TOSHIBA
Leading Innovation

Evaluation of TIPC

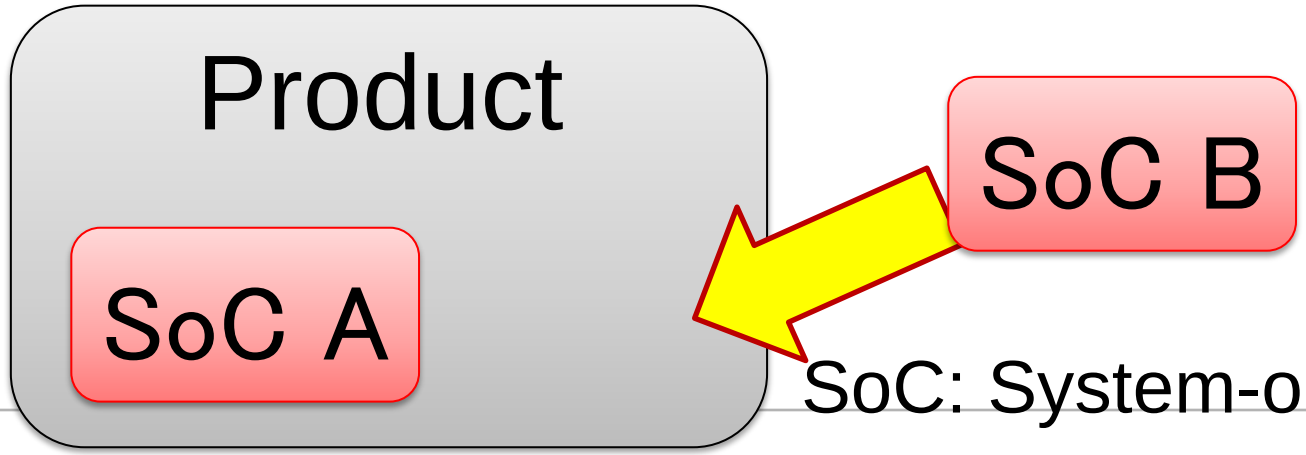
Mitsuhiro Kimura
TOSHIBA CORPORATION

CELF Japan Technical Jamboree #37, May 20, 2011

Toshiba Group contributes to
the sustainable future of planet Earth.

Background

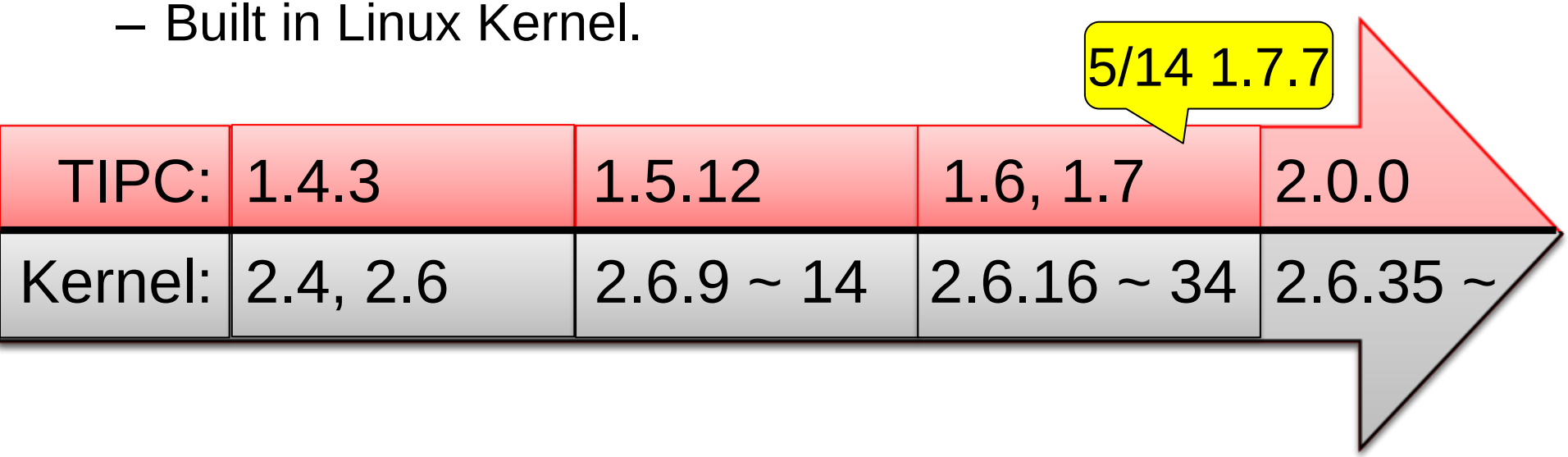
- **Integrating one more SoC in the embedded system for adding more functions.**
- **We will discuss the interconnection of two SoCs:**
  - It needs to choose a bus from available interface buses in SoCs.
  - Ethernet is more commonly available.
    - Required a quick and lightweight protocol stack on Ethernet.



SoC: System-on-a-chip

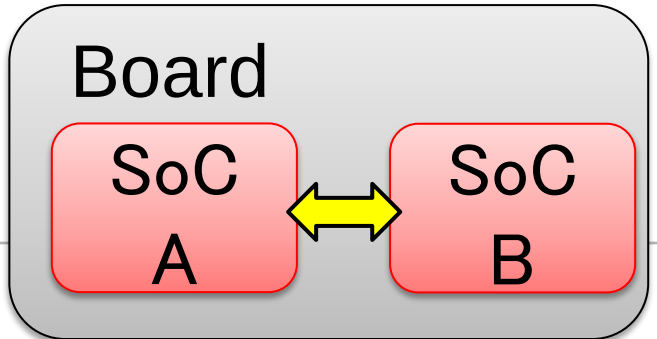
TIPC : Transparent Inter-Process Communication

- **The protocol specifically designed for cluster communication – quick and lightweight**
- **Supposing little packet loss and infrequent retransmission**
  - Socket I/F
    - sockets corresponding TCP, UDP.
  - Built in Linux Kernel.



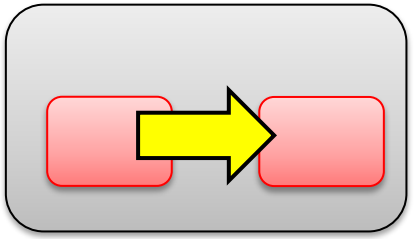
Evaluation environment

- **Interconnection of SoCs in embedded systems is similar to cluster connections in that both of them are little packet loss intra-communications.**
- **Testing SoC's used for embedded systems**
  - (reference [2], evaluation with Xeon, gigabit ethernet)
- **Benchmarking throughput as well as CPU load**
  - MIPS32®24K® 533MHz ×2, 100Mbps Ethernet
  - Linux Kernel 2.6.20.19, TIPC Version 1.7.7-rc1

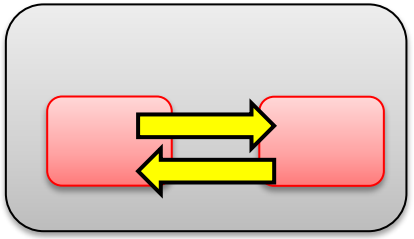


Test cases

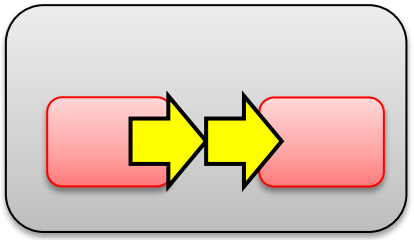
- (1) Max throughput
  - Send data in one direction.
  - Measure throughput and CPU load per message size



- (2) Latency
  - Data flows in both directions alternately in a ping-pong fashion.
  - Measure the roundtrip time per message size.



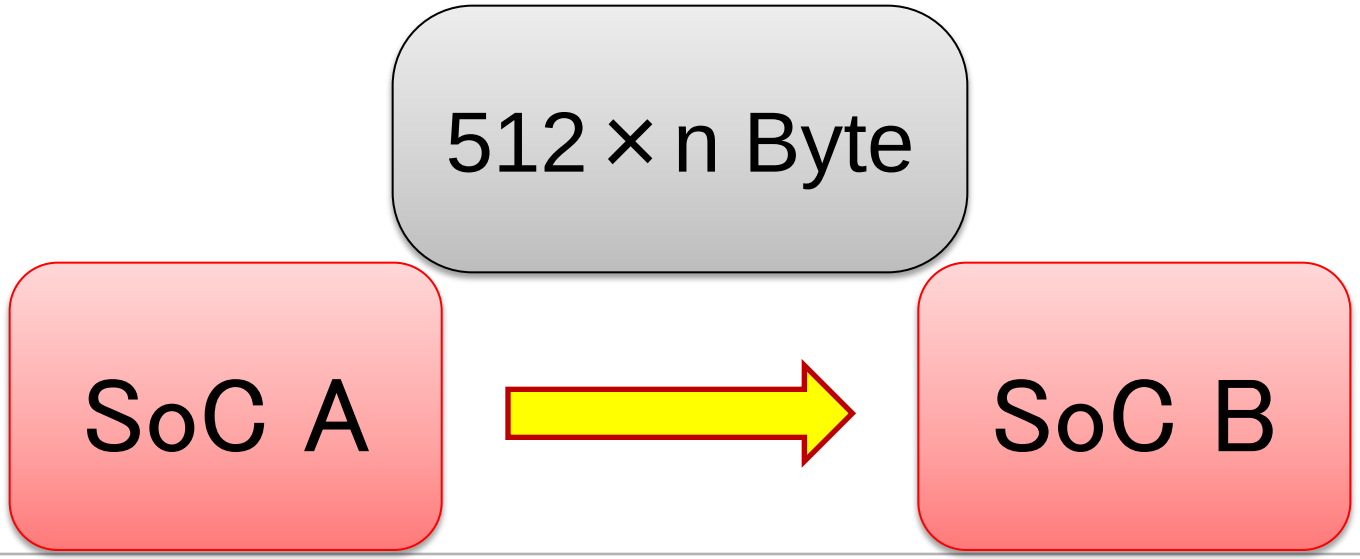
- (3) CPU load for Streaming
  - Send data in one direction at a constant rate.
  - Measure the CPU load.



(1) Max throughput, CPU load

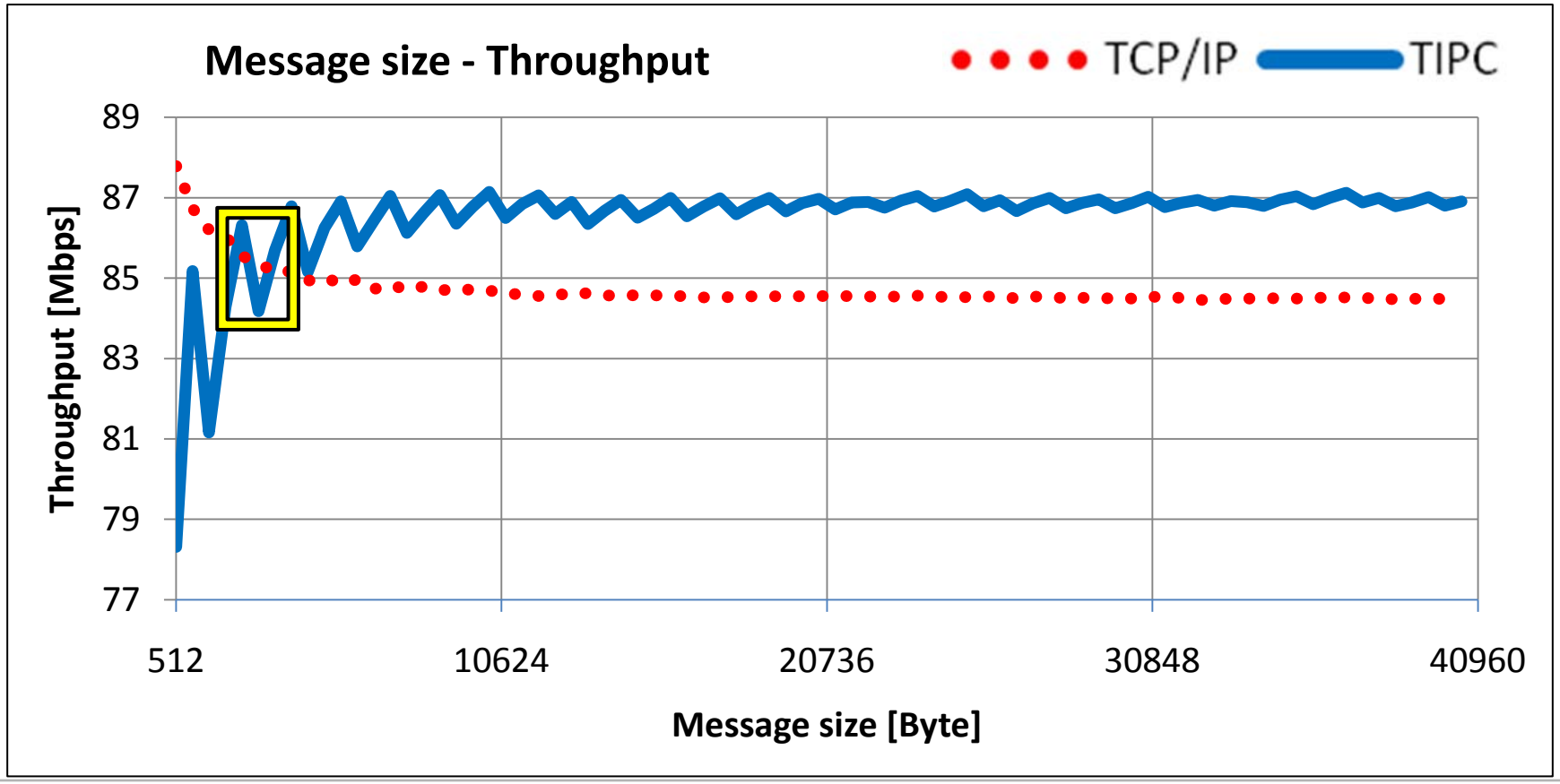
- **Message size: 512, 1024, 1536, … 40448Byte**
  - Watch a change depending on message size.
  - Compare throughput and CPU load of TIPC with TCP/IP.

- **Measure the time of send()/recv() 10,000 times.**



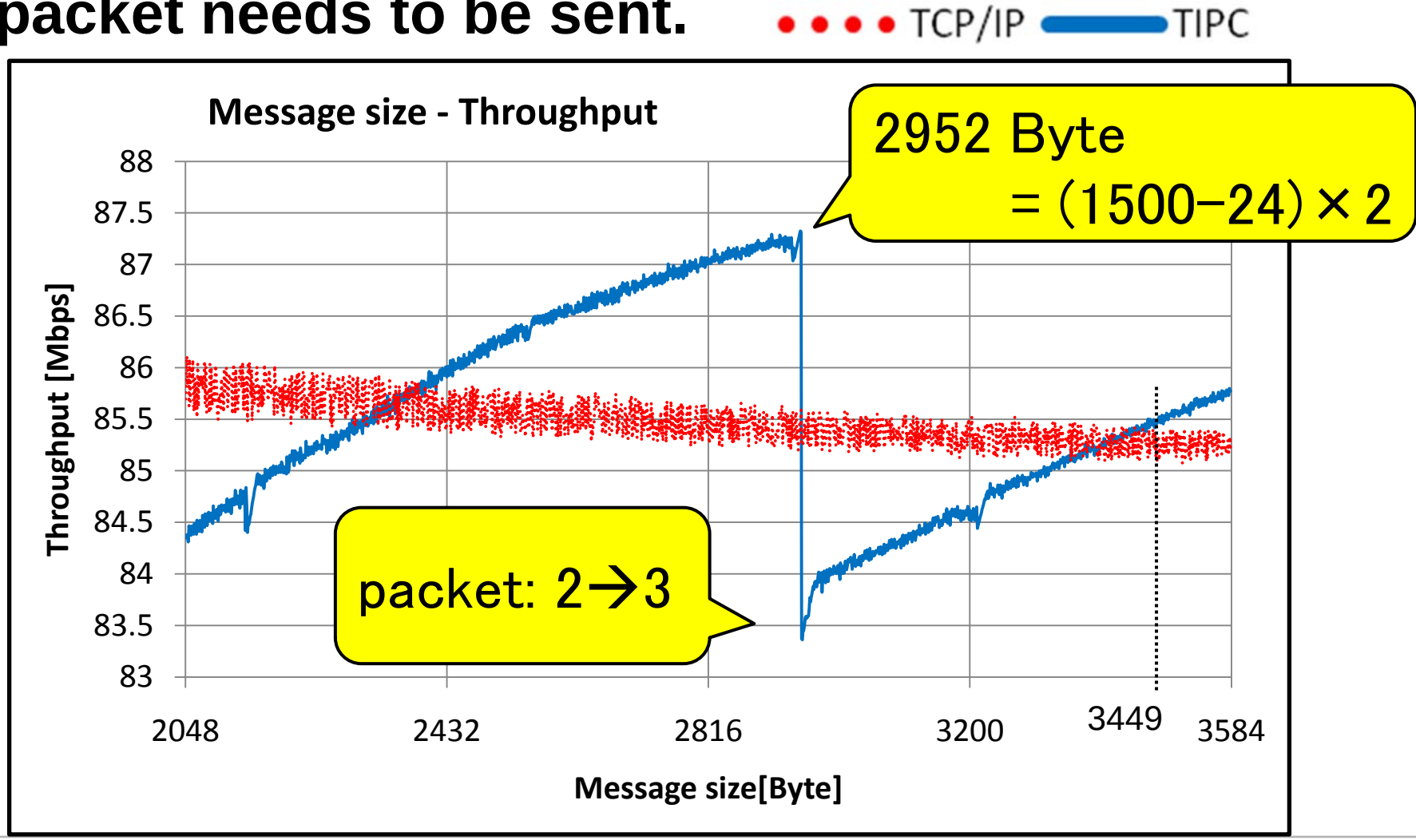
(1) Max throughput

- TCP/IP < TIPC
  - Header: TCP/IP:40Byte, TIPC:24Byte, MTU:1500Byte
  - Small header → high transfer efficiency



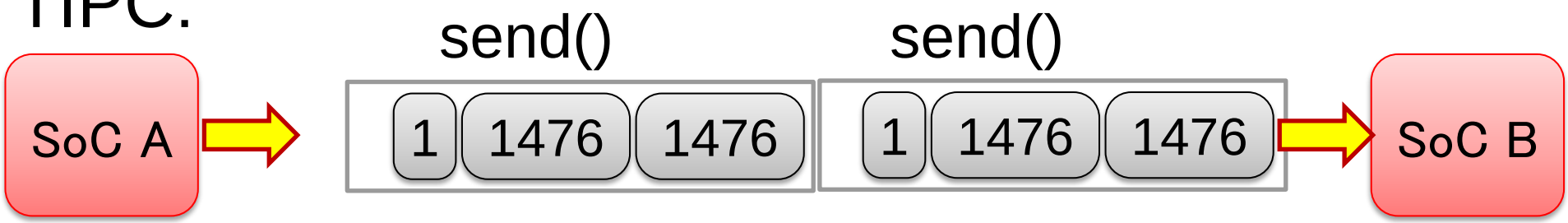
(1) Max throughput

- **TIPC throughput drops in case one more packet needs to be sent.**



Reason of TIPC throughput drop

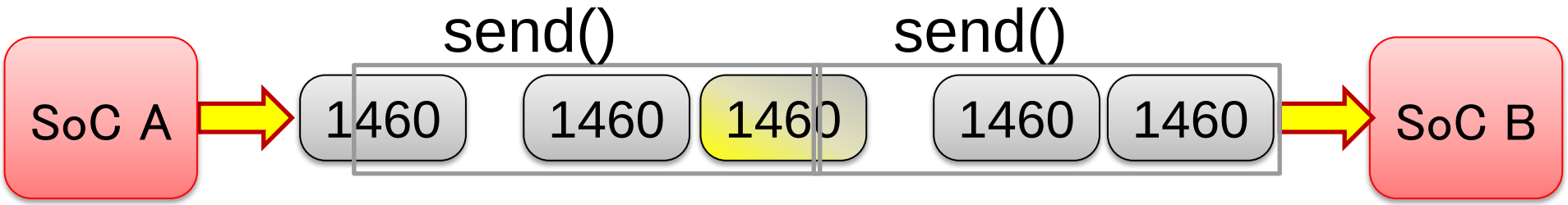
TIPC:



- 1500Byte – 24Byte = 1476Byte.

  2953Byte = 1476Byte × 2 + 1Byte

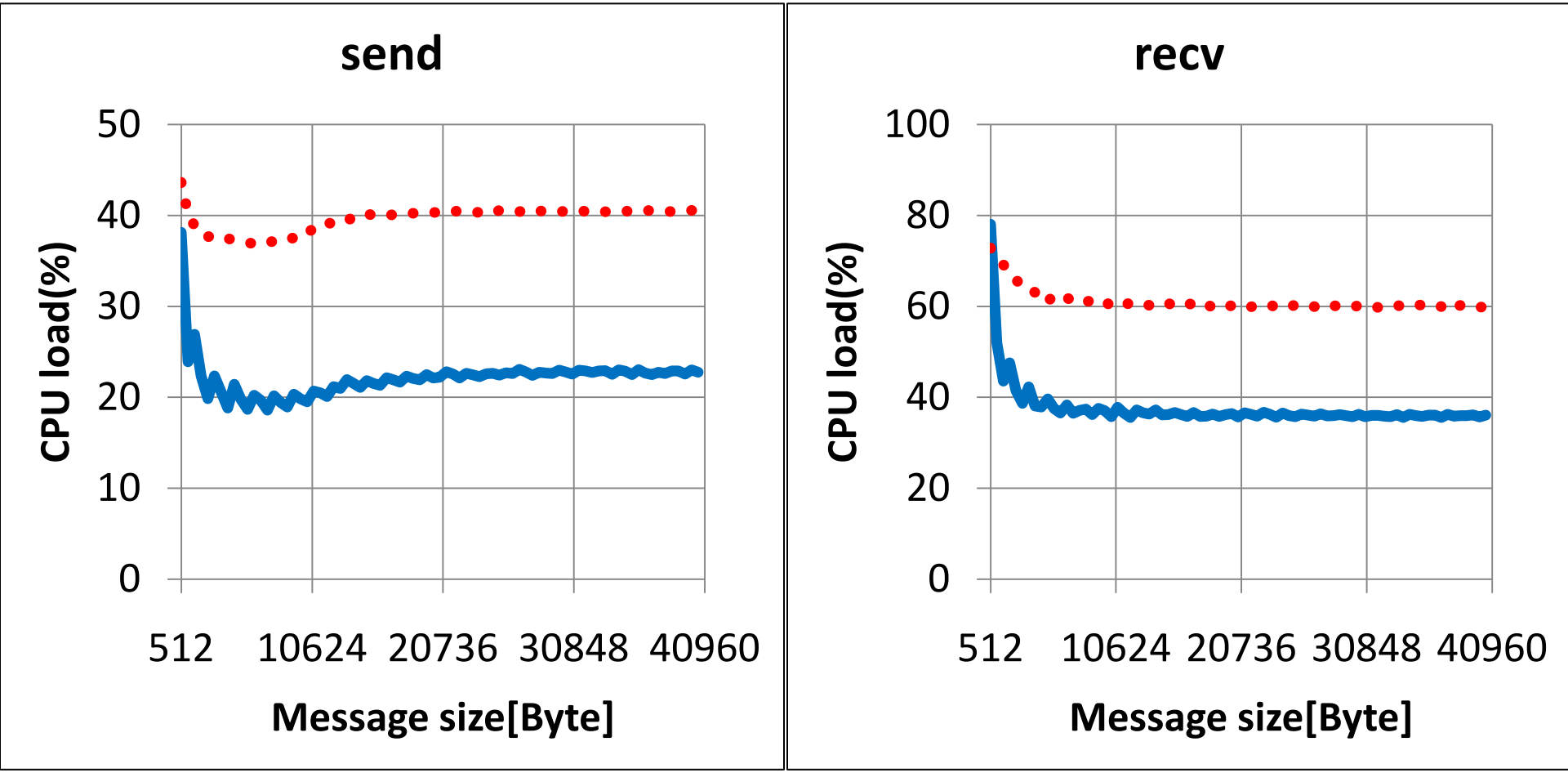
TCP/IP:



- 1500Byte – 40Byte = 1460Byte.

(1) CPU load

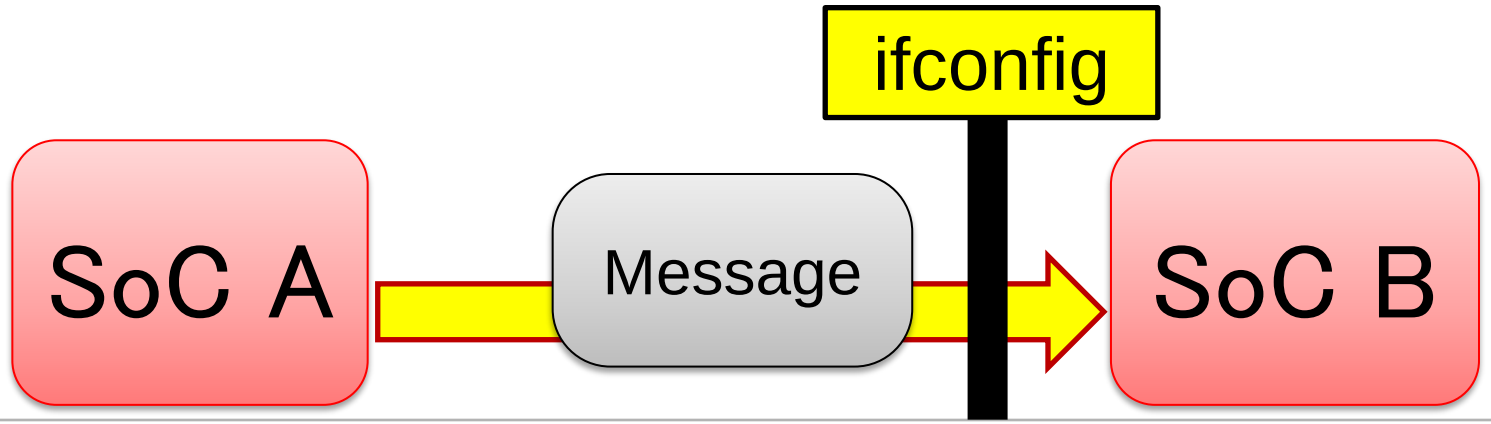
- CPU load of TIPC is less than TCP/IP



Packets observance

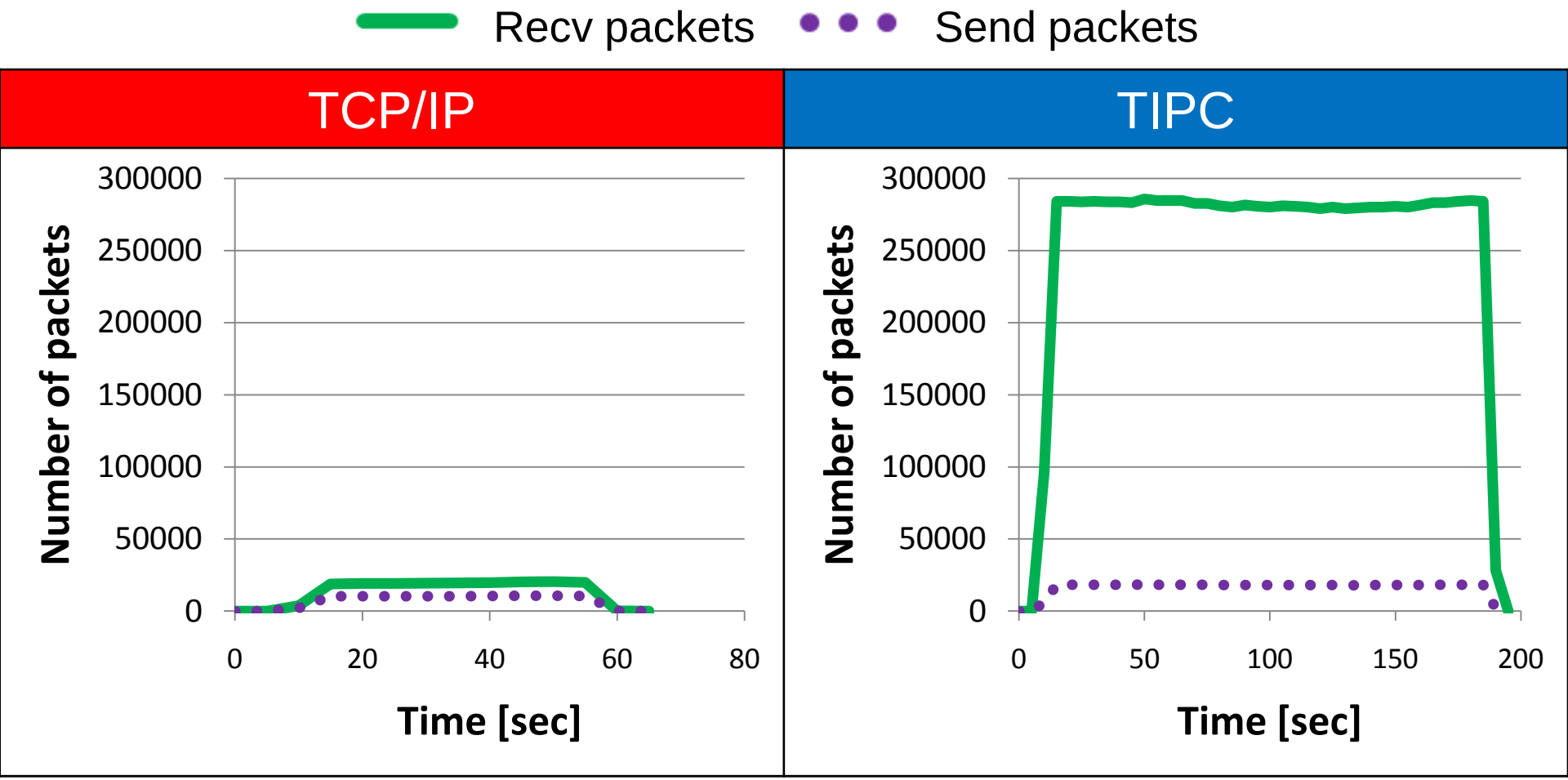
- **TIPC performance in case of small size packets**
  - Counting RX/TX packets.

```
eth0      Link encap:Ethernet  HWaddr XX:XX:XX:XX:XX:XX
          inet addr:X.X.X.X  Bcast:X.X.X.X  Mask:X.X.X.X
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```



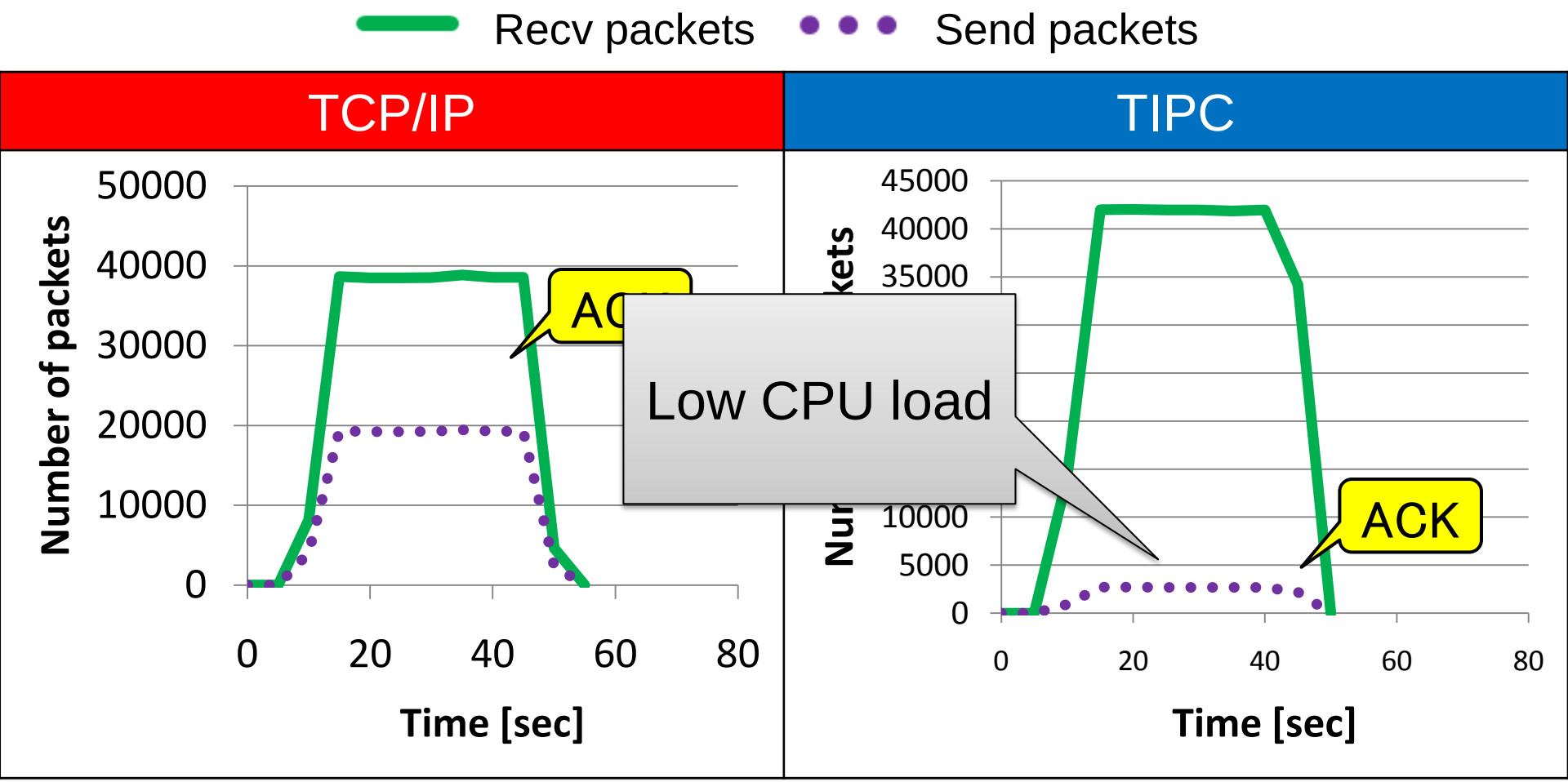
Message Packets: TCP/IP < TIPC

- Message size: 512 Byte
- TCP/IP small packets coalesce.



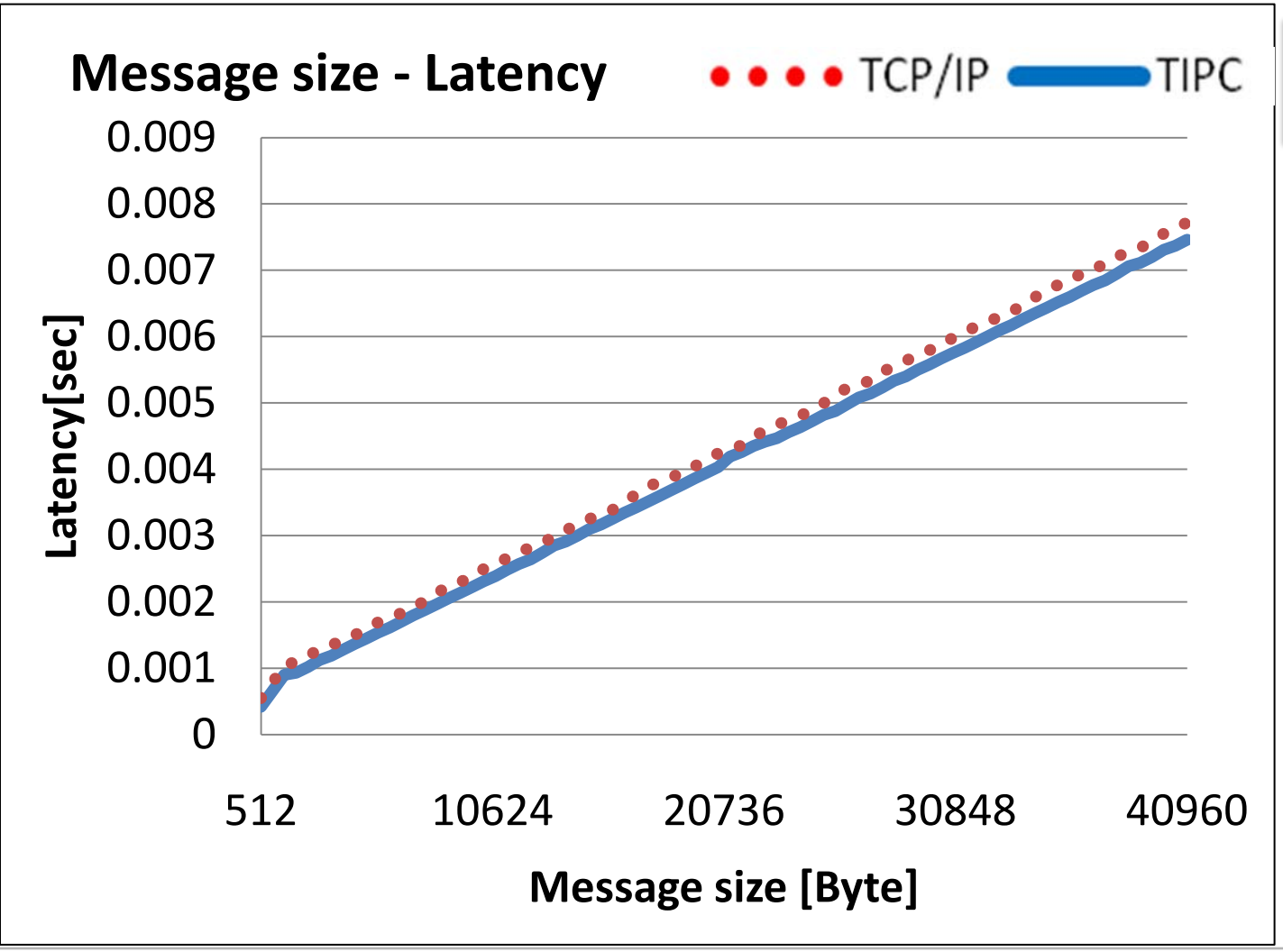
ACK: TCP/IP > TIPC

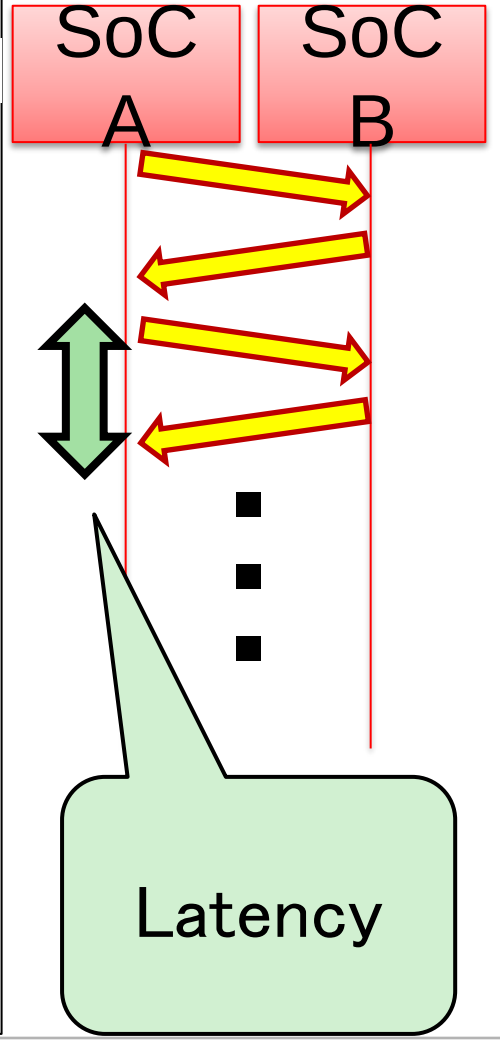
- Message size: 4096 Byte
- TIPC is less frequent ACK.



(2) Latency

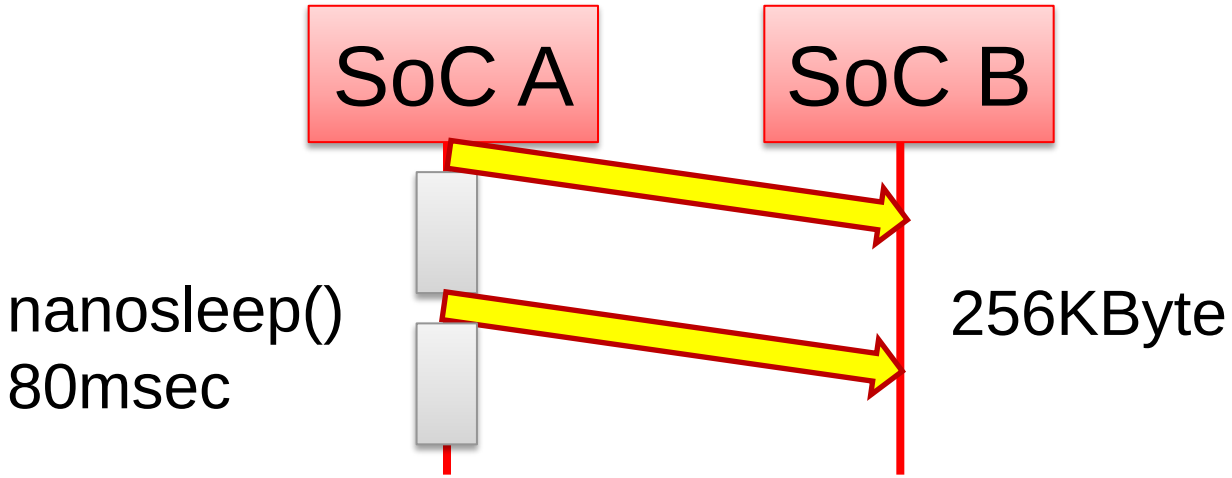
- TCP/IP's latency nearly equals TIPC's.





(3) CPU load in sending streaming data.

- Data transmission rate: 25Mbps



CPU load[%]

	TCP/IP	TIPC	TCP/IP - TIPC
SoC A(send)	9.16	5.18	3.98
SoC B(recv)	19.25	10.25	9.00

Use TIPC (1/2)

- **TIPC doesn’t support SO_SNDTIMEO.**
  - **→ Blocking-send is hard to use.**
  - **→ Nonblocking-send repeat.**
    - **→ CPU load increases. (test program: ~5%)**

Use TIPC (2/2)

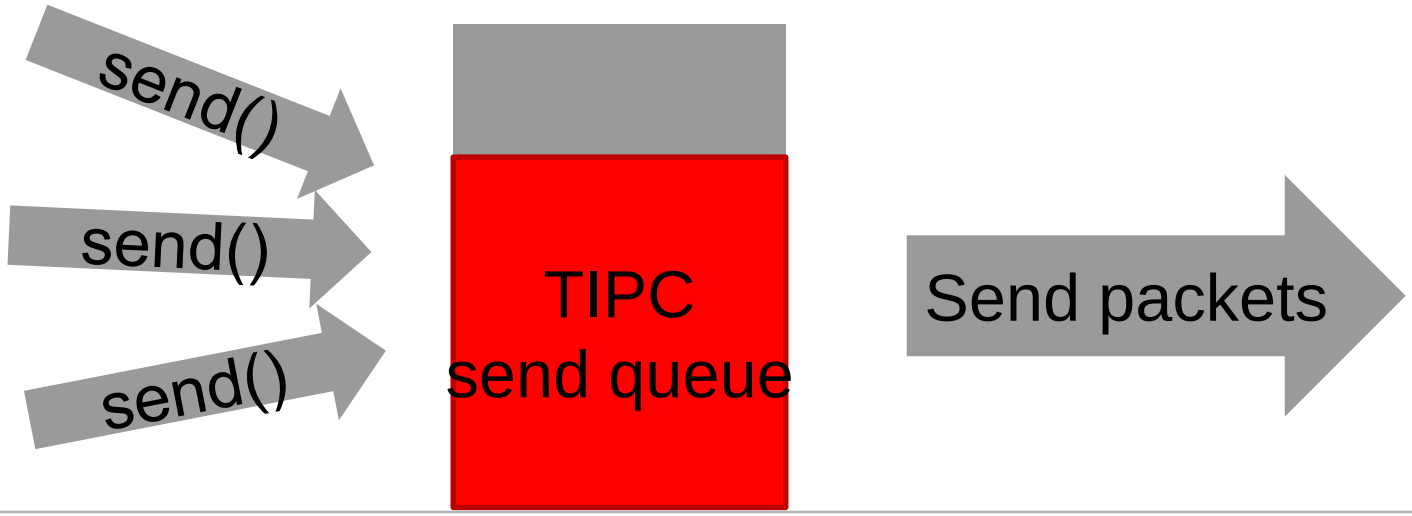
- **Processes/threads issue a lot of send request → TIPC send queue is filling up.**

  (congestion control)

  – Increase processes waiting for queue to decrease.

  – Even if queue decreases, and processes get up, wait again without issue a send request.

  – Need to schedule between processes.



Conclusion

- **TIPC is faster than TCP/IP**
  - Smaller header size
  - TCP/IP is faster in case of small message size.
    - Merging small messages would improve the throughput.
- **TIPC has lower CPU load than TCP/IP**
  - Less frequent ACK (other reasons would exist)

- **Verified that TIPC is useful for embedded systems.**
  - (e.g. Reducing CPU load when transferring streaming data.)

Reference

- **Reference:**

1. TIPC Home Page, http://tipc.sourceforge.net/
2. Florian Westphal, 2007, TIPC Analysis and Optimization, http://webuser.hs-furtwangen.de/~reich/AdvancedMiddlewareWorkshop.SS07
3. Mitsuhiro Kimura: “Application of TIPC to communicate between CPUs in an embedded device and performance comparison to TCP/IP”, The 73rd National Convention of IPSJ 6G-3 1-43 – 1-44 (2011-3)