Pktgen receiver tool

Daniel Turull (KTH)

31st March 2011

Outline

1. Introduction
2. Design
3. Evaluation
4. Conclusions

Outline

1 Introduction

2 Design

3 Evaluation

4 Conclusions

Introduction

- Traffic analysis is crucial for developlment of network systems
- New features in modern systems
  - Multicore
  - Multiples queues in network cards
- Pktgen: packet **generator** at high rates inside Linux Kernel

Network analysis

Diferent aproaches

- **Dedicated hardware:** *Ixia IxNetwork, Spirent SmartBits*
- **Software based:**
  - Libraries: *Pcap, Ncap, DashCap*
  - User space: *Iperf, Netperf, NetPIPE, LMBench, Ttcp, nuttcp, Mausezahn, D-ITG, Harpoon, RUDE, BRUTE*
  - Kernel space: *Pktgen, Kute*
- **Network processors:** *Caldera Technologies - LANforge-FIRE, TNT Pktgen, BRUNO*

Outline

1 Introduction

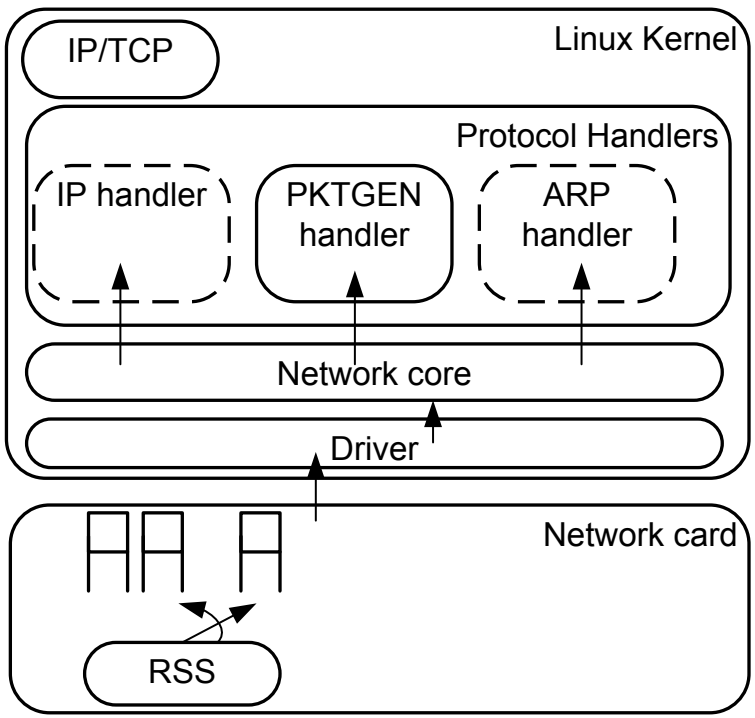
2 Design

3 Evaluation

4 Conclusions

Architecture

- Each CPU has its counters and variables for the different flows or different NICS
- Multiqueue support
- Load balancing (configured via SMP affinity)



Operation

Layer 3

- Advantages
  - No device dependent and more generic
  - Transparent to other communications
  - The reception is made by the kernel
- Drawbacks
  - Less performance (Theoretically)

Auto-configuration

- New pktgen header
- Configure packet at the beginning (pkts to send, bytes to send)
- Reset counters

Receiver metrics (I)

- Packets / Byte Received
  - Counters
- Packet / Byte loss
  - Offline. Subtract (Data extracted from initialization)
- Throughput
  - Time first packet arrive, Time last packet arrive

$$\text{Throughput} = \frac{\text{packets received}}{\text{end time} - \text{start time}} \text{ (pps)}$$

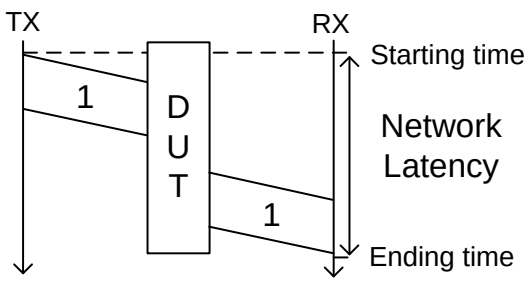
$$\text{Throughput} = \frac{\text{bytes received} \times 8}{\text{end time} - \text{start time}} \text{ (bps)}$$

Receiver metrics (II)

- Inter-arrival time: avg, var, max, min

$$\text{Inter arrival time} = T_{current} - T_{last\ arrival}$$

- Jitter: avg, var, max, min
  - Necessary constant rate
  - Method used: Inter-arrival
  - Subtract of two consecutive inter-arrival times
- Latency: avg, var, max, min



Aplication intereface

Control:

- Transmission:
  - rate [rate in Mbps]
  - ratep [rate in pps]
  - config [0 or 1]
- Reception:

  new proc file: /proc/net/pktgen/pgrx
  - rx [device]
  - rx_reset
  - rx_disable
  - display [human or script]
  - statistics [counter, basic, basic6 or time]

Display: per CPU and Global

Example of receiver intereface

```
File Edit View Terminal Help
host2:/# cat /proc/net/pktgen/pgrx
                RECEPTION STATISTICS
        PER-CPU Stats
CPU 0:  Rx packets: 0    Rx bytes: 0
CPU 1:  Rx packets: 18095827     Rx bytes: 1085749620
        Work time 30004739 us
        Rate:  603098pps 289Mb/sec (289487309bps)
        Inter-arrival
                Average: 1658 ns Variance 176085481 ns2
                Max: 394241 ns Min:: 563 ns
                Samples: 18095826
        Jitter
                Average: 1957 ns Variance 348300932 ns2
                Max: 393330 ns Min:: 0 ns
                Samples: 18095825
        Latency
                Average: 5387191 ns Variance 18446715568905749193 ns2
                Max: 5941779 ns Min:: 471175 ns
                Samples: 18095827
CPU 2:  Rx packets: 0    Rx bytes: 0
```

Outline

1 Introduction

2 Design

3 Evaluation

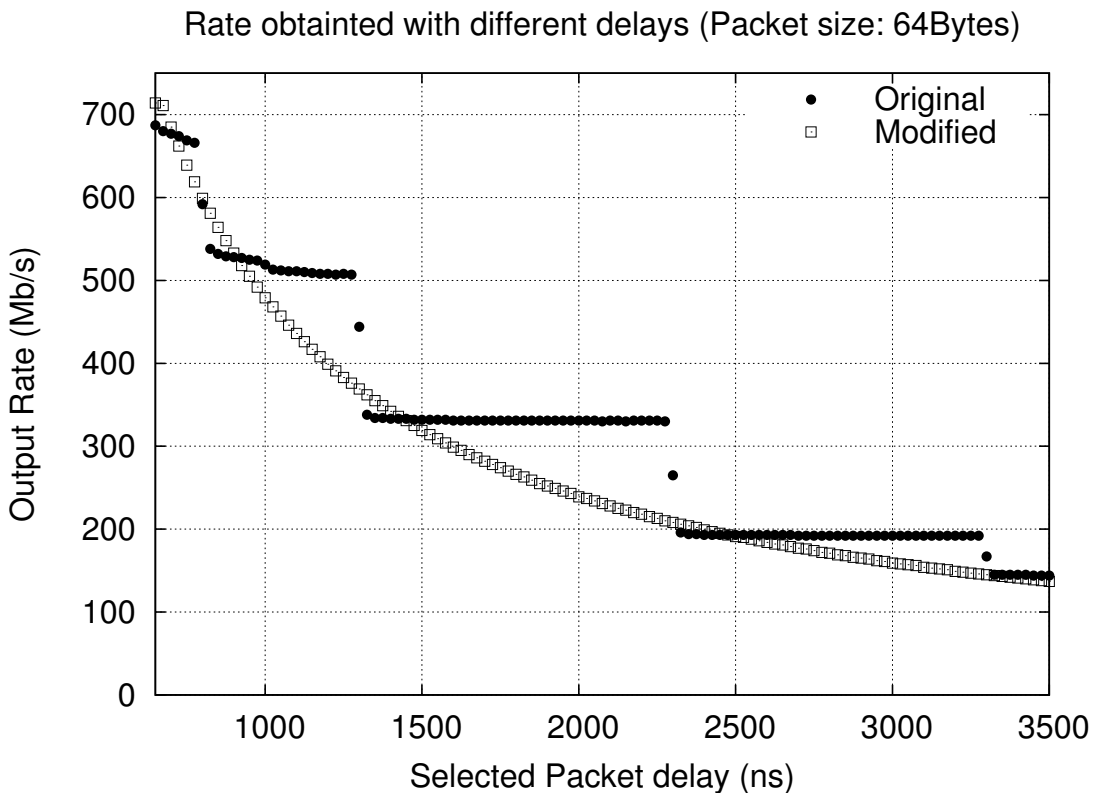
4 Conclusions

Scenario

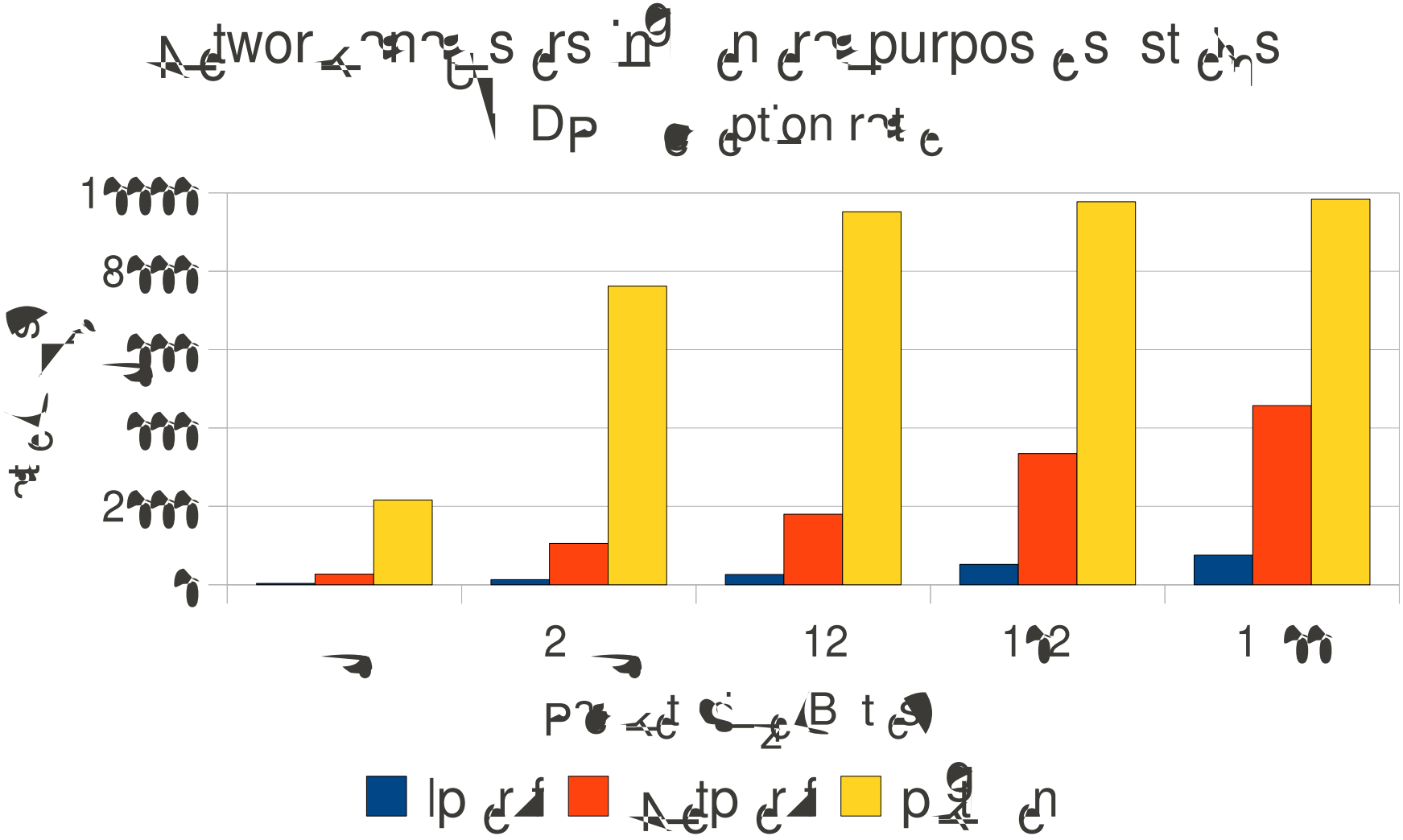
- Intel(R) Xeon(R) CPU E5520 at 2.27GHz
  (Quad-Core Hyperthreading)
- 3 GB of RAM (DDR3 1333MHz)
- 4 Intel 82576 Gigabit Network (2 x Dual Copper Port)
- 2 Intel 82599EB 10-Gigabit Network (1 x Dual Fibre Port)
- Bifrost Distribution.
  Kernel: net-next-2.6 (2.6.34-rc2) (April 2010)

Transmission rate

- Changed resolution from microseconds to nanoseconds
- New commands for a direct control
- Accepted in the Linux Kernel (11 June 2010)

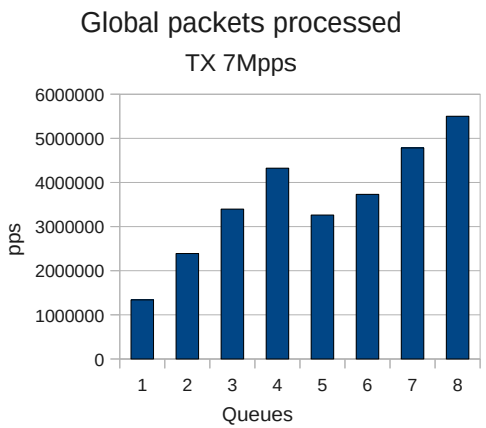


Comparison with iperf and netperf

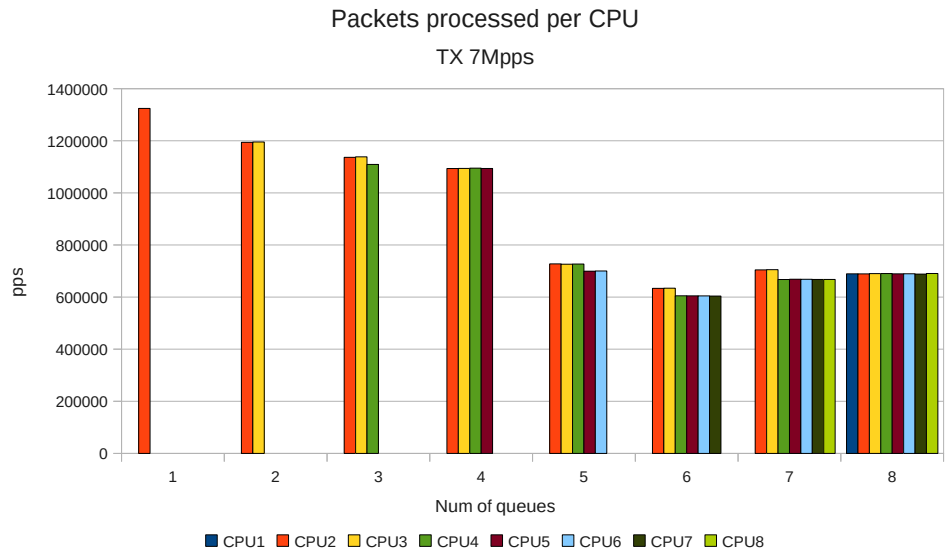


Throughput

RX: depens on num CPUs and type

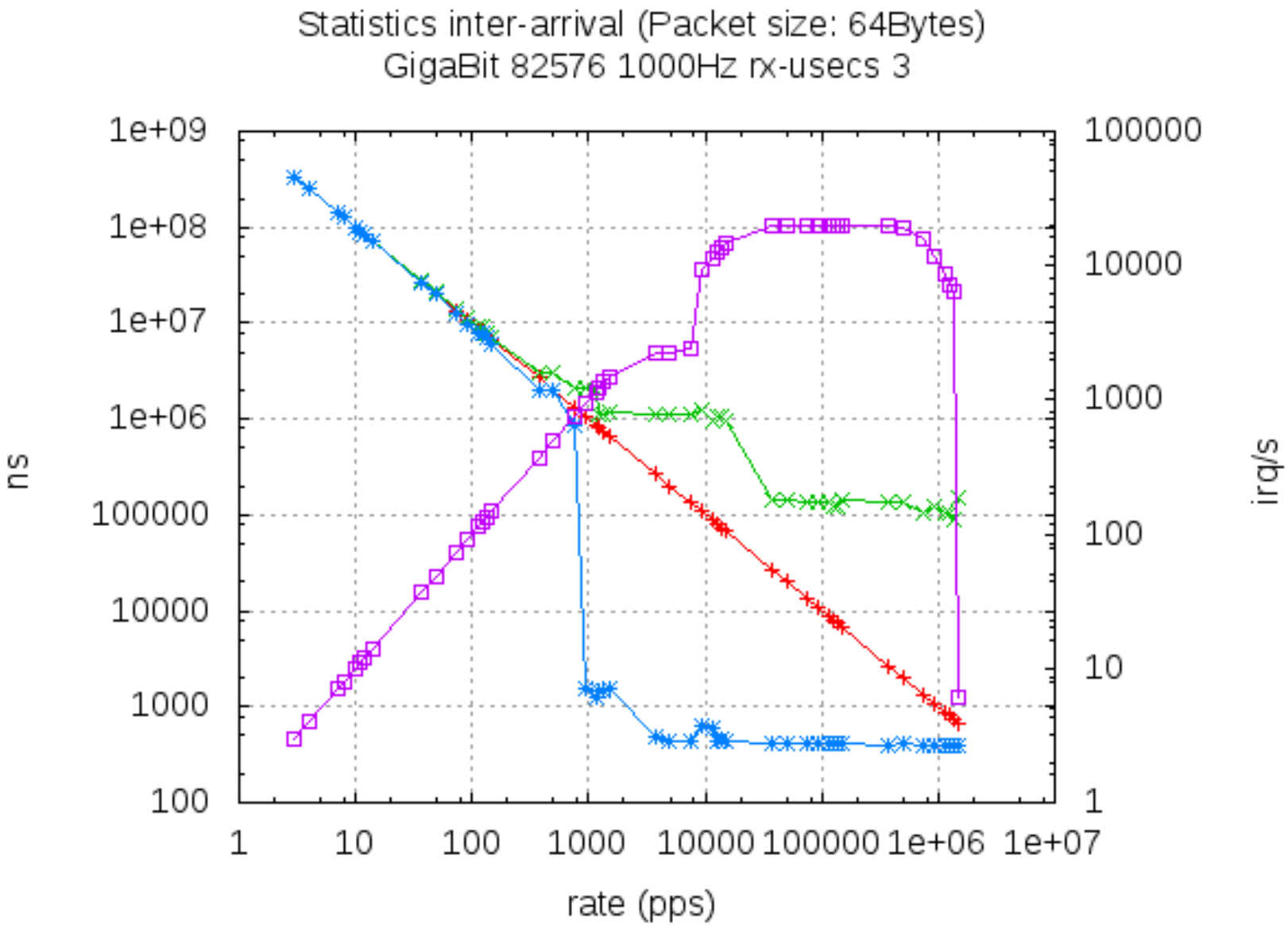


(a) Receiver with different number of RSS queues



(b) Packets processed per CPU

Inter-arrival Time



Outline

1 Introduction

2 Design

3 Evaluation

4 Conclusions

Conclusions

- Used new features of modern network cards and SMP systems
- Receiver side statistics for different scenarios
  - Counters, basic, basic6, time
- Receiver is a powerful tool to understand how the Linux kernel behave
  - Receiving packets in SMP
  - Inter-arrival time and jitter
  - Latency in function of the rate
- Displaying results in human and script readable
- Integrated in current version of pktgen
- Available in:
  - TX: vanilla Linux Kernel (since 2.6.35)
  - RX: `http://tslab.ssvl.kth.se/pktgen` and Bifrost

Thank you for your attention

Any question?

Backup slides

Linux Network (I)

Basic elements in network subsystem:

- Socket buffer (skb)
- Net device

Packet reception

- Interrupt driven
- Polling

NAPI. Advantages of both of them

- Low load (interrupt)
- High load (polling)
- Moreover: Direct access to device memory and no queues

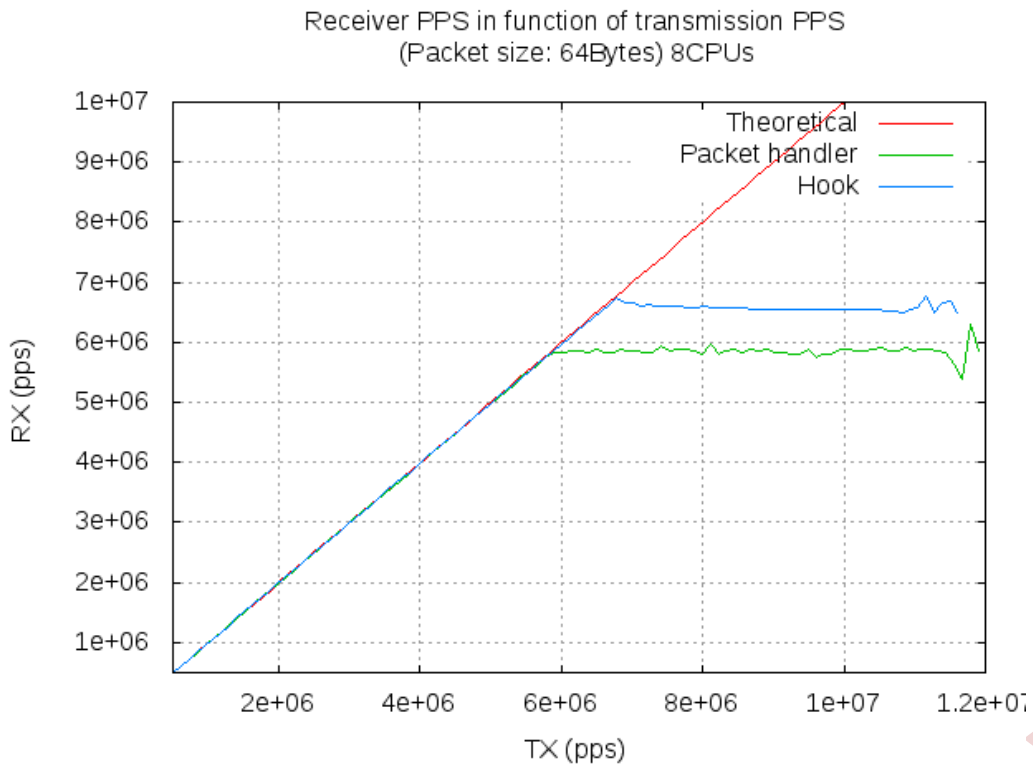
Pktgen Features

- MPLS, VLAN, IPSEC
- IPv4 and IPv6
- Customized packets with multiples addresses
- Clone packets to improve performance
- Multi queue
- Proc file systems as user interface
- Control the delay between packets
- UDP to send its headers

Hook

- Avoid IP process
- Modification of the network core (dev.c)

1. Check if pktgen packet
2. Process packet with pktgen (if pktgen, packet drop)
3. Otherwise, packet continues its path to other protocols



Header split

- Header + Data in different memory regions
- IP stack is were most of the perfomance is drop

Test	Received Rate (No hook)	Received Rate (Hook)
Split headers	5.8 Mpps	6.64 Mpps
NO Split	6.5 Mpps	6.74 Mpps

Inter-arrival Time and Jitter

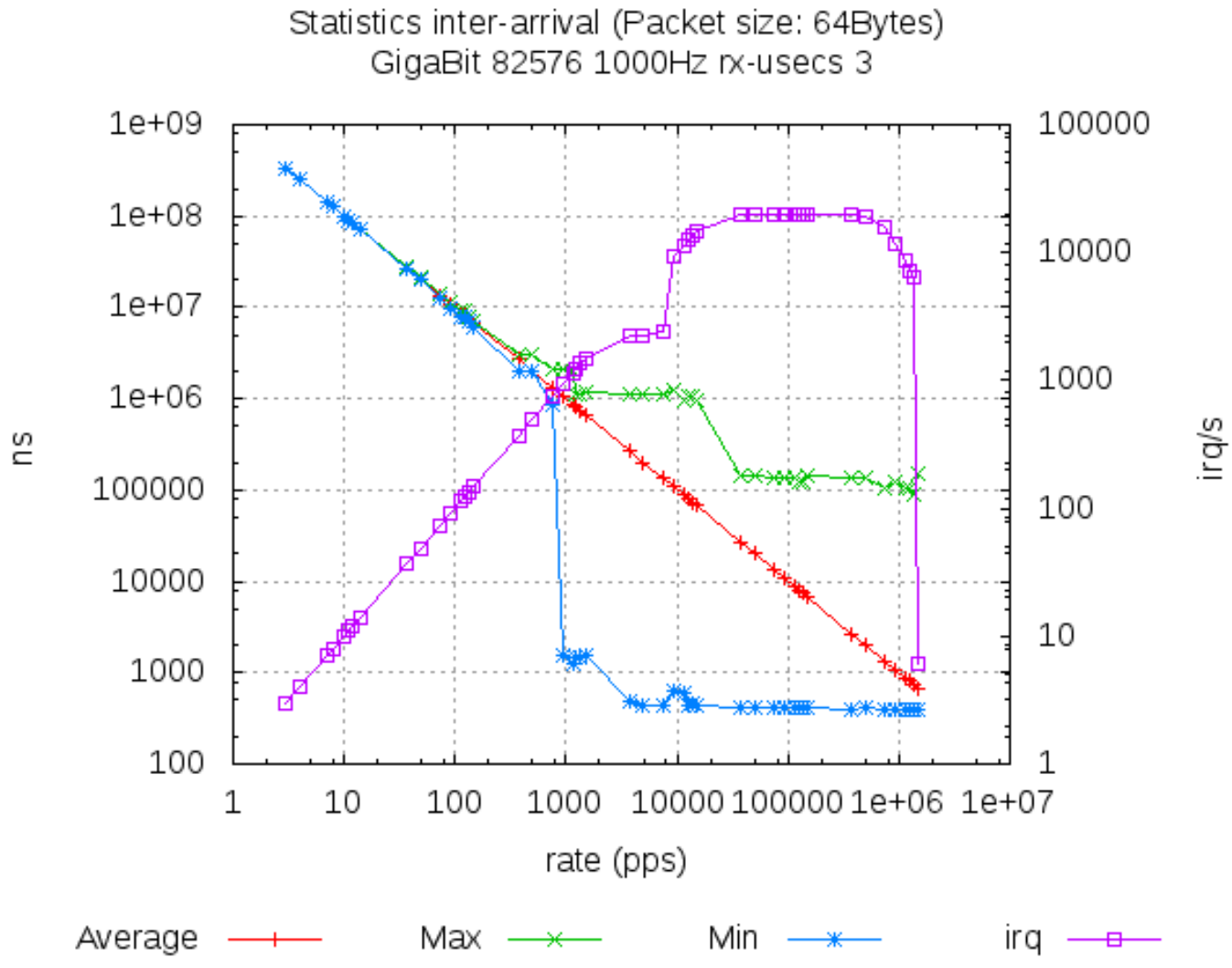


Figure: Inter-arrival. Current

Inter-arrival Time and Jitter

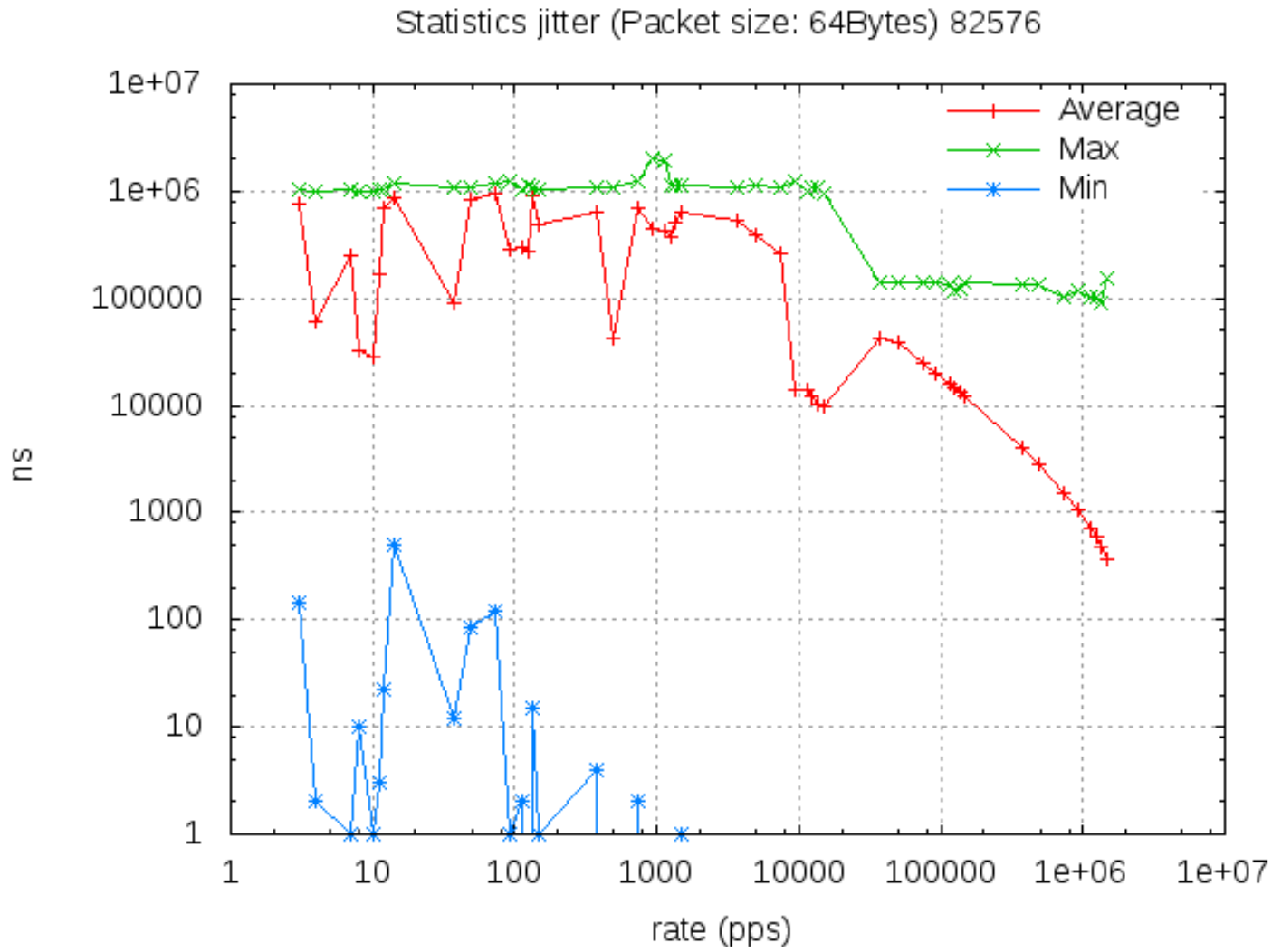


Figure: Jitter. Current

Inter-arrival Time and Jitter

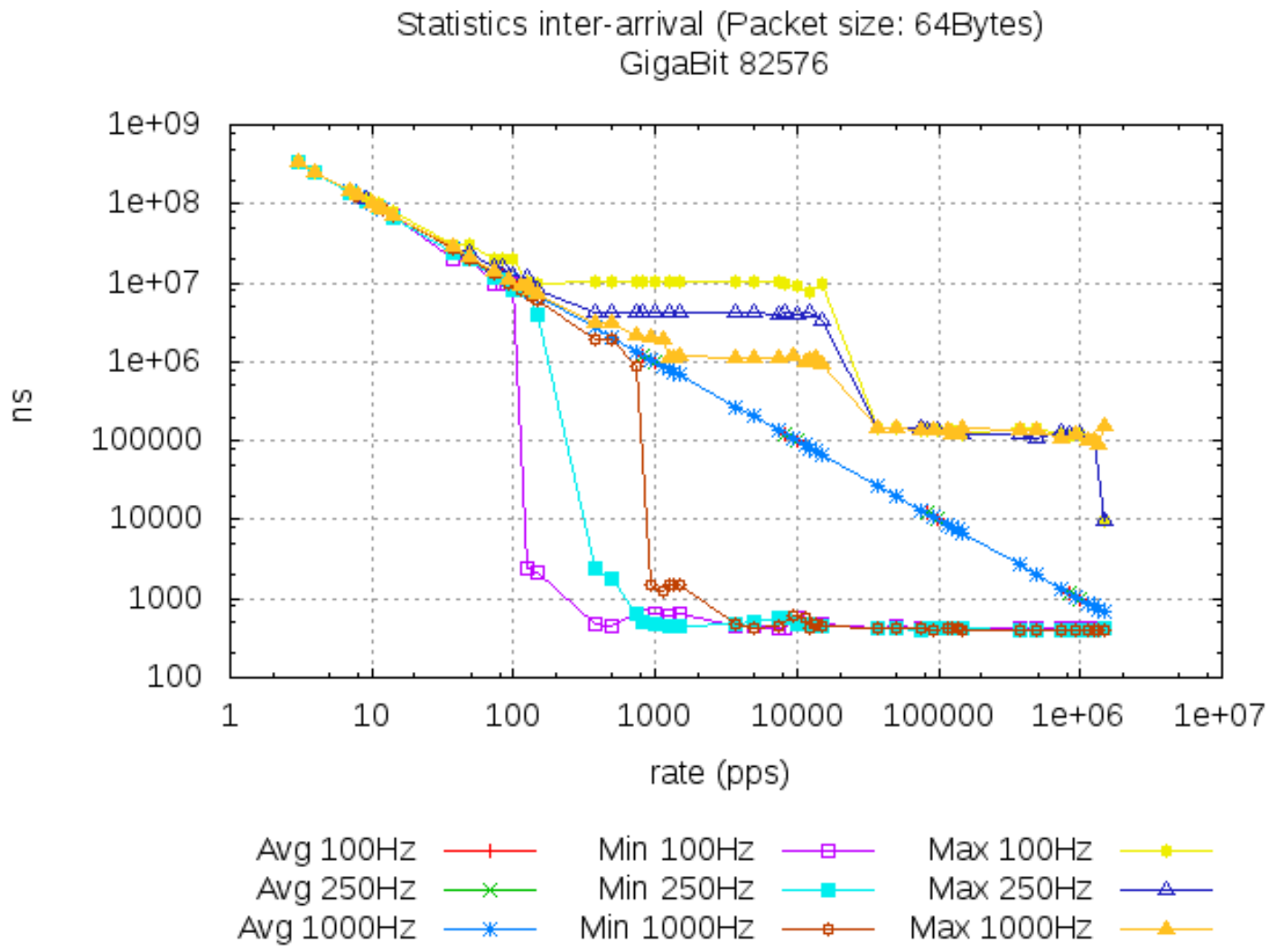


Figure: Inter-arrival. Frequency

Inter-arrival Time and Jitter

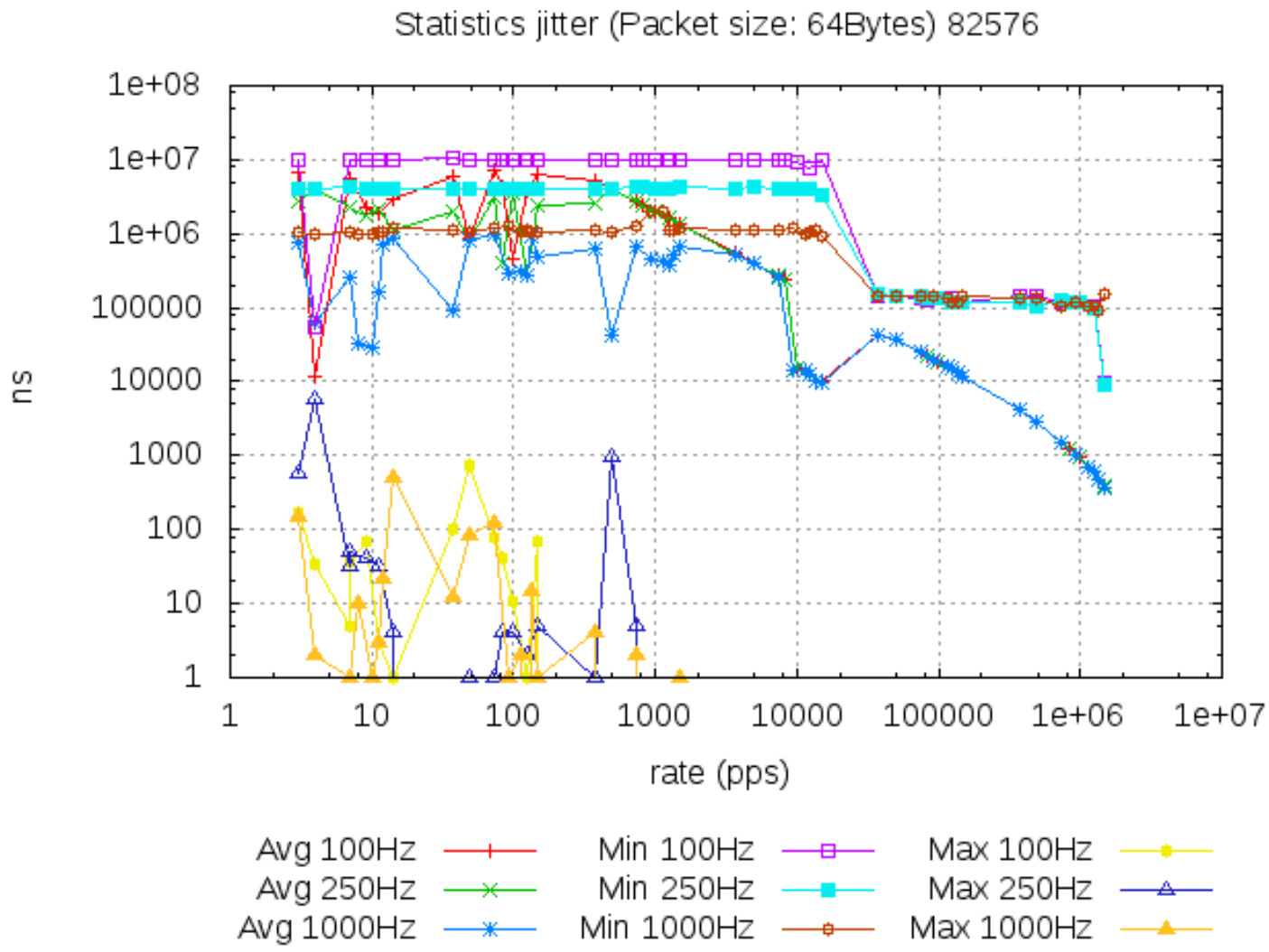


Figure: Jitter. Frequency

Inter-arrival Time and Jitter

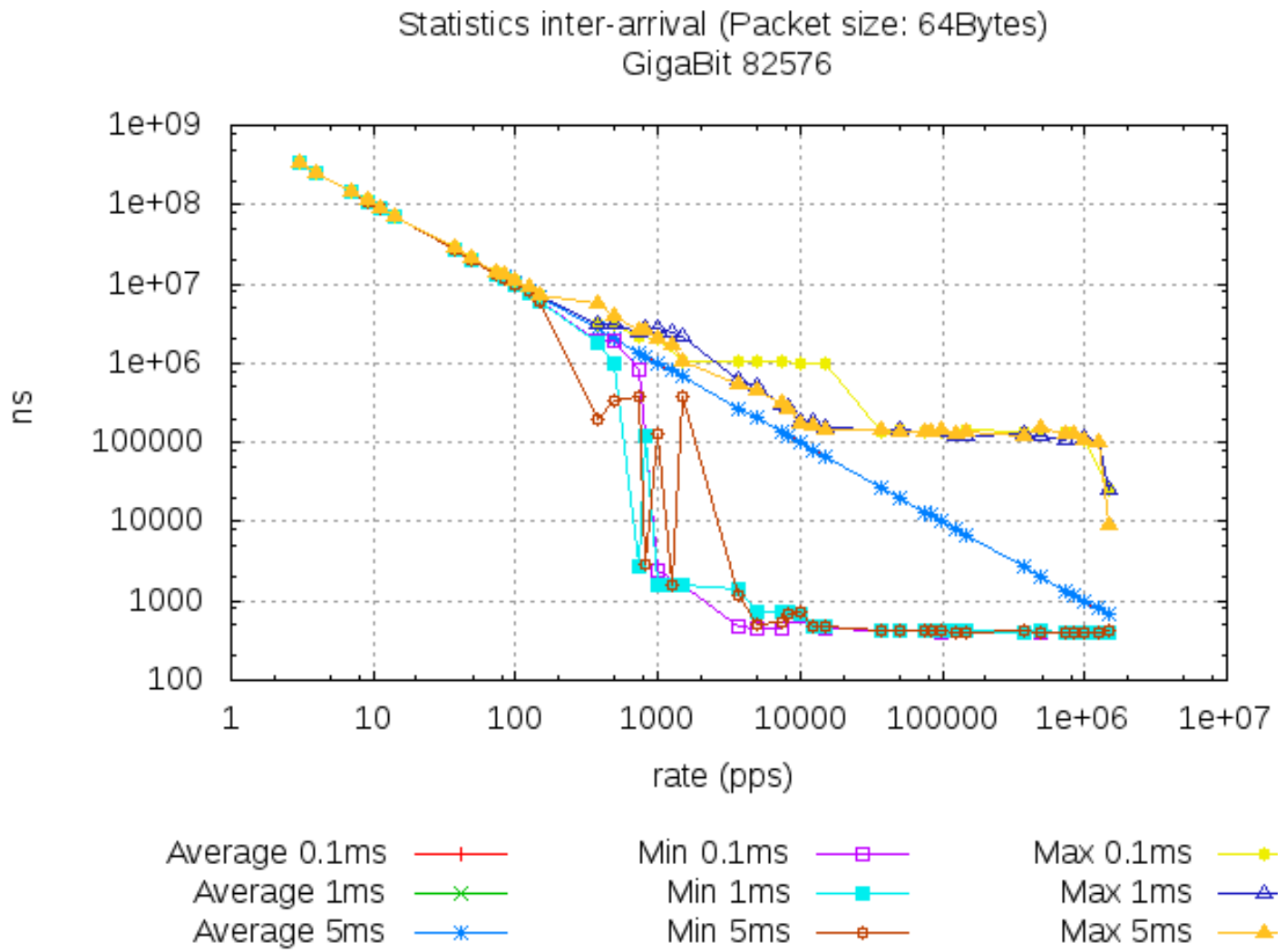


Figure: Inter-arrival. Spin

Inter-arrival Time and Jitter

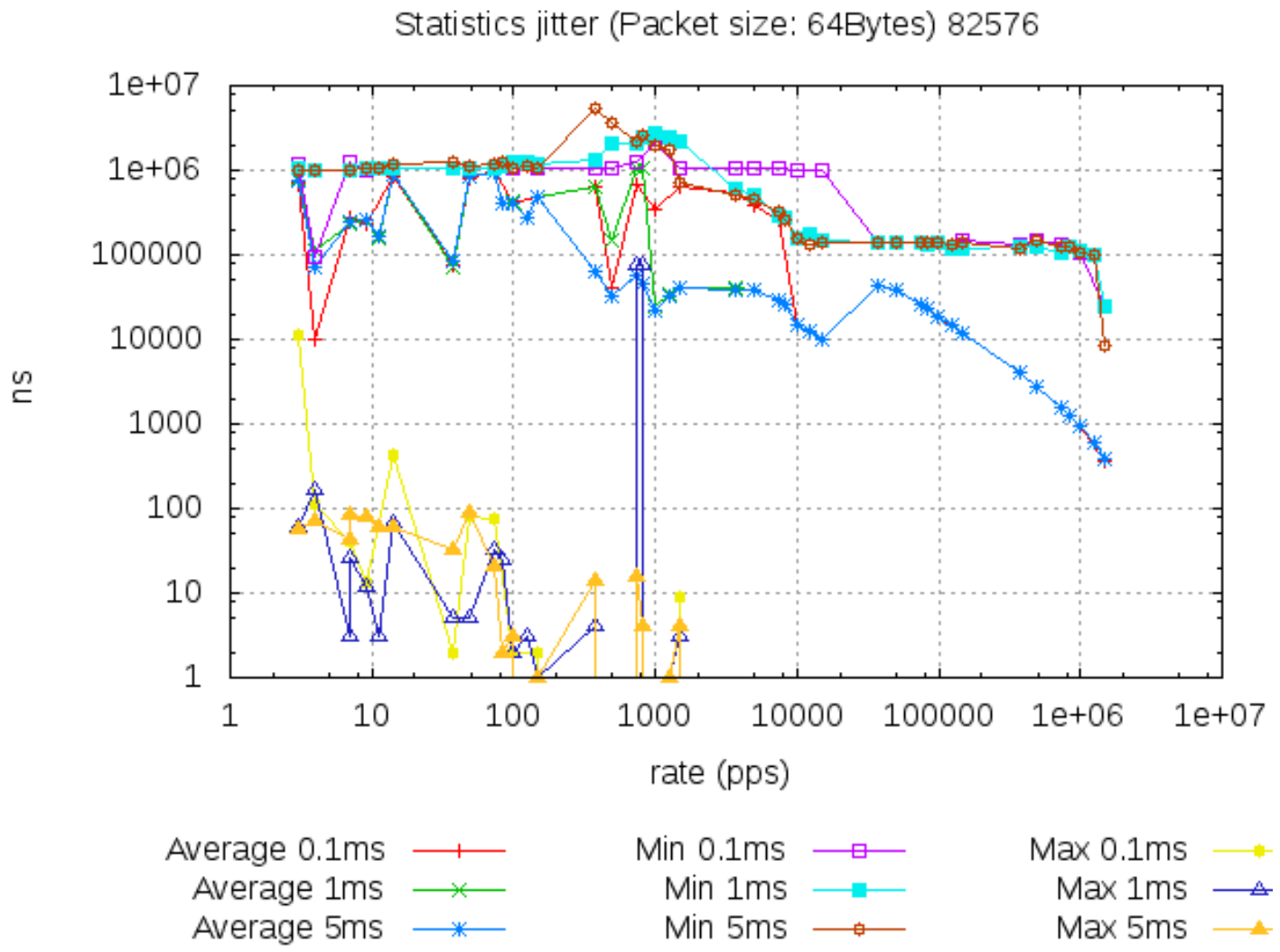


Figure: Jitter. Spin

Inter-arrival Time and Jitter

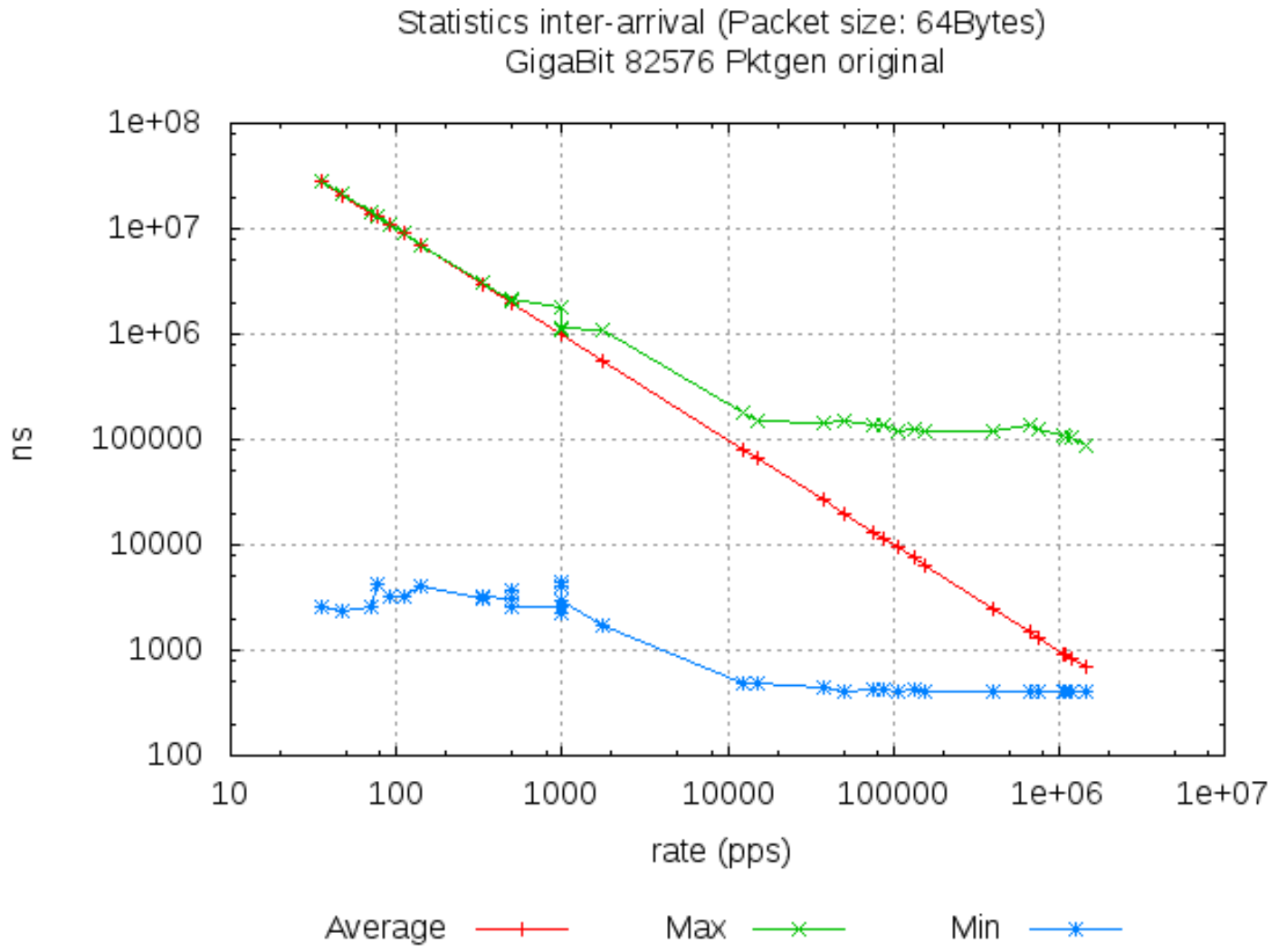


Figure: Inter-arrival. Original

Inter-arrival Time and Jitter

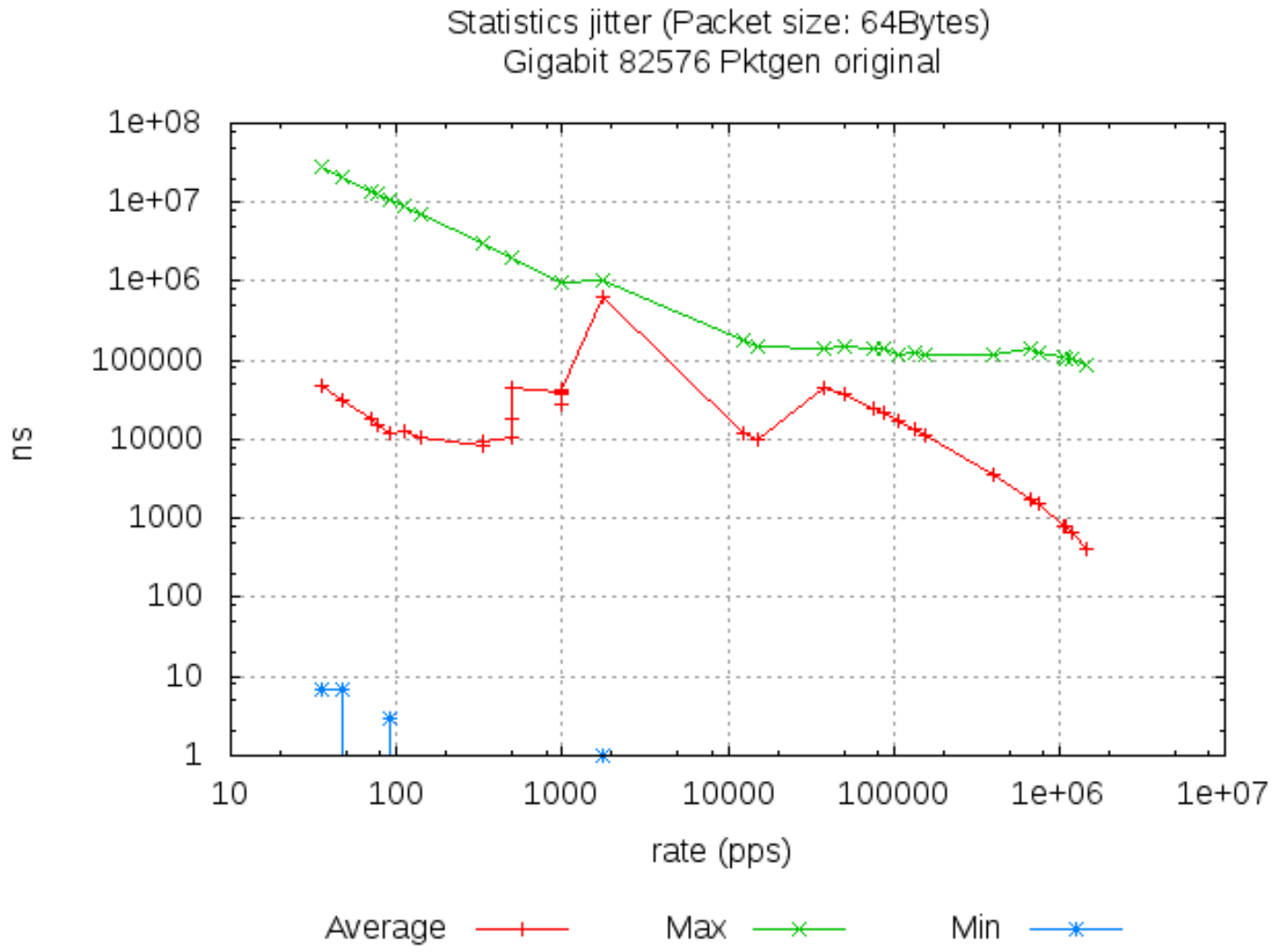
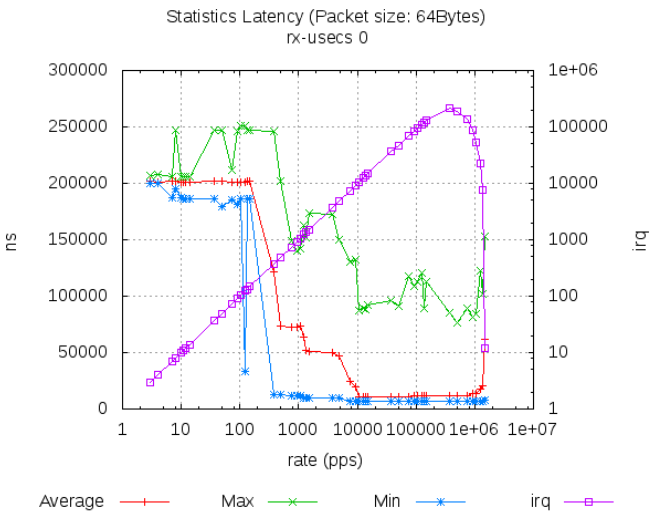


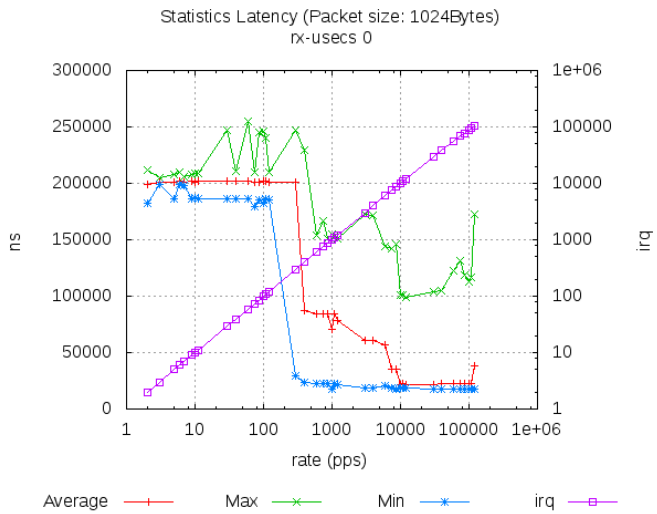
Figure: Jitter. Original

Latency

- TX and RX same machine
- Unexpected behaviour: high latency at low rates



(a) Packet size 64 Bytes



(b) Packet size 1024 Bytes

Comparison between methods of collecting statistics

- Counters
- Basic
- Time