ETH zürich

D ITET

Networked Systems
ETH Zürich — seit 2015

# High-speed Traffic Generation

Semester Thesis Presentation

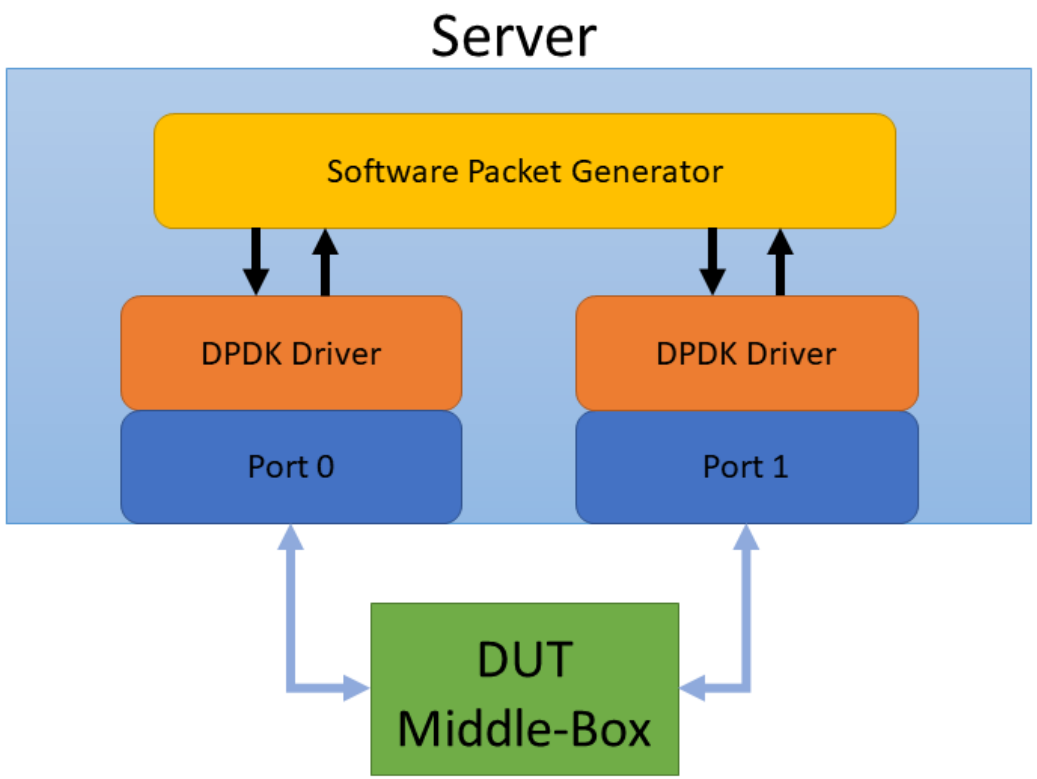
Author: Leonardo Rodoni

Tutor: Tobias Bühler

Supervisor: Prof. Dr. Laurent Vanbever

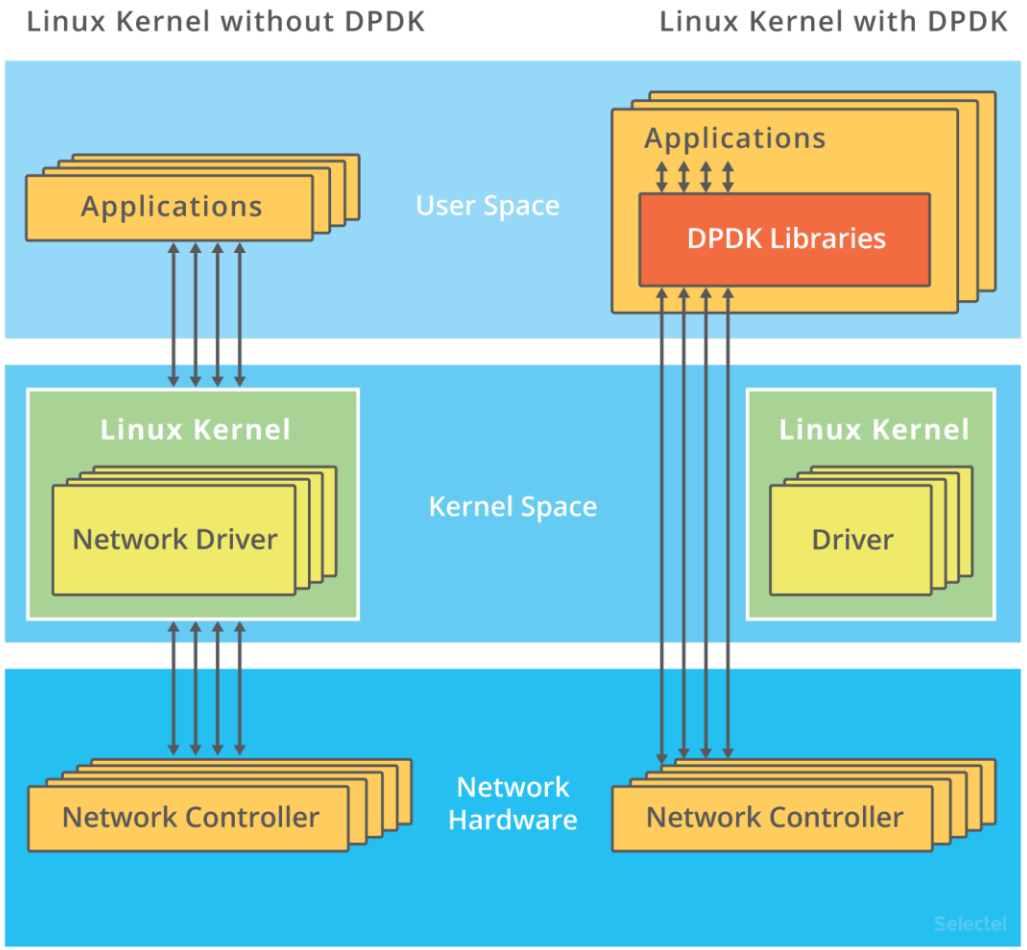
# Traffic Generators

- Software and Hardware solutions exist
- Used for benchmarking of new applications or devices (Router, switches, firewalls, reverse-proxies,…)
- Used to replicate real-life traffic (simulate cyber attacks, stress-testing,…)



# Modern software Traffic Generators

- Make use of another framework to access NICs (bypass Linux kernel networking stack)
- Most common framework is the Data Plane Development Kit (DPDK)
- Intel Project, now open source
- Widely compatible, scalable with number of cores



# Goals of the Thesis

- Analyse three modern software traffic generators
  - Moongen
  - Warp17
  - T-Rex
- Compare performance (on 10Gbps link) and available features

➡ Ultimate goal: provide insight and advice on the tool utilization, to help readers choose the right tool

# Moongen

- Simple, stateless traffic generator
- **Operation**: runs with Lua scripts

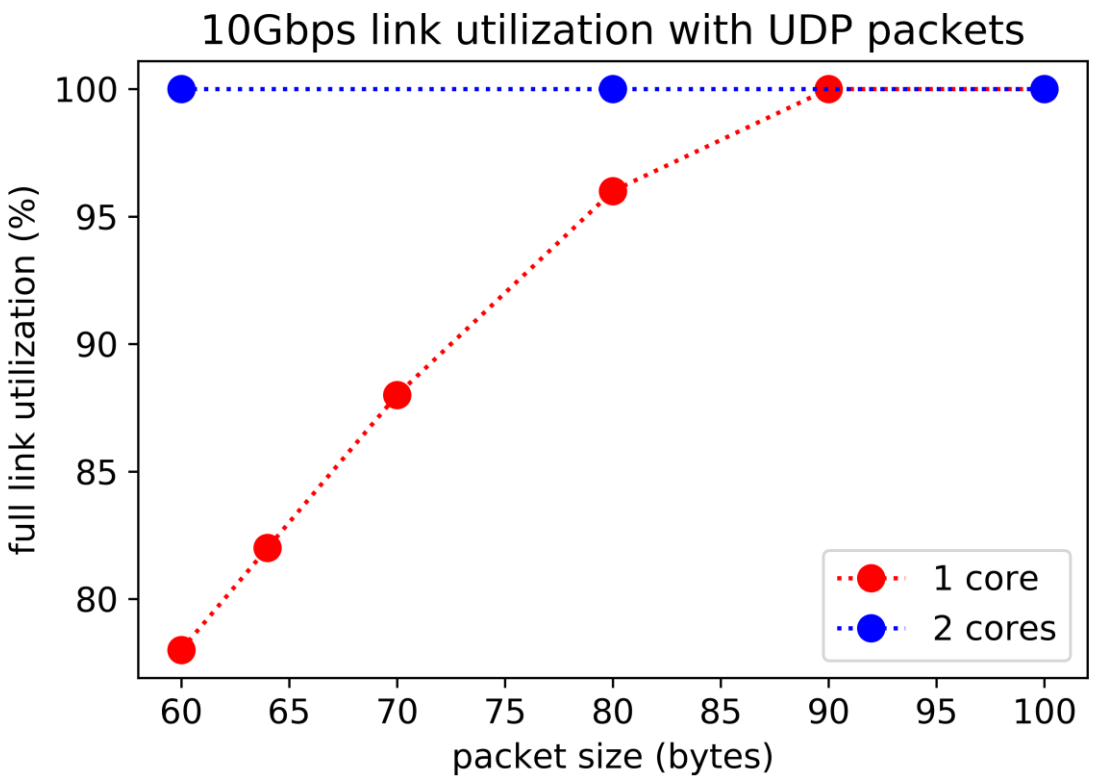
```
[Device: id=0] RX: 0.00 Mpps, 0 Mbit/s (0 Mbit/s with framing)
[Device: id=1] RX: 3.75 Mpps, 2042 Mbit/s (2643 Mbit/s with framing)
[Device: id=0] TX: 3.75 Mpps, 2042 Mbit/s (2643 Mbit/s with framing)
[Device: id=1] TX: 0.00 Mpps, 0 Mbit/s (0 Mbit/s with framing)
[Device: id=0] RX: 0.00 Mpps, 0 Mbit/s (0 Mbit/s with framing)
[Device: id=1] RX: 14.12 Mpps, 7681 Mbit/s (9941 Mbit/s with framing)
[Device: id=0] TX: 14.12 Mpps, 7682 Mbit/s (9941 Mbit/s with framing)
[Device: id=1] TX: 0.00 Mpps, 0 Mbit/s (0 Mbit/s with framing)
```

Figure: snapshot of Moongen's user interface with live statistics

- All protocols and traffic patterns can in theory be implemented on the Lua scripts, though more code → more overhead
- Statistic output also defined in Lua script

# Moongen 10Gbps link saturation

- Extremely efficient in terms of CPU utilization
- Saturates link with 60 bytes **Ethernet packets** using 1 core
- Saturates link with 90 bytes **UDP packets** with 1 core



# WARP17

THE STATEFUL TRAFFIC GENERATOR

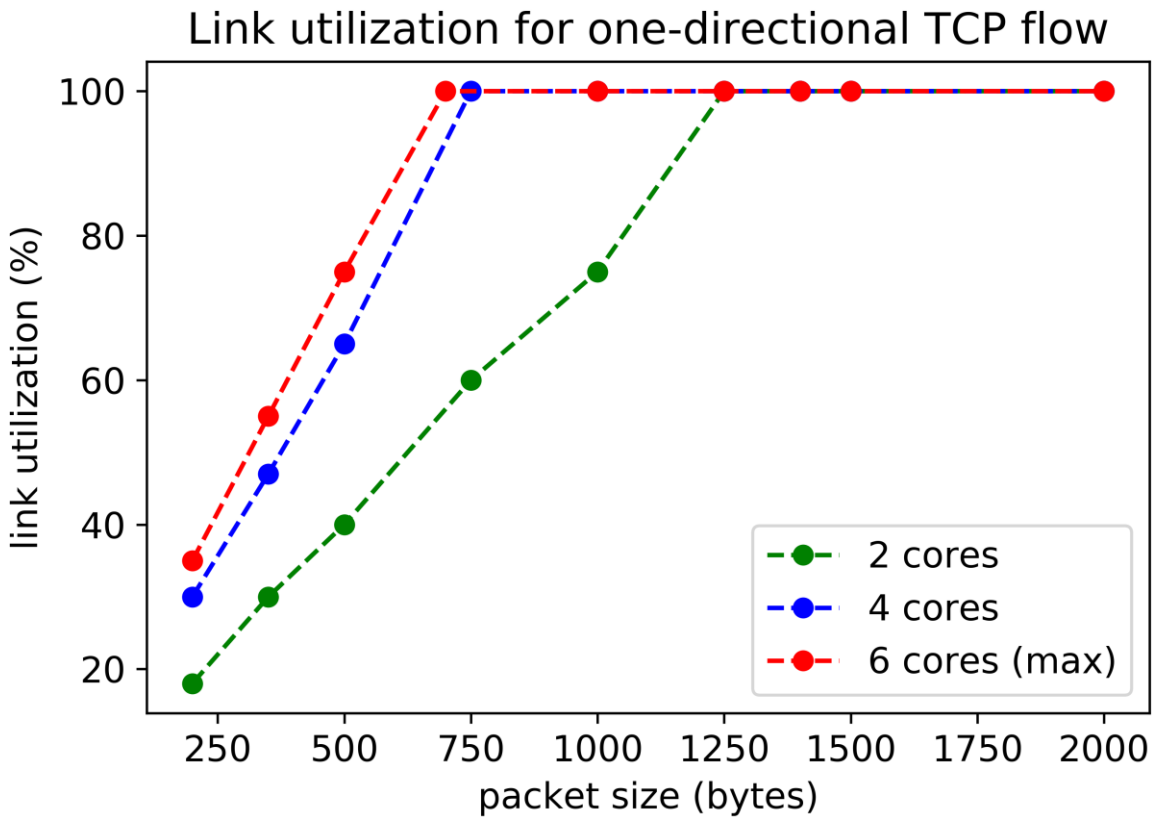
- Stateful traffic generator
- **Operation**: runs with CLI code, can be pre-defined in configuration file
- TCP full stack implemented over DPDK, with many configurable parameters
- Memory hugepages
- Internet-Mix traffic

```
Port 0: link UP, speed 10Gbps, duplex full(auto), TX: 68.38% RX: 68.5
=====================================================================
  CL s/s  Established       Closed         Data
 TCP/UDP         9991            0       903242

=====================================================================
TCP SM Stats
=====================================================================
INIT       :                0      LISTEN     :                0
SYN_SENT   :                1      SYN_RECV   :                0
ESTAB      :            65501      FIN_WAIT_1 :                0
FIN_WAIT_2 :                0      LAST_ACK   :                0
CLOSING    :                0      TIME_WAIT  :                0
CLOSE_WAIT :                0      CLOSED     :                0

=====================================================================
IP Stats
=====================================================================
Rx Pkts     :         12030255  Invalid Cksum   :      0
Rx Bytes    :       6457584200  Small Mbuf      :      0
Rx ICMP     :                0  Small Hdr       :      0
Rx TCP      :         12030255  Invalid Len     :      0
Rx UDP      :                0  Rx Frags        :      0
Rx Other    :                0
Invalid Ver :                0  Res Bit         :      0
Invalid Pad :                0  Invalid option  :      0

=====================================================================
Link Stats
=====================================================================
                      Link       Rate              SW
   Rx Pkts        12029197    1811406        12030258
  Rx Bytes      6661747296 1002784898      6662330242
   Tx Pkts        12194101    1821348        12194863
  Tx Bytes      6782000646 1005761689      6745784566
    Rx Err               0          0             N/A
    Tx Err               0          0               0
Rx No Mbuf               0          0             N/A
```

Figure: snapshot of Warp17's user interface with live statistics

# Warp17 link saturation with TCP traffic

- Test establishes 200k connections then sends data packets at maximum rate
- 2 cores always allocated for CLI



# TRex

Realistic Traffic Generator

- CISCO tool
- Stateful, Stateless and Advanced Stateful (ASTF) modes are available
- Tested mostly the Stateful mode
- ASTF mode has TCP stack implemented on DPDK

```
-Per port stats table
      ports |               0 |               1
 -----------------------------------------------------
   opackets |        74745383 |        73254195
     obytes |     76797918134 |      4688268480
   ipackets |        73254195 |        74745408
     ibytes |      4688268480 |     76797944584
    ierrors |               0 |               0
    oerrors |               0 |               0
      Tx Bw |       9.81 Gbps |     597.93 Mbps

-Global stats enabled
 Cpu Utilization : 70.6  %  7.4 Gb/core
 Platform_factor : 1.0
 Total-Tx        :      10.41 Gbps
 Total-Rx        :      10.41 Gbps
 Total-PPS       :       2.36 Mpps
 Total-CPS       :      10.92 Kcps

 Expected-PPS    :       4.86 Mpps
 Expected-CPS    :      22.50 Kcps
 Expected-BPS    :      21.45 Gbps

 Active-flows    :   497523  Clients :      508
 Open-flows      :   931357  Servers :    65532
 Total_queue_full : 95295405
 drop-rate       :       0.00  bps
 current time    : 80.3 sec
 test duration   : 0.0 sec
```

Figure: snapshot of T-Rex's user interface with live statistics

# Stateful T-Rex

- Stateful traffic generator based on replay of pcap files
- **Operation**: Runs with yaml configuration file
- No TCP stack implemented
- Only UDP and TCP packets in pcap file

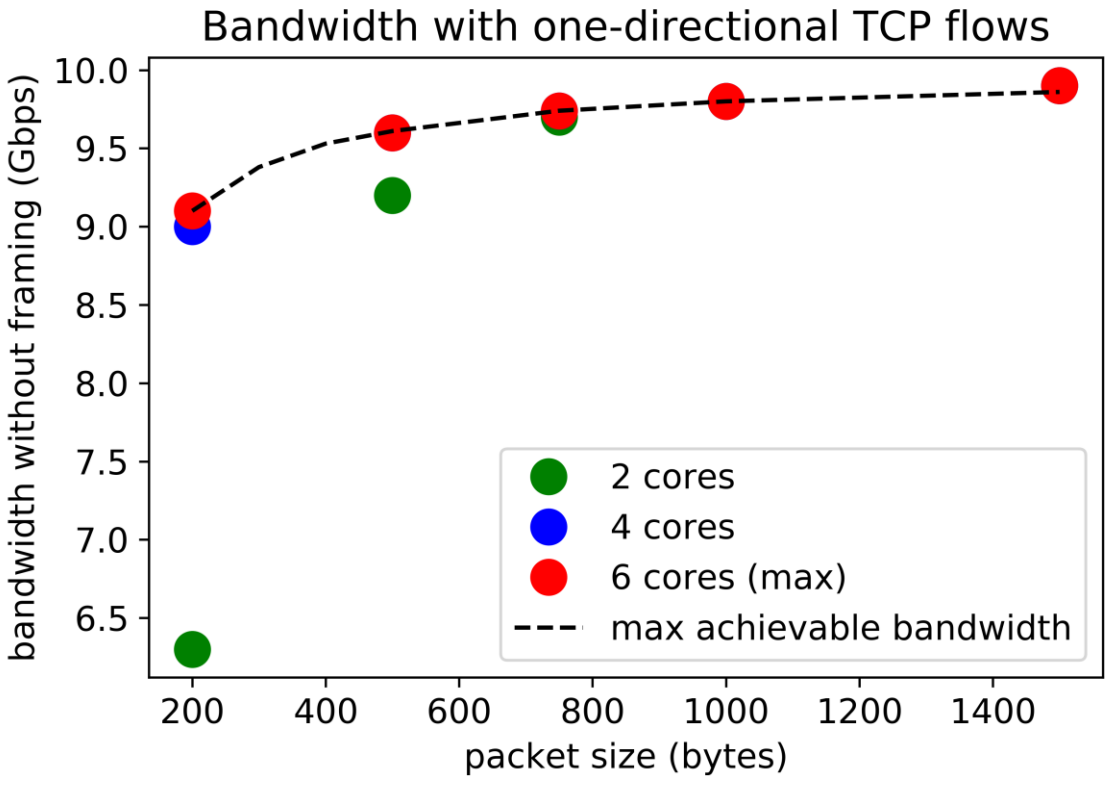
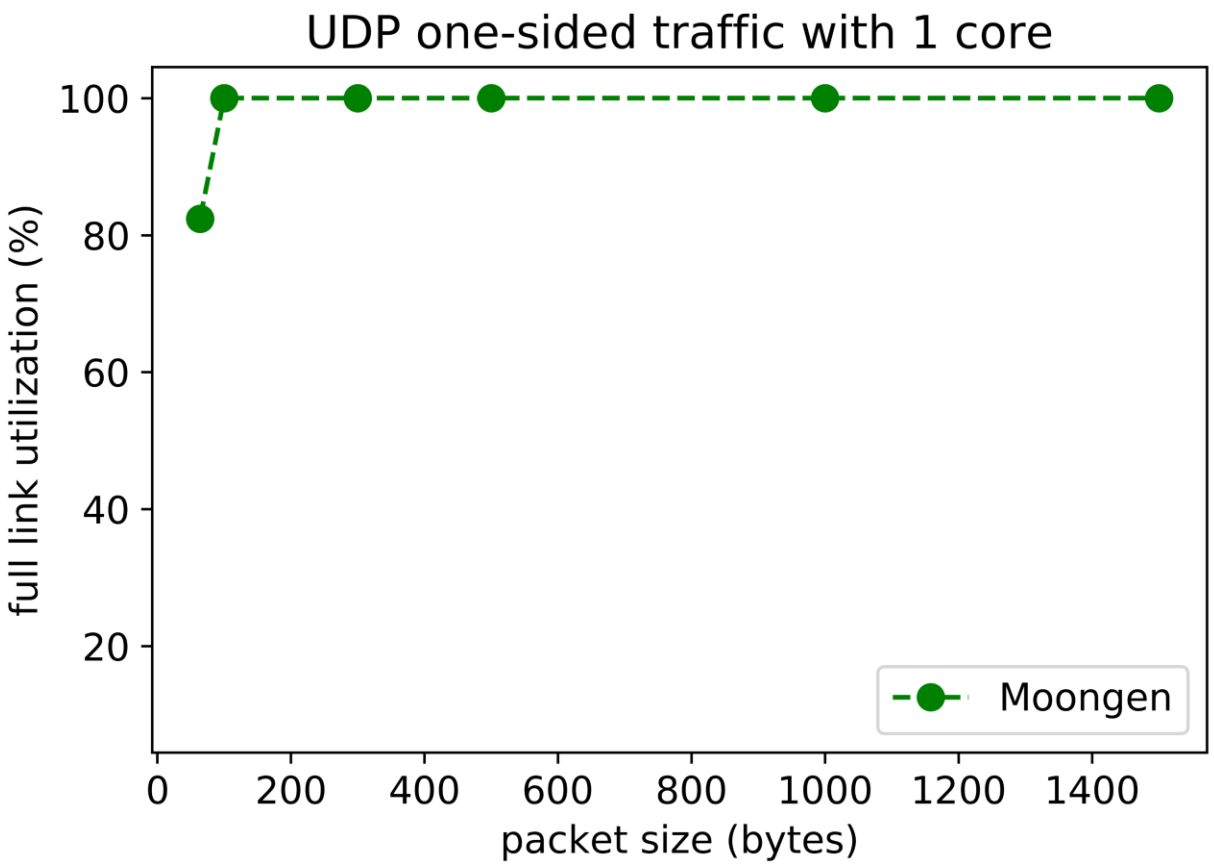



Figure: test with Warp17-generated pcap files

# Results

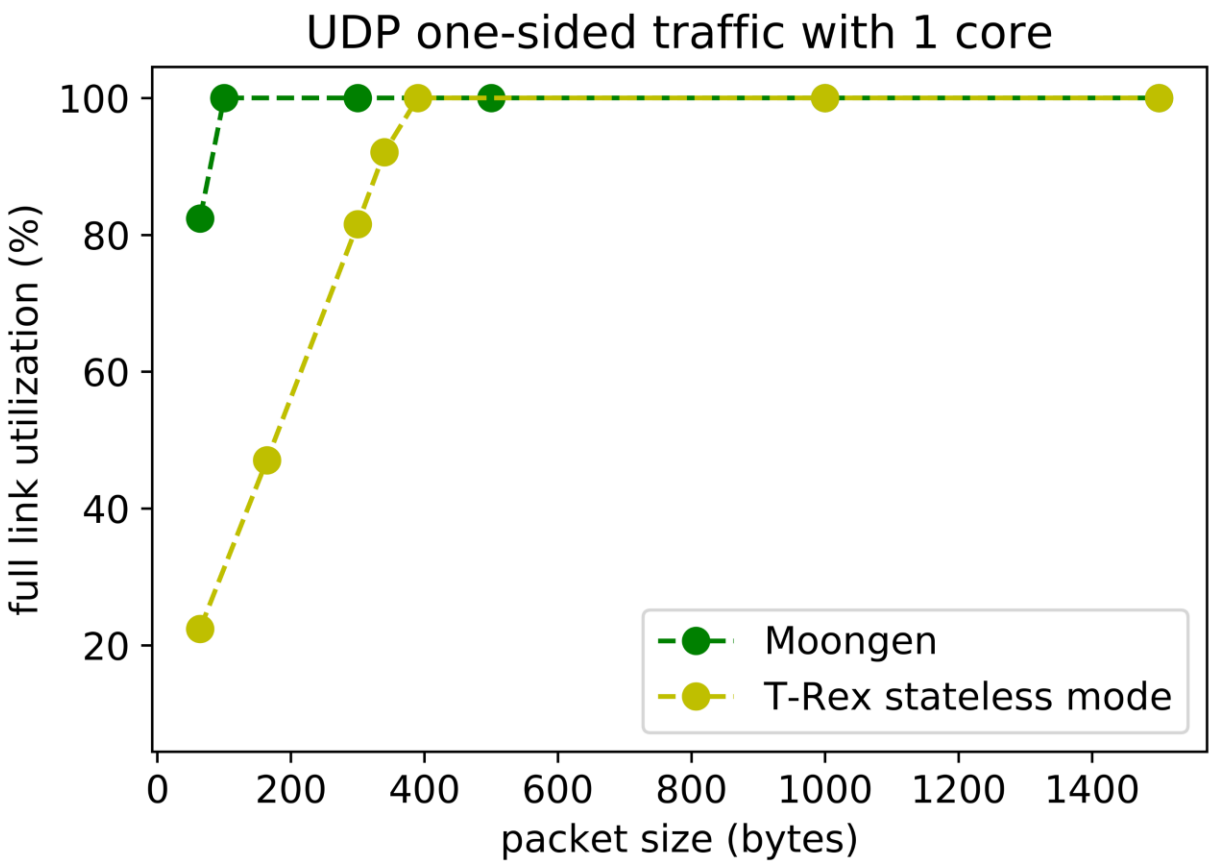
# Results – Comparison stateless traffic gen.

- Moongen has the best performance in UDP stream generation
- Simpler tools with less overhead perform in general better



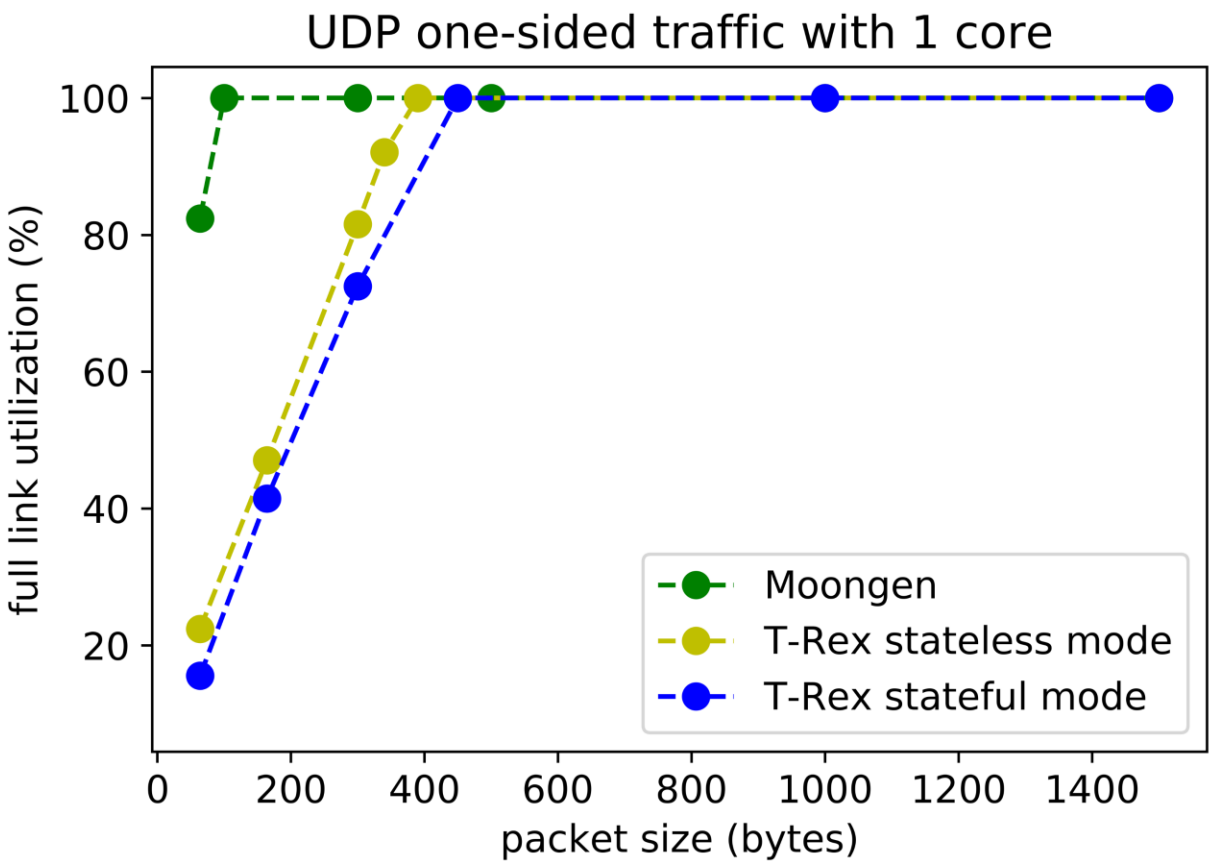
# Results – Comparison stateless traffic gen.

- Moongen has the best performance in UDP stream generation
- Simpler tools with less overhead perform in general better



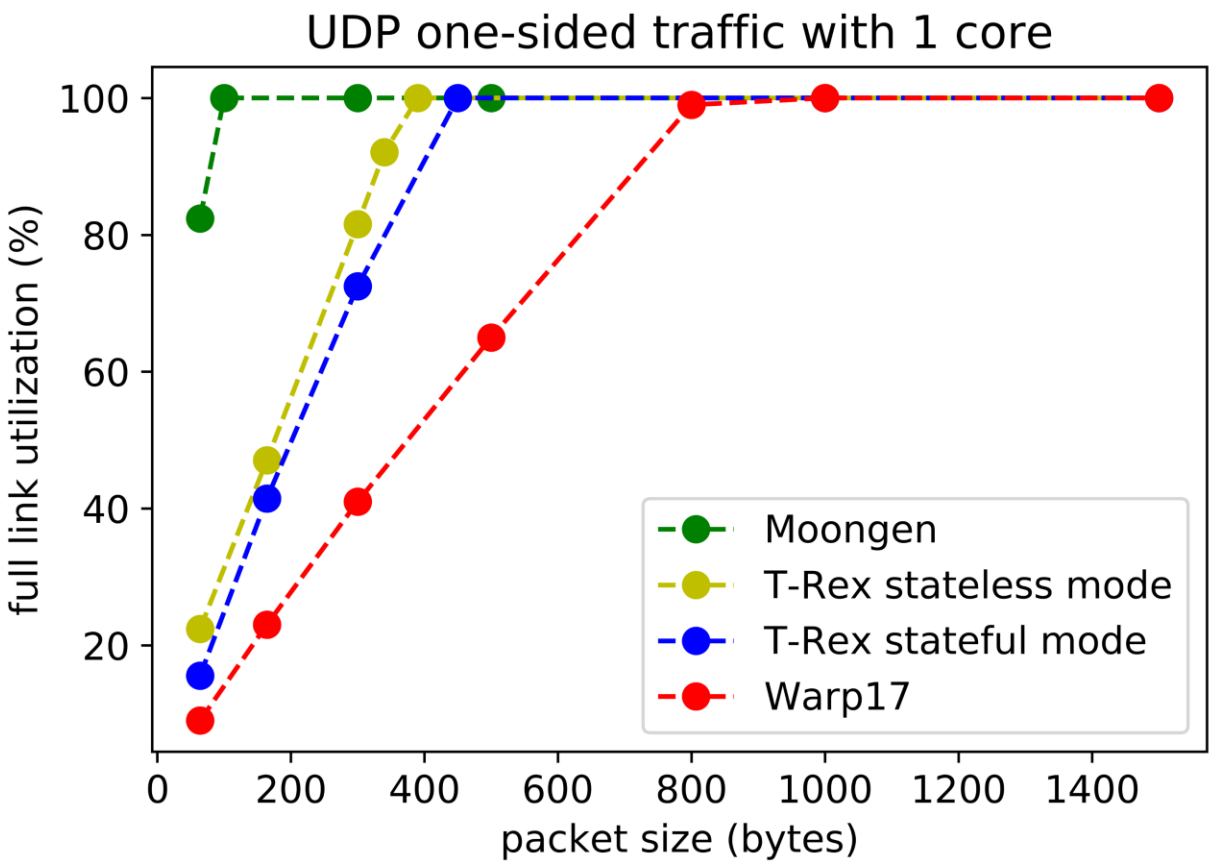
# Results – Comparison stateless traffic gen.

- Moongen has the best performance in UDP stream generation
- Simpler tools with less overhead perform in general better



# Results – Comparison stateless traffic gen.

- Moongen has the best performance in UDP stream generation
- Simpler tools with less overhead perform in general better



# Results – Overall Comparison

Moongen

- Simple tool

# Results – Overall Comparison

Moongen

- Simple tool
- Efficient in terms of CPU utilization

# Results – Overall Comparison

Moongen

- Simple tool
- Efficient in terms of CPU utilization
- Flexible

# Results – Overall Comparison

Moongen

- Simple tool
- Efficient in terms of CPU utilization
- Flexible
- No automation

# Results – Overall Comparison

| Moongen | Warp17 |
| --- | --- |
| • Simple tool<br>• Efficient in terms of CPU utilization<br>• Flexible<br>• No automation | • Full TCP stack |

# Results – Overall Comparison

| Moongen | Warp17 |
| --- | --- |
| • Simple tool<br>• Efficient in terms of CPU utilization<br>• Flexible<br>• No automation | • Full TCP stack<br>• Can be deployed as client only |

# Results – Overall Comparison

| Moongen | Warp17 |
| --- | --- |
| • Simple tool<br>• Efficient in terms of CPU utilization<br>• Flexible<br>• No automation | • Full TCP stack<br>• Can be deployed as client only<br>• Documentation sometimes poorly explained |

# Results – Overall Comparison

| Moongen | Warp17 |
| --- | --- |
| • Simple tool<br>• Efficient in terms of CPU utilization<br>• Flexible<br>• No automation | • Full TCP stack<br>• Can be deployed as client only<br>• Documentation sometimes poorly explained<br>• Internet-Mix traffic complicated |

# Results – Overall Comparison

| Moongen | Warp17 |
| --- | --- |
| • Simple tool<br>• Efficient in terms of CPU utilization<br>• Flexible<br>• No automation | • Full TCP stack<br>• Can be deployed as client only<br>• Documentation sometimes poorly explained<br>• Internet-Mix traffic complicated<br>• Limited traffic patterns |

# Results – Overall Comparison

| Moongen | Warp17 | T-Rex |
| --- | --- | --- |
| • Simple tool<br>• Efficient in terms of CPU utilization<br>• Flexible<br>• No automation | • Full TCP stack<br>• Can be deployed as client only<br>• Documentation sometimes poorly explained<br>• Internet-Mix traffic complicated<br>• Limited traffic patterns | • Can replicate any existing traffic pattern |

# Results – Overall Comparison

| Moongen | Warp17 | T-Rex |
| --- | --- | --- |
| • Simple tool<br>• Efficient in terms of CPU utilization<br>• Flexible<br>• No automation | • Full TCP stack<br>• Can be deployed as client only<br>• Documentation sometimes poorly explained<br>• Internet-Mix traffic complicated<br>• Limited traffic patterns | • Can replicate any existing traffic pattern<br>• Well documented |

# Results – Overall Comparison

| Moongen | Warp17 | T-Rex |
| --- | --- | --- |
| • Simple tool<br>• Efficient in terms of CPU utilization<br>• Flexible<br>• No automation | • Full TCP stack<br>• Can be deployed as client only<br>• Documentation sometimes poorly explained<br>• Internet-Mix traffic complicated<br>• Limited traffic patterns | • Can replicate any existing traffic pattern<br>• Well documented<br>• Not so easy to use |

# Results – Overall Comparison

## Moongen

- Simple tool
- Efficient in terms of CPU utilization
- Flexible
- No automation

## Warp17

- Full TCP stack
- Can be deployed as client only
- Documentation sometimes poorly explained
- Internet-Mix traffic complicated
- Limited traffic patterns

## T-Rex

- Can replicate any existing traffic pattern
- Well documented
- Not so easy to use
- Many options, can be overwhelming

# Results – Overall Comparison

## Moongen

- Simple tool
- Efficient in terms of CPU utilization
- Flexible
- No automation

## Warp17

- Full TCP stack
- Can be deployed as client only
- Documentation sometimes poorly explained
- Internet-Mix traffic complicated
- Limited traffic patterns

## T-Rex

- Can replicate any existing traffic pattern
- Well documented
- Not so easy to use
- Many options, can be overwhelming

# Outlook

- Move the testing on server with 100Gb Interface. DPDK should scale linearly with number of cores, should be feasible to saturate link
- Test ASTF mode of T-Rex. This would allow for a (fair) comparison with Warp17
- Compare latency measurement feature over DPDK

# THE END

Thank you for your attention!

Any Questions?