Improving the accuracy of non-cooperative active measurement

Rocky K. C. Chang
(with Ricky Mok, Weichao Li, and Star Poon)
Department of Computing
The Hong Kong Polytechnic University

Our recent works

- "On the Accuracy of Smartphone-based Mobile Network Measurement," in *Proc. IEEE INFOCOM*, Apr. 2015.
- "Improving the Packet Send-time Accuracy in embedded devices," in *Proc. PAM*, Mar. 2015.
- "Appraising the Delay Accuracy in Browser-based Network Measurement," in *Proc. ACM/USENIX IMC*, Oct. 2013.

Networked Devices

- Embedded network devices are everywhere.
- Researchers use them to measure the Internet.

Home Router

BISmark

Travel Router

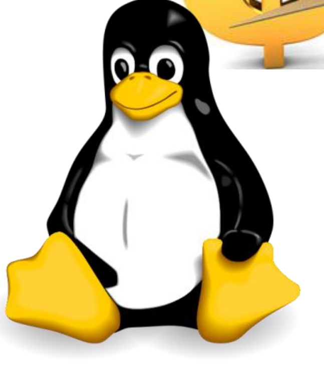
RIPE

Raspberry Pi

CAIDA

Advantages

- Green
  - Operated in low power
- Ease to deploy
  - Small and portable
- Low cost
  - From USD 25
- Linux-based
  - Run the same software in PCs

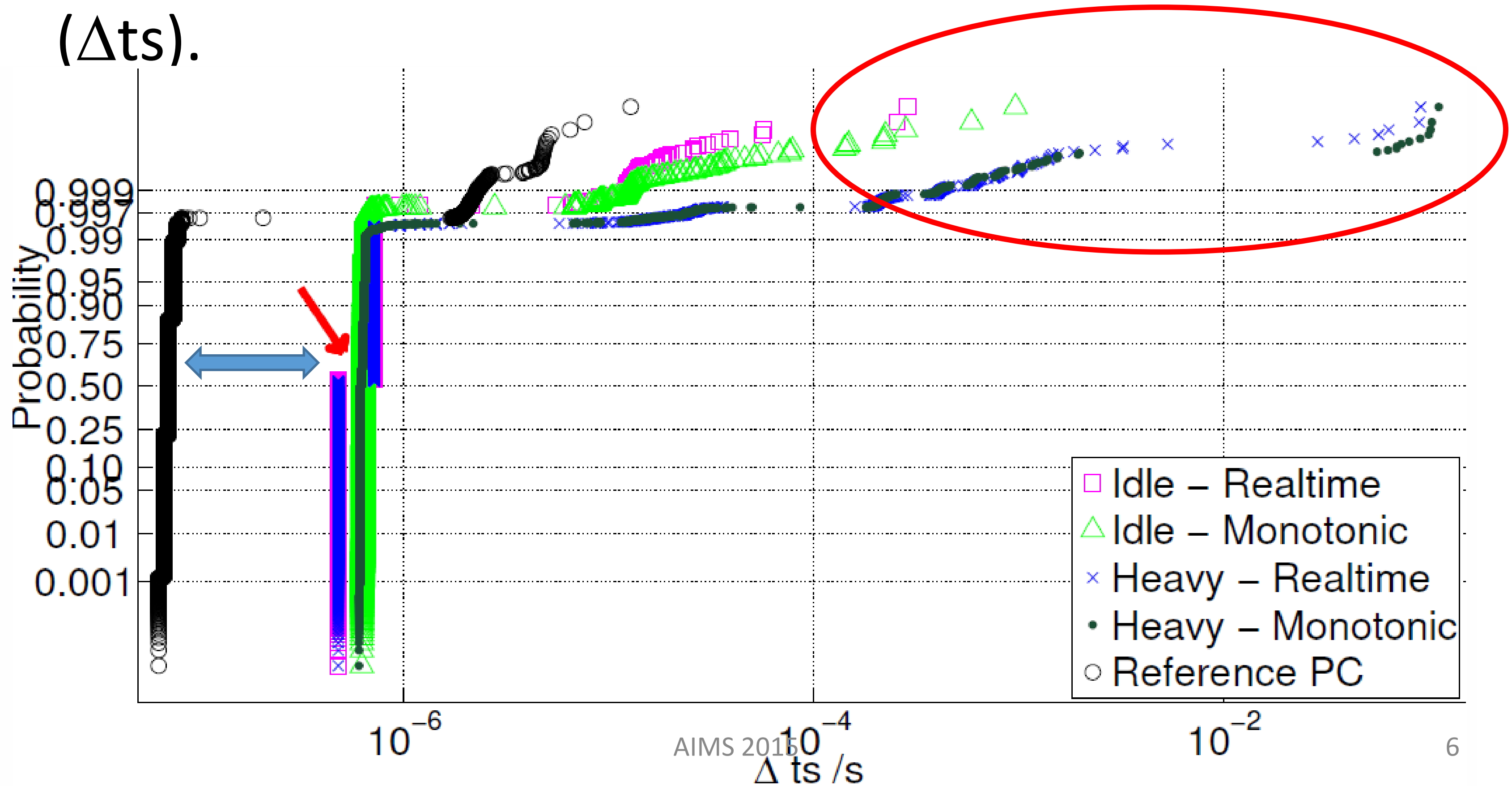


Three main problems

- Timestamp retrieval
  - Low timestamp resolution
- Sleep accuracy
  - Oversleep
- Packet sending performance
  - Large inter-departure time between packets
- Further aggravation by other computation overheads (e.g., processing other traffic)

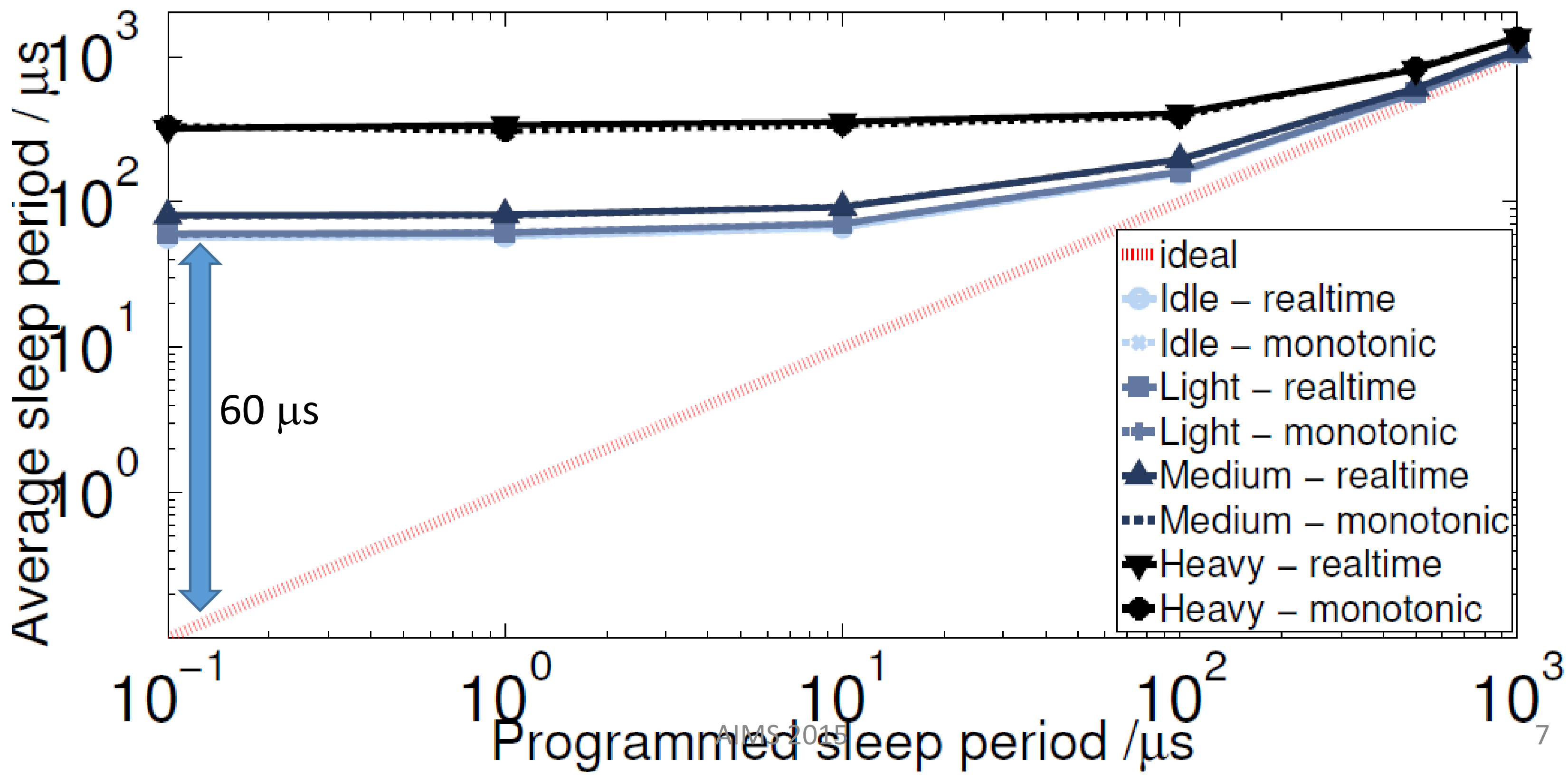
Timestamp Retrieval

- Use `clock_gettime()` to get nanosec resolution.
- Compute the difference of consecutive timestamps (Δts).



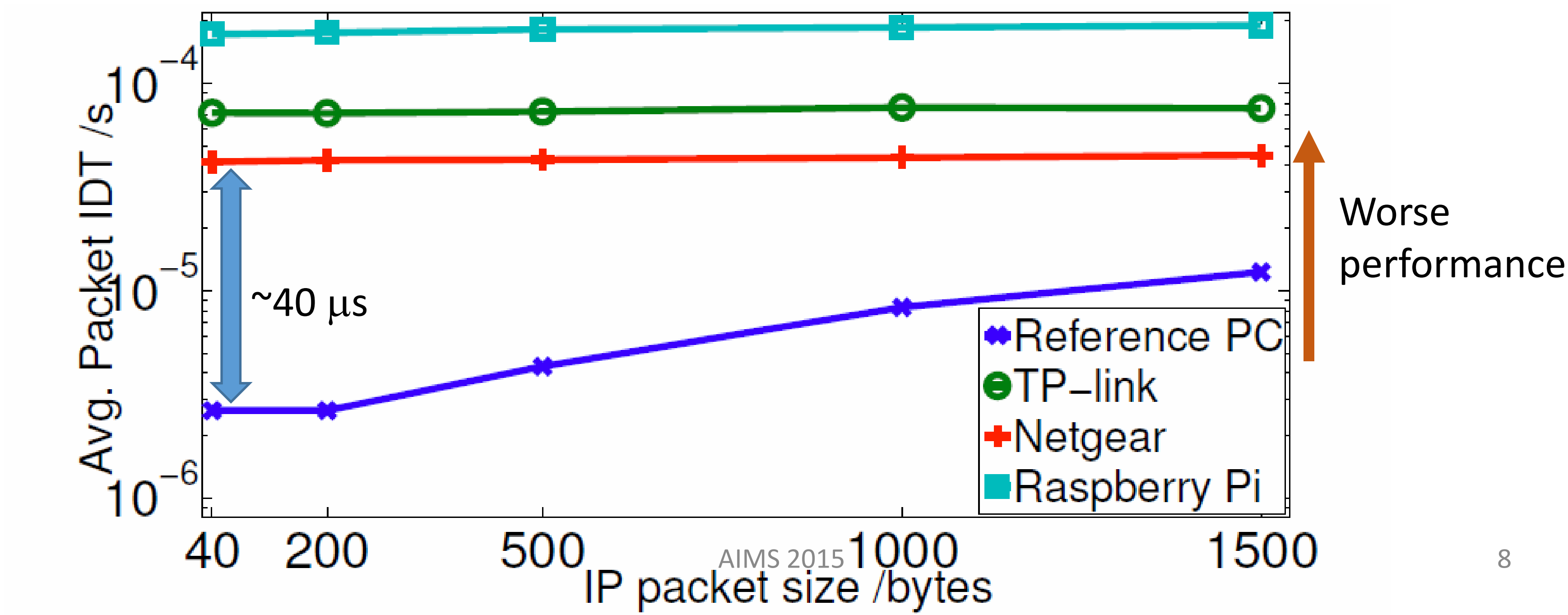
Sleep accuracy

- Use `clock_nanosleep()` for the evaluation.
- The sleep function in user-space is not accurate.



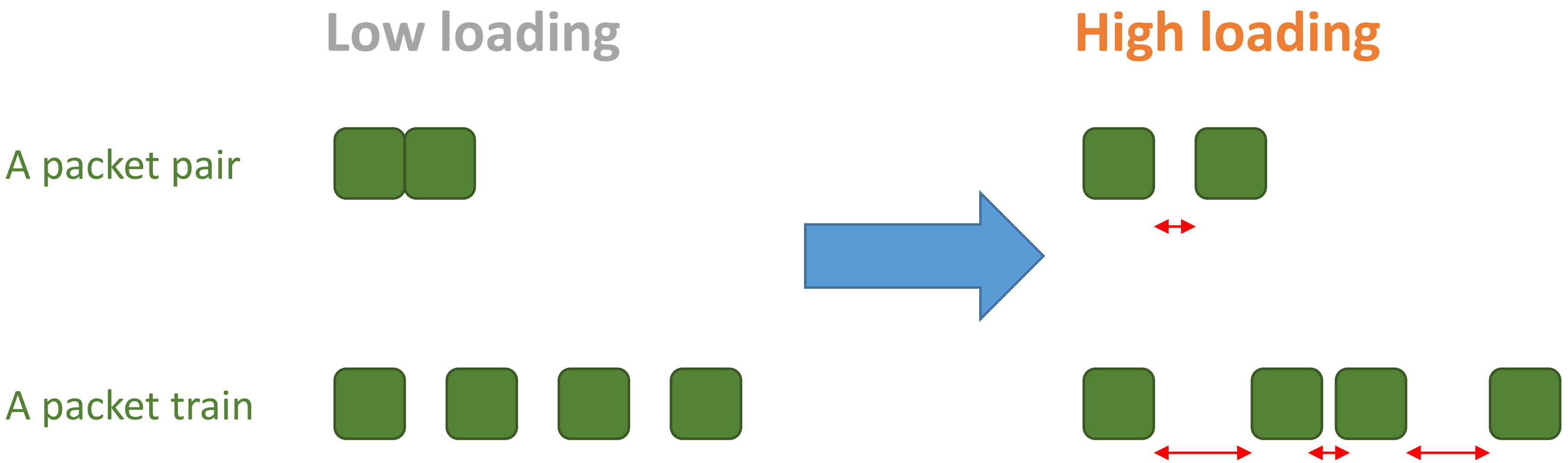
Packet sending performance

- The minimum packet inter-departure time (IDT) is much higher for embedded devices.
- Flush out 100,000 identical TCP packets using raw socket (i.e., `sendto()`).



How to improve the packet send-time accuracy?

- We define as the difference between the scheduled probe packet pattern and the true pattern.
- Wrong patterns can seriously affect the accuracy of network measurement tools.

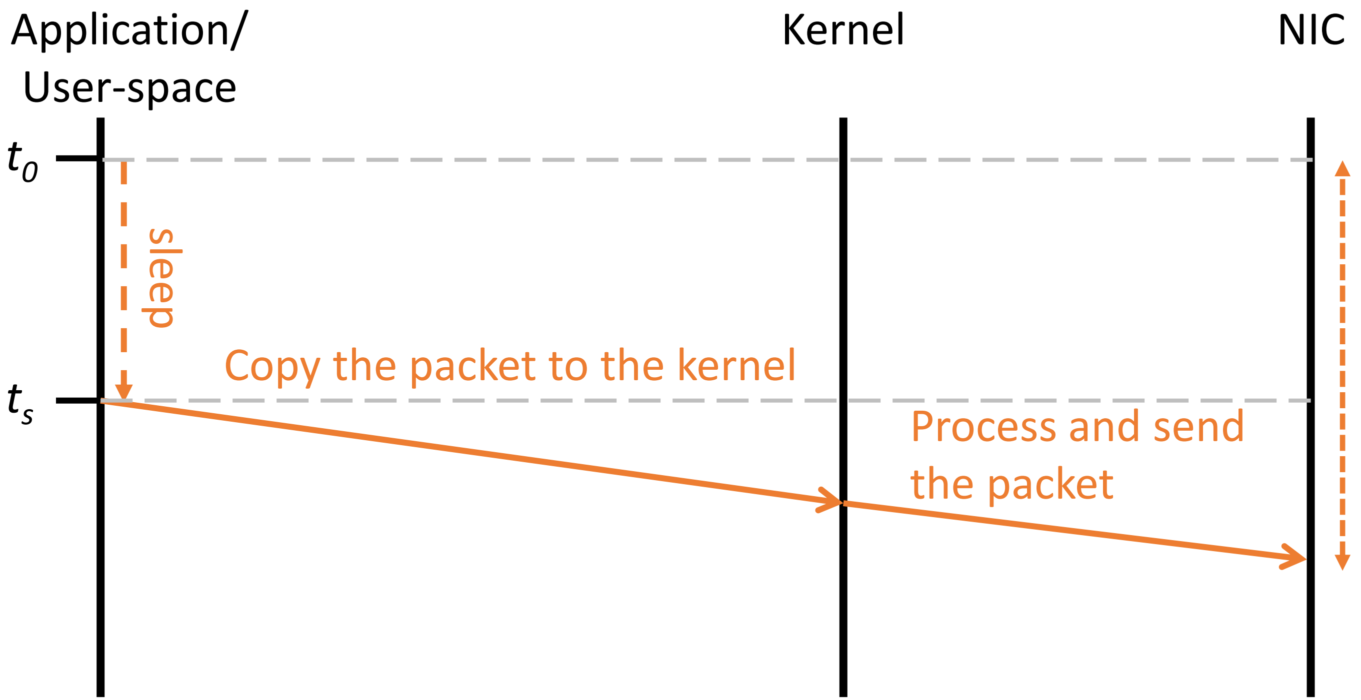


Our solution: Pre-dispatch model

- Probe packets can often be prepared **before** the actual sending time.
- In pre-dispatch model, the packets can be buffered in the kernel and wait for the actual sending time.
  - Reduce the critical path of sending packets
  - The timestamp retrieval and sleep are much less affected by other loading.
- Our implementation: OMware

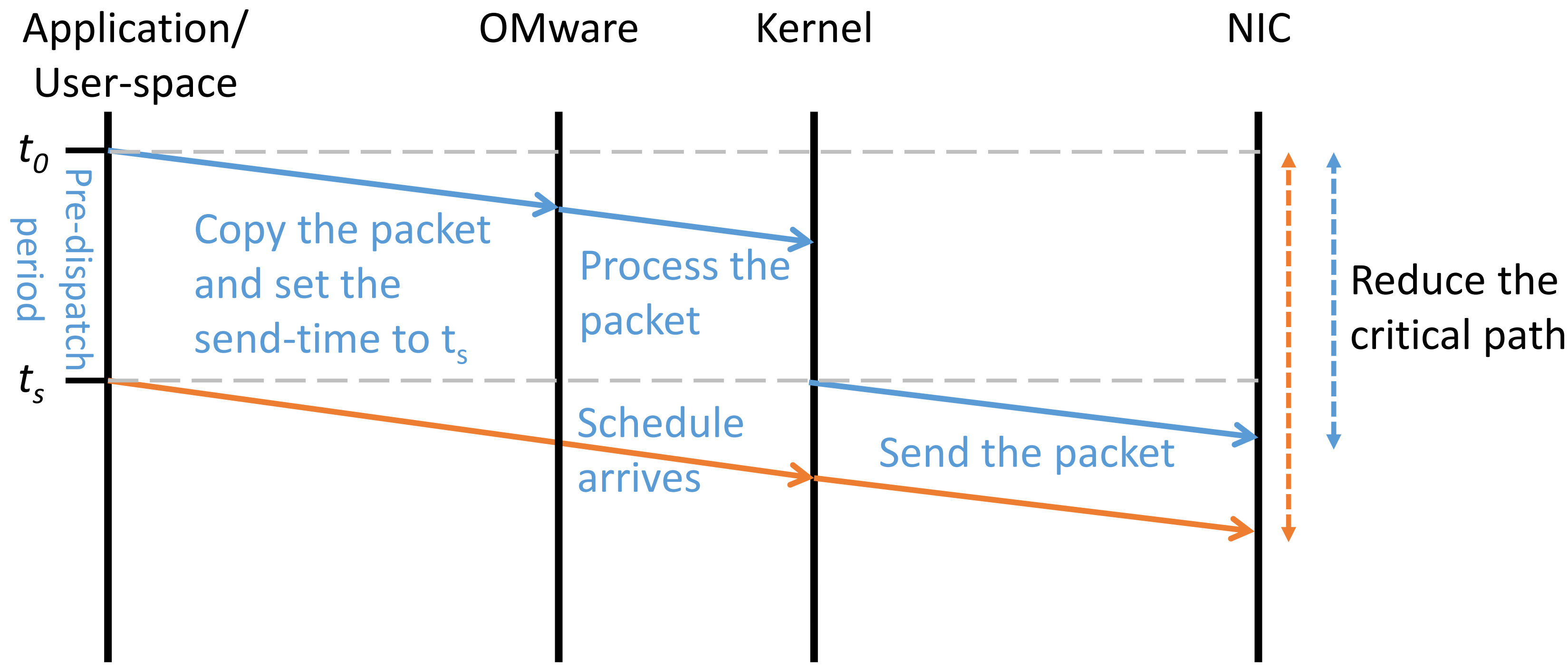
The pre-dispatch model

- Sequential model vs. pre-dispatch model



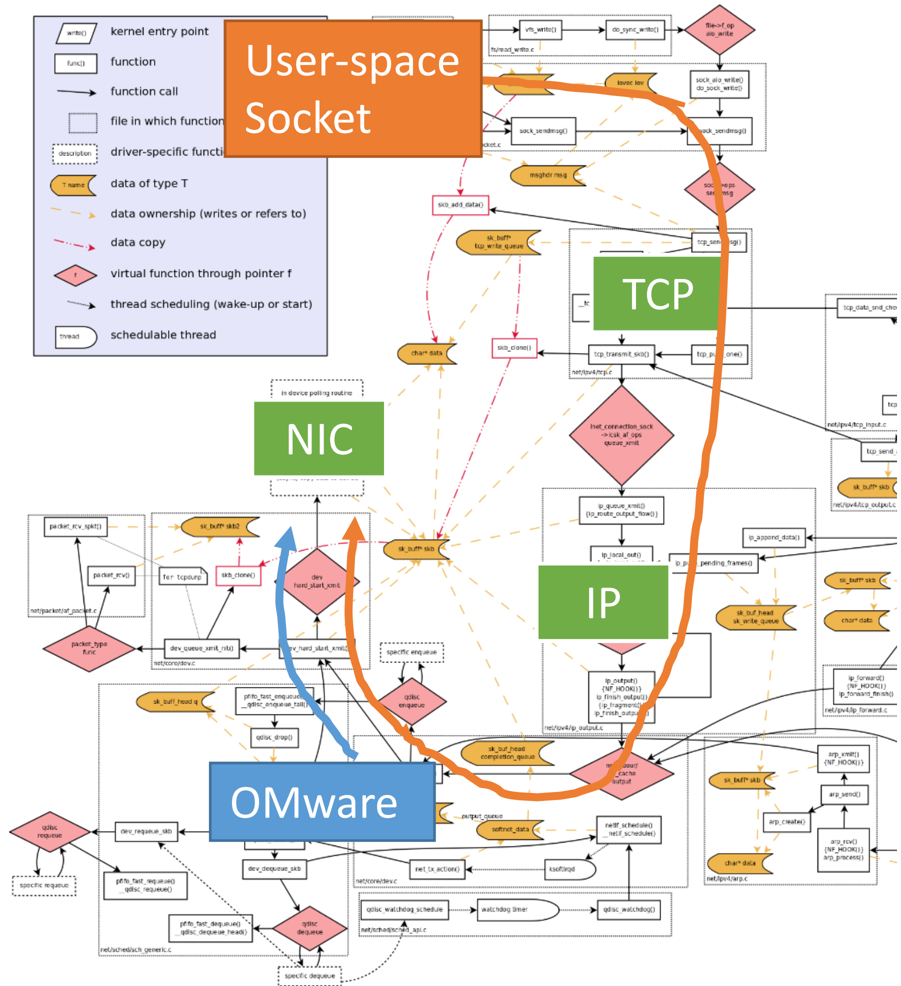
The pre-dispatch model

- Sequential model vs. pre-dispatch model



Packet flow in Linux

- Long path for packet traverse from user-space to the network interface.



Source: http://www.linuxfoundation.org/images/1/1c/Network_data_flow_through_kernel.png

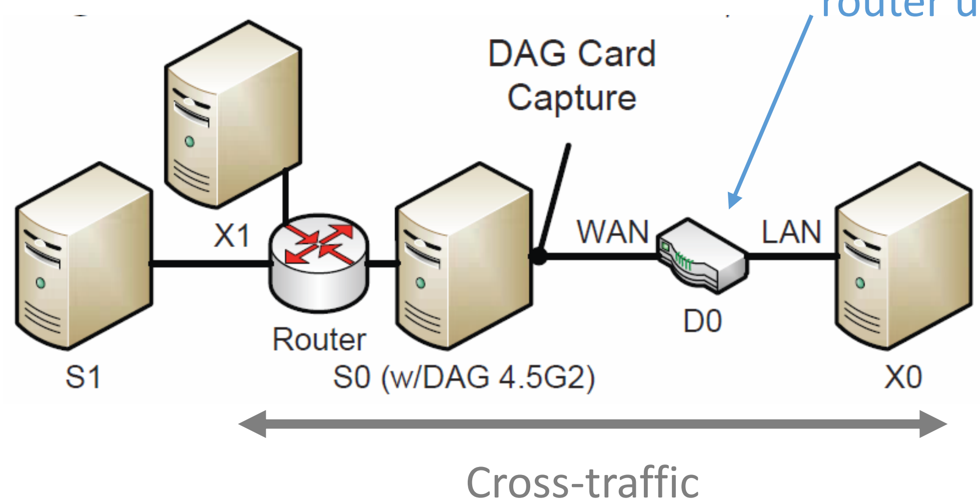
OMware

- Loadable kernel module
- Buffer the pre-dispatch packets
- Employ high resolution timer (HR_TIMER) to trigger the packet sending schedule
- Provide interface to communicate with user-space applications using netlink
- Optimized call for sending packet pairs

Evaluation with Netgear and TP-link routers

- Two home routers are used.
  - NETGEAR WNDR-3800
  - TP-LINK WR1043ND
- Endace DAG is used to capture the packets sent by the router.

router under test

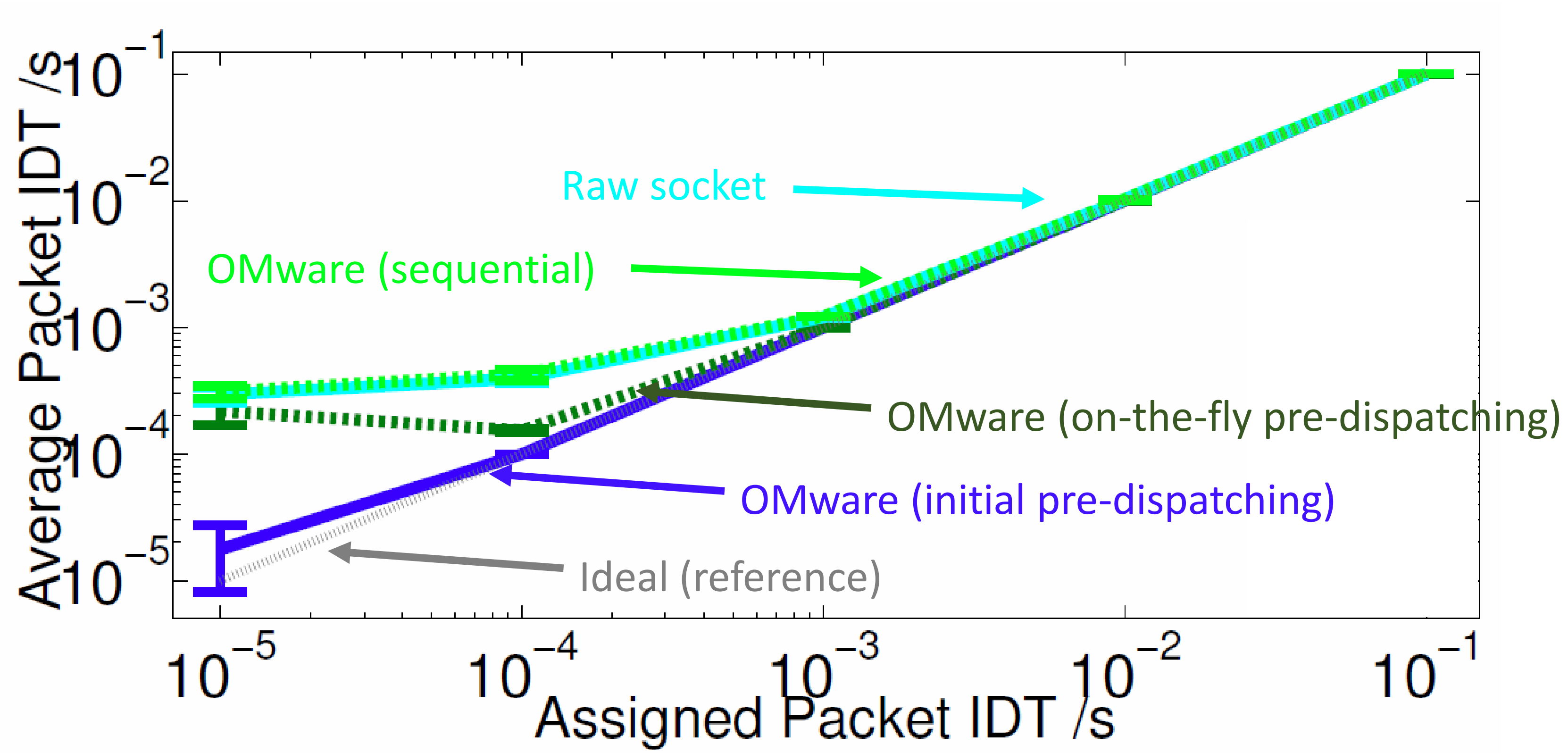


Evaluation settings

- Sending packet trains/packet pairs under different levels of cross-traffic
  - OMware (initial pre-dispatching)
  - OMware (on-to-fly pre-dispatching)
  - Raw socket without OMware
- Evaluate
  - Packet train's send-time accuracy
  - Pre-dispatching period
  - Packet-pair accuracy
  - Packet send timestamp accuracy

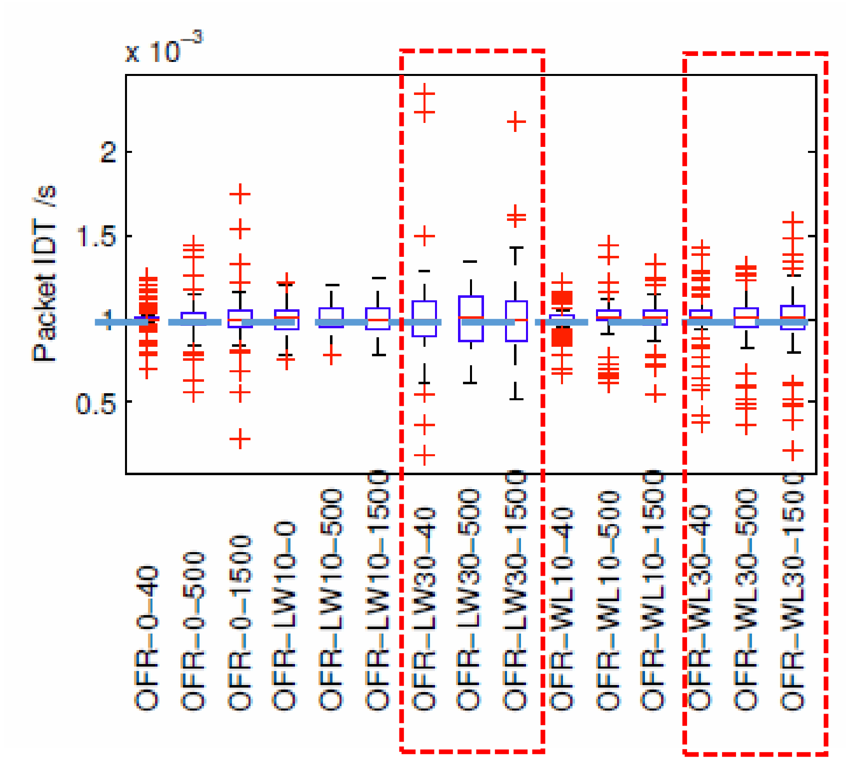
(1) Packet train's IDT at idle

- Pre-dispatch model can send packet train with smaller IDT.

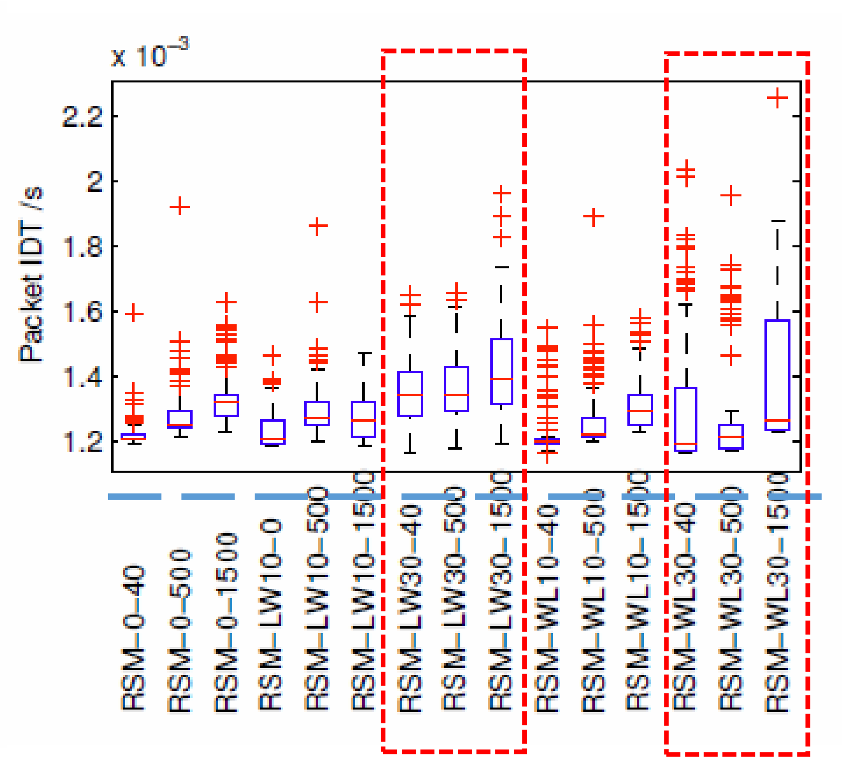


(1) Packet train's IDT accuracy with cross traffic

- OMware (with pre-dispatching) performs well under heavy cross-traffic.



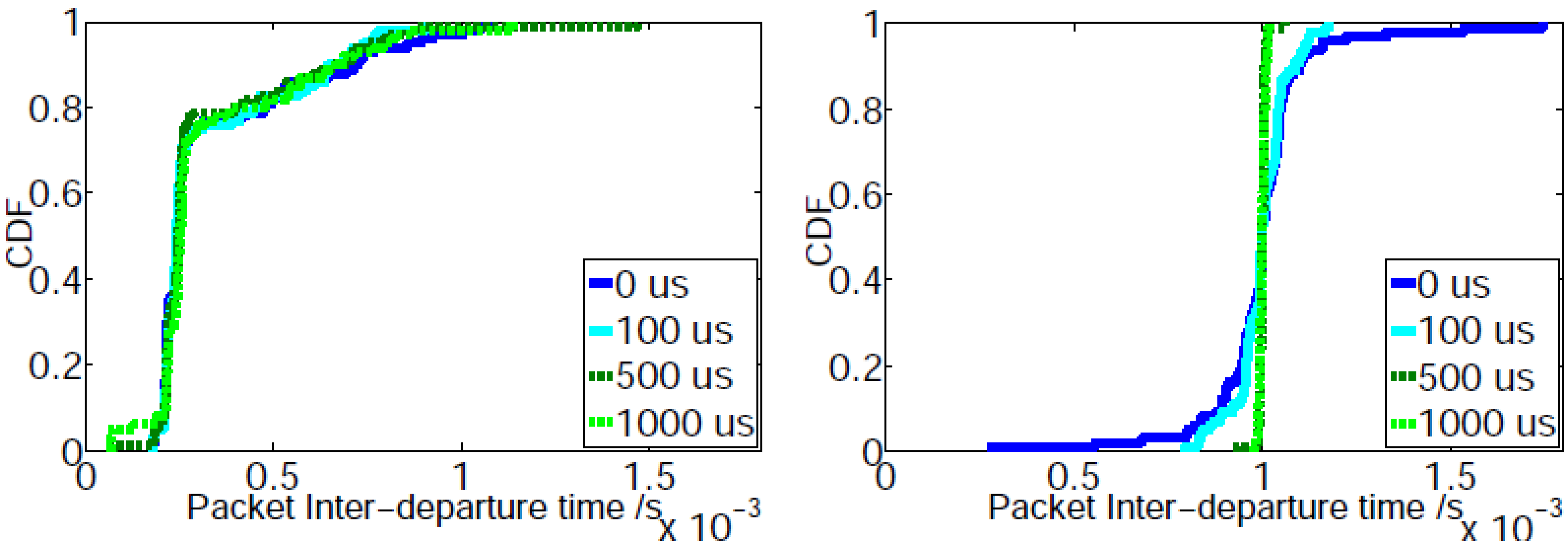
OMware (on-the-fly pre-dispatching)



Raw socket

(2) Determining the pre-dispatch period

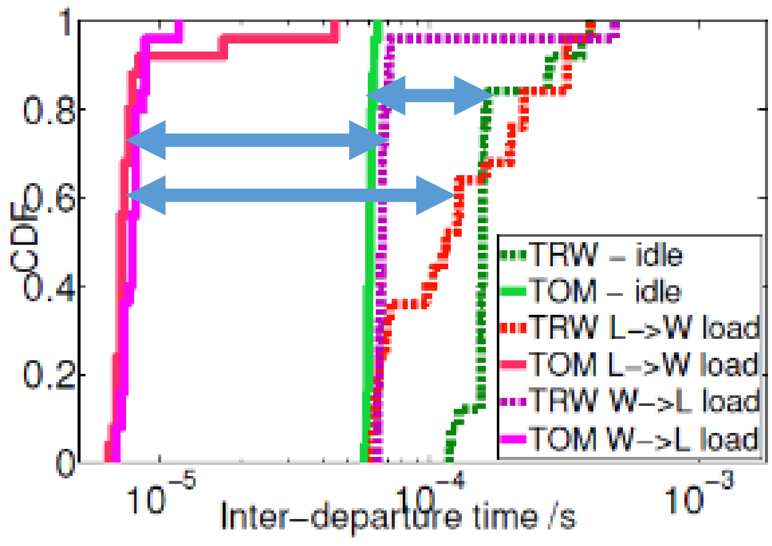
- How long should the pre-dispatching period be?
- Two IDT: 10μs and 1000μs
- Four pre-dispatch period: 0/100/500/1000μs



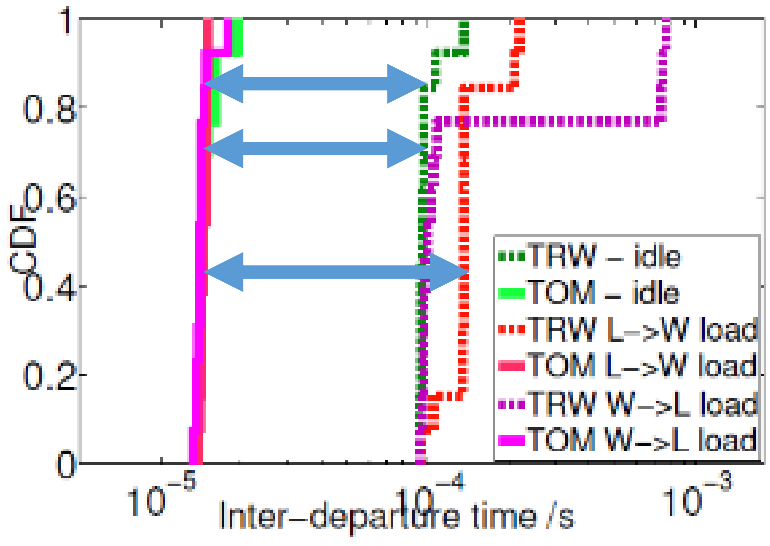
(a) $\alpha = 10\mu s.$ (b) $\alpha = 1000\mu s.$

(3) Packet-pair accuracy

- OMware can reduce the IDT of packet pairs for 2 to 10 times against raw socket.
  - Increase the highest measureable capacity.
  - TRW/TOM: Raw socket/OMware



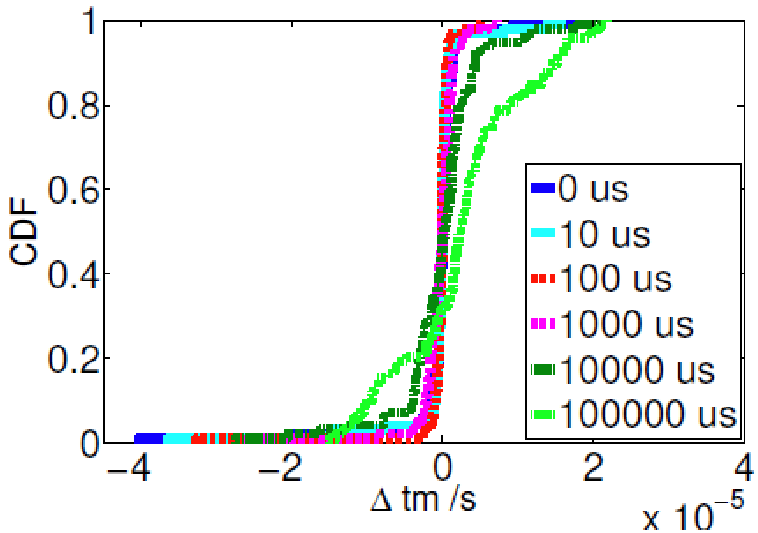
(a) NETGEAR.



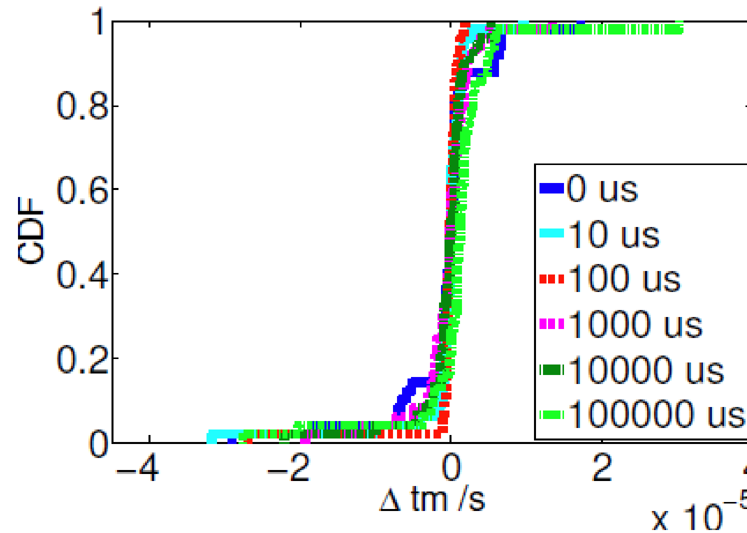
(b) TP-LINK.

(4) Timestamp accuracy

- Compute Δtm = sent time returned by OMware – the actual sent time reported by DAG card.
- OMware can provide microsecond-level accuracy.



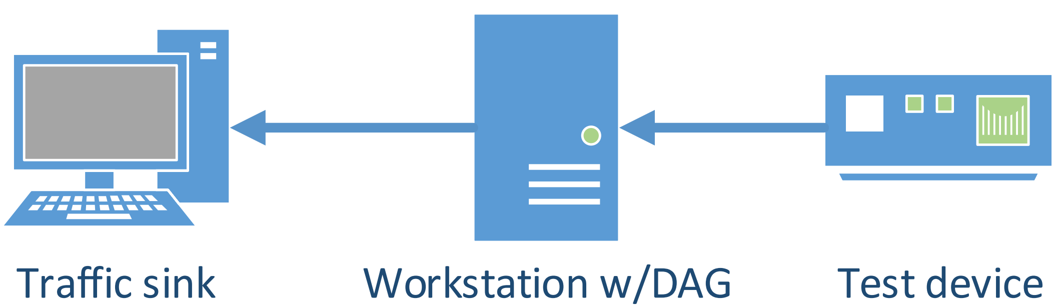
(a) NETGEAR.



(b) TP-LINK.

Evaluation with single-board computers

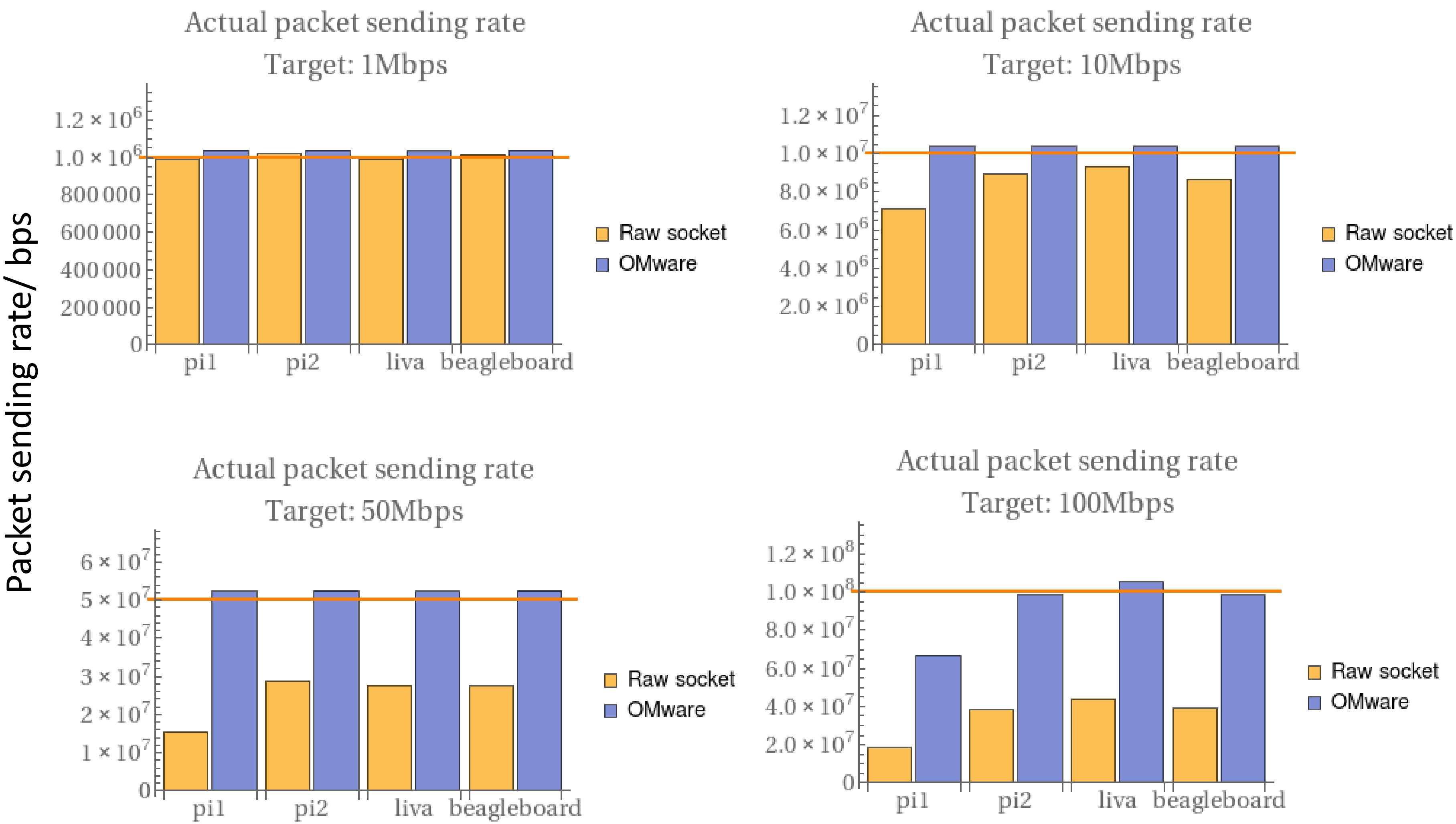
- Simple Dumbbell testbed
  - Test device sends packet trains (consisted of 50 packets) at different sending rates.
  - We used both tcpdump (run on the test device) and DAG card (installed in external workstation) to capture the packet timestamps.
- The test device generates different number of cross-traffic flows to the traffic sink using D-ITG.



Tested devices

Tested Devices	Raspberry Pi Model B	Raspberry Pi 2 Model B	ECS LIVA	Beagleboard black
Kernel version	3.18.0-trunk-rpi	3.18.0-trunk-rpi2	3.13.0-39-generic	3.17.4-301.fc21.armv7hl
Network Interface	100Mbps	100Mbps	1Gbps	100Mbps
Ethernet Controller	LAN9512 - USB to Ethernet	LAN9514 - USB to Ethernet	RTL8111/8168/8411 PCI-E Gigabit Ethernet Controller	Fast Ethernet (MII based)
Distribution	Raspbian 2015-02-16	Raspbian 2015-02-16	Ubuntu 14.04.1 LTS	Fedora 21 for ARM

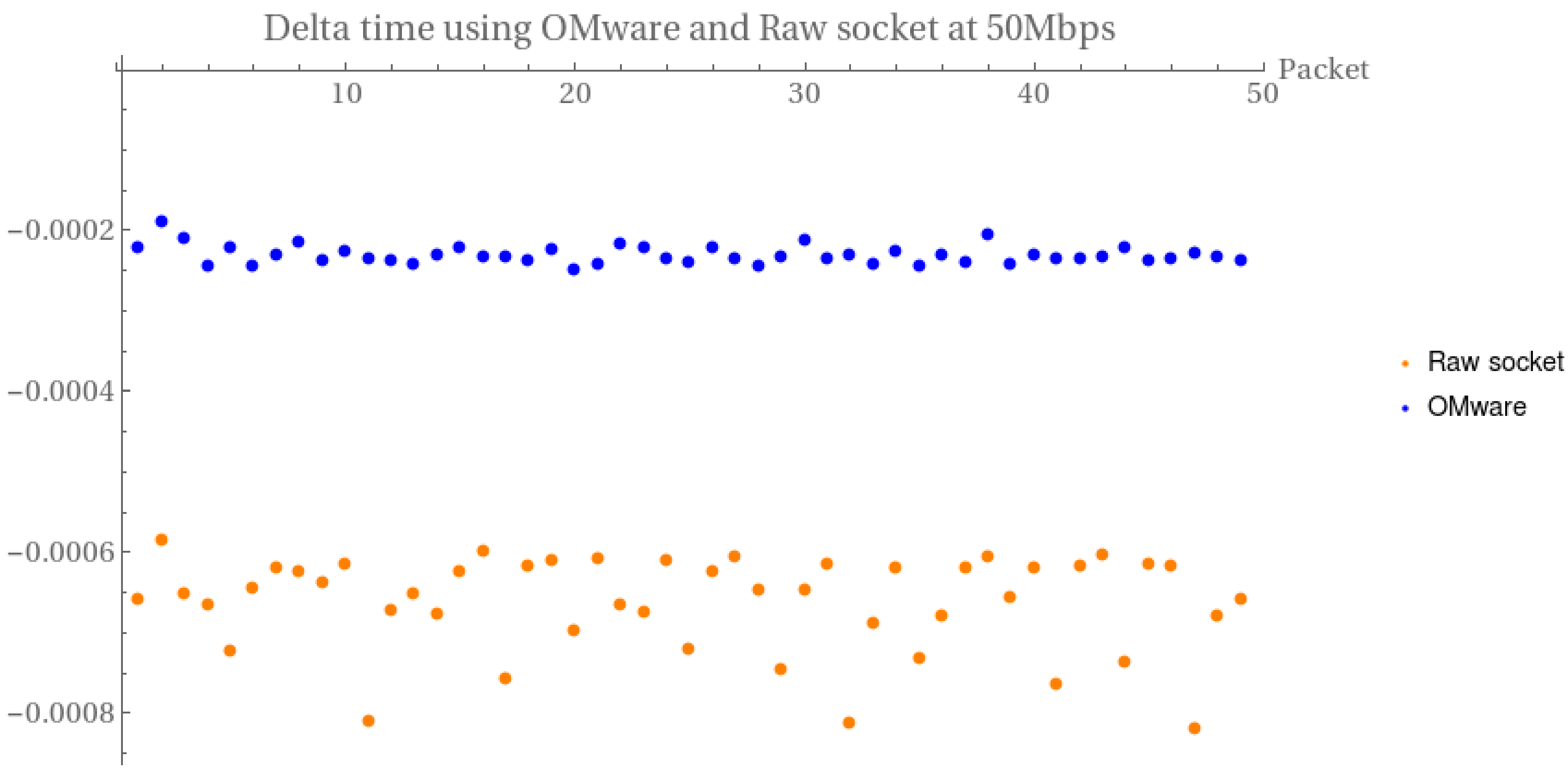
(1) Actual packet sending rate using OMware and raw socket



(1) Actual packet sending rate using OMware and raw socket

- By using OMware, we can accurately send the packet trains at the pre-defined sending rate.
- The delay from user space to kernel space is significant at high sending rate on embedded devices.

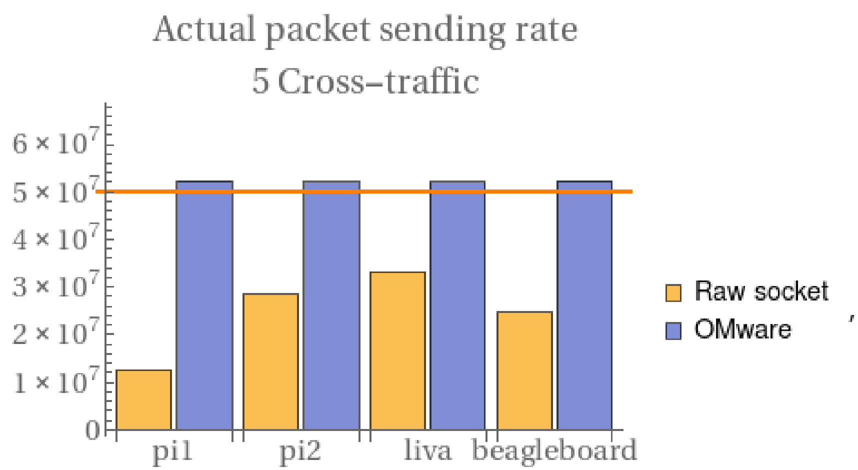
(2) Inter-packet delay using OMware and raw socket

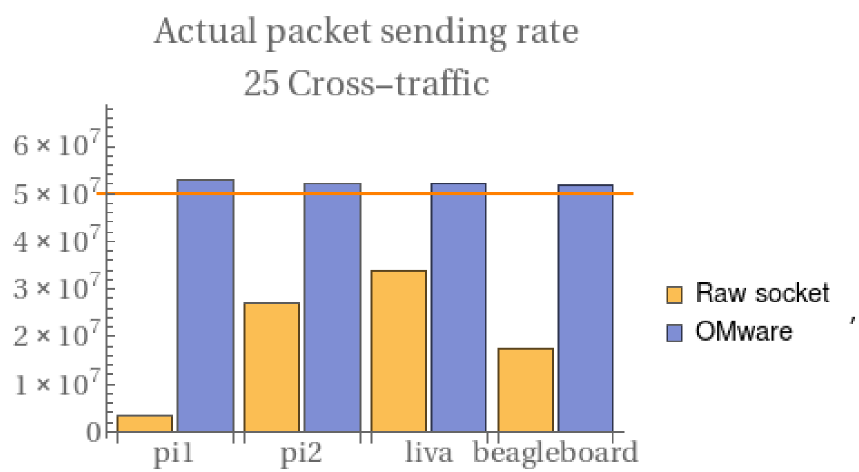


- Raspberry Pi Model B at 50Mbps, timestamp from Dag card
- Theoretical inter-packet delay: 1/(50Mbps/(1514Byte*8)) = 0.0002422

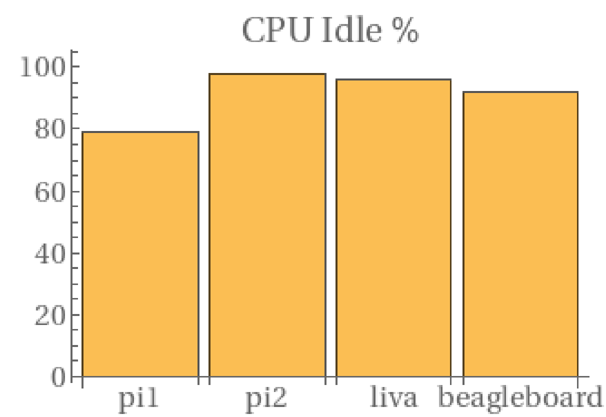
(3) Interference from CPU loading

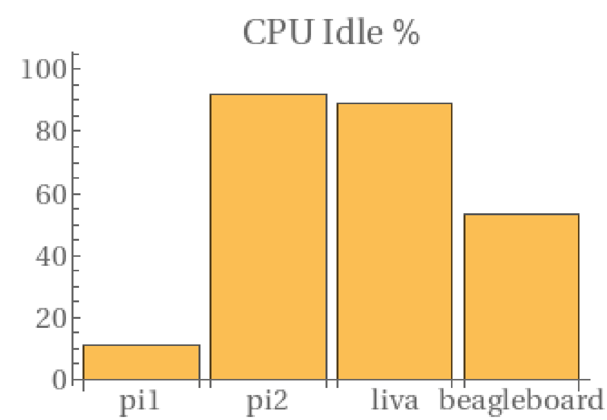
Packet sending rate/ bps





Percentage /%



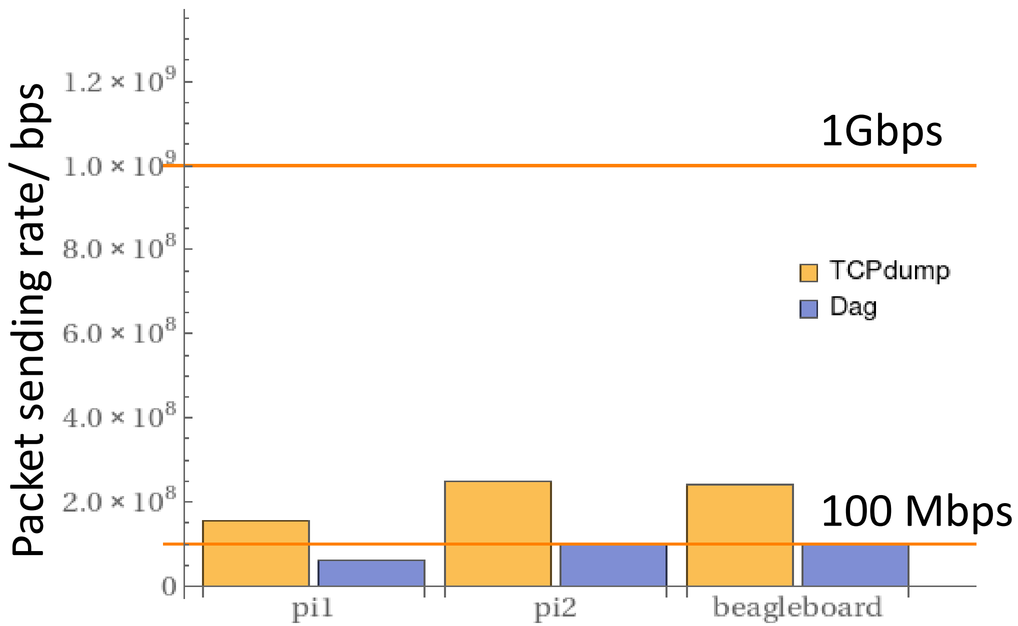


(3) Interference from CPU loading

- OMware, which is implemented as a kernel module, has a high execution priority in the system. It can mitigate the interference from user-space processes which
  - consume the CPU resources.
  - consume the NIC resources.
- The packet sending schedule in a busy system has almost no impact when OMware is used.

(4) tcpdump's timestamp problem

- Send a packet train at the maximum speed.
- The packet sending rates computed by using tcpdump can be higher than the NIC speed.



(4) tcpdump's timestamp problem

- The packet sending timestamps captured by tcpdump do not match with the DAG ones which the software requires to send packets close to/higher than the line rate.
- tcpdump reports a timestamp **before** the packets are actually sent onto the wire.
  - tcpdump timestamps cannot reflect the queuing delay induced by the driver queue.

Conclusions

- OMware can be used to send scheduled probe packets accurately.
- High CPU loading does not affect the accuracy of packet sending for OMware-enabled devices.
- Timestamp from tcpdump may deviate from the actual sending time on the wire at a high sending rate.

Thanks