it was much faster than iperf3
- Modifying existing applications for DPDK is relatively easy by using ANS
- iperf3 speeds with ANS are currently slower than with Linux
- When missing ANS features are disabled for Linux ANS is faster
- For multiple streams using multiple cores is probably easier

Future work

- Compare iperf3 performance after features have been implemented in ANS
- Investigate performance of Moongen on 100GbE