SoNIC: Precise Realtime Software Access and Control of Wired Networks

Ki Suh Lee, Han Wang, Hakim Weatherspoon
Cornell University
Interpacket Delay and Network Research

- Interpacket gap, spacing, arrival time, ...

- Important metric for network research
  - Can be improved with access to the PHY

Packet Generation
Packet Capture
Characterization
Estimating bandwidth
Increasing Throughput
Detecting timing channel

Application
Transport
Network
Data Link
Physical
Network Research enlightened via the PHY

- Valuable information: Idle characters
  - Can provide precise timing base for control
    - Each bit is ~97 ps wide
Network Research enlightened via the PHY

• Valuable information: Idle characters
  - Can provide precise timing base for control
  - Each bit is ~97 ps wide

\[ 12 /I/s = 100 \text{bits} = 9.7 \text{ns} \]

Packet Generation

Detected timing channel

Packet Capture
Principle #1: Precision

Precise network measurements is enabled via access to the physical layer (and the idle characters and bits within interpacket gap)
How to control the idle characters (bits)?

- Access to the entire stream is required
- 

- **Issue1:** The PHY is simply a black box
  - No interface from NIC or OS
  - Valuable information is invisible (discarded)

- **Issue2:** Limited access to hardware
  - We are network systems researchers a.k.a. we like software
Principle #2: Software

Network Systems researchers need software access to the physical layer
Precision + Software = Physics equipment???

- **BiFocals** [IMC’10 Freedman, Marian, Lee, Birman, Weatherspoon, Xu]
  - Enabled novel network research
  - Precision + Software = Laser + Oscilloscope + Offline analysis
  - Allowed precise control in software

- **Limitations**
  - Offline (not *realtime*)
  - Limited Buffering
  - Expensive
Principle #3: Realtime

Network systems researchers need access and control of the physical layer (interpacket gap) continuously in realtime.
Challenge

• **Goal:** Control *every* bit in *software* in *realtime*

  – Enable novel network research

• **Challenge**

  – Requires unprecedented software access to the PHY
Outline

• Introduction

• SoNIC: Software-defined Network Interface Card
  – Background: 10GbE Network Stack
  – Design

• Network Research Applications

• Conclusion
SoNIC: Software-defined Network Interface Card

- Implements the PHY in software
  - Enabling control and access to every bit in realtime
  - With commodity components
  - Thus, enabling novel network research

- How?
  - Backgrounds: 10 GbE Network stack
  - Design and implementation
    - Hardware & Software
    - Optimizations
10GbE Network Stack

Application
Transport
Network
Data Link
Physical

64/66b PCS
Encode
Scrambler
Gearbox
PMA

Decode
Descrambler
Blocksync
PMD

Commodity NIC

SW
HW

Packet i
Packet i+1

L2 Hdr
L3 Hdr
Data

Data

L2 Hdr
L3 Hdr


0110100101101001011010010110100101101001011010010110100101101001
10GbE Network Stack

- Application
- Transport
- Network
- Data Link

Physical
- 64/66b PCS
  - Encode
  - Decode
  - Scrambler
  - Descrambler
  - Gearbox
  - Blocksync
- PMA
- PMD

SW

Packet i

Packet i+1

HW

SoNIC

NetFPGA

SoNIC NSDI 2013

4/11/13
SoNIC Design

Application
Transport
Network
Data Link

Physical
64/66b PCS
Encode
Decode
Scrambler
Descrambler
Gearbox
Blocksync
PMA
PMD

Data

L3 Hdr
Data

L2 Hdr
L3 Hdr
Data

Preamble
Eth Hdr
L2 Hdr
L3 Hdr
Data
CRC
Gap


011010010110100101101001011010010110100101101001011010010110100101101
SoNIC Design and Architecture

Application
Transport
Network
Data Link

Physical
64/66b PCS
- Encode
- Decode
- Scrambler
- Descrambler
- Gearbox
- Blocksync
PMA
PMD

Data Link
Application
Transport
Network

SoNIC

Physical Components:
- Gearbox
- Blocksync
- Transceiver
- SFP+

SoNIC SW/HW Components:
- TX PCS
- RX PCS
- TX MAC
- RX MAC

Layers:
- L2 Hdr
- L3 Hdr
- L3 Hdr
- L2 Hdr
- GAP
- Eth Hdr
- Preamble
- Data

Network Transport
Data Link
Application

SoNIC NSDI 2013
SoNIC Design: Hardware

- To deliver every bit from/to software
  - High-speed transceivers
  - PCIe Gen2 (=32Gbps)
- Optimized DMA engine
SoNIC Design: Software

- Dedicated Kernel Threads
  - TX / RX PCS, TX / RX MAC threads
  - APP thread: Interface to userspace
SoNIC Design: Synchronization

Application
Transport
Network
Data Link
Physical
64/66b PCS
  Encode
  Decode
Scrambler
Descrambler
Gearbox
Blocks sync
PMA
PMD

Port 0
- APP
- TX MAC
- TX PCS
- RX MAC
- RX PCS

Port 1
- APP
- TX MAC
- TX PCS
- RX MAC
- RX PCS

Low-latency FIFOs

Pointer-polling
No Interrupts

SW
HW
SoNIC Design: Optimizations

- Scrambler

\[ G(x) = x^{58} + x^{39} + 1 \]

<table>
<thead>
<tr>
<th>Naïve Implementation</th>
<th>Optimized Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s \leftarrow \text{state} )</td>
<td>( s \leftarrow \text{state} )</td>
</tr>
<tr>
<td>( d \leftarrow \text{data} )</td>
<td>( d \leftarrow \text{data} )</td>
</tr>
<tr>
<td>( \text{for } i = 0 \rightarrow 63 ) do</td>
<td>( r \leftarrow (s &gt;&gt; 6) \oplus (s &gt;&gt; 25) \oplus d )</td>
</tr>
<tr>
<td>( \quad \text{in} \leftarrow (d &gt;&gt; i) &amp; 1 )</td>
<td>( r \leftarrow r \oplus (r &lt;&lt; 39) \oplus (r &lt;&lt; 58) )</td>
</tr>
<tr>
<td>( \quad \text{out} \leftarrow (\text{in} \oplus (s &gt;&gt; 38) \oplus (s &gt;&gt; 57)) &amp; 1 )</td>
<td>( \text{state} \leftarrow r )</td>
</tr>
<tr>
<td>( \quad s \leftarrow (s &lt;&lt; 1) \oplus \text{out} )</td>
<td>( )</td>
</tr>
<tr>
<td>( \quad r \leftarrow r \oplus (\text{out} &lt;&lt; i) )</td>
<td>( )</td>
</tr>
<tr>
<td>( \text{state} \leftarrow s )</td>
<td>( )</td>
</tr>
<tr>
<td>( \text{end for} )</td>
<td>( )</td>
</tr>
</tbody>
</table>

0.436 Gbps

- CRC computation
- DMA engine
SoNIC Design: Interface and Control

- Hardware control: *ioctl* syscall
- I/O: character device interface
- Sample C code for packet generation and capture

```c
#include "sonic.h"

struct sonic_pkt_gen_info info = {  
  .mode = 0,
  .pkt_num = 1000000000UL,
  .pkt_len = 1518,
  .mac_src = "00:11:22:33:44:55",
  .mac_dst = "aa:bb:cc:dd:ee:ff",
  .ip_src = "192.168.0.1",
  .ip_dst = "192.168.0.2",
  .port_src = 5000,
  .port_dst = 5000,
  .idle = 12,
};

/* OPEN DEVICE*/
f1 = open(SONIC_CONTROL_PATH, O_RDWR);
fd2 = open(SONIC_PORT1_PATH, O_RDONLY);

/* CONFIG SONIC CARD FOR PACKET GEN*/
ioctl(fd1, SONIC_IOC_RESET)
ioctl(fd1, SONIC_IOC_SET_MODE, PKT_GEN_CAP)
ioctl(fd1, SONIC_IOC_PORT0_INFO_SET, &info)

/* START EXPERIMENT*/
ioctl(fd1, SONIC_IOC_START)
// wait till experiment finishes
ioctl(fd1, SONIC_IOC_STOP)

/* CAPTURE PACKET */
while ((ret = read(fd2, buf, 65536)) > 0) {
  // process data
}

/* OPEN DEVICE*/

ioctl(fd1, SONIC_IOC_RESET)
ioctl(fd1, SONIC_IOC_SET_MODE, PKT_GEN_CAP)
ioctl(fd1, SONIC_IOC_PORT0_INFO_SET, &info)

/* START EXPERIMENT*/
ioctl(fd1, SONIC_IOC_START)
// wait till experiment finishes
ioctl(fd1, SONIC_IOC_STOP)

/* CAPTURE PACKET */
while ((ret = read(fd2, buf, 65536)) > 0) {
  // process data
}
```

4/11/13
Outline

• Introduction
• SoNIC: Software-defined Network Interface Card
• Network Research Applications
  – Packet Generation
  – Packet Capture
  – Covert timing channel
• Conclusion
Network Research Applications

- Interpacket delays and gaps

![Diagram showing network research applications with interpacket delays and gaps.](image-url)
Packet Generation and Capture

- **Basic functions for network research**
  - **Generation:** SoNIC allows control of IPGs in # of /l/s
  - **Capture:** SoNIC captures what was sent with IPGs in bits

![Diagram showing packet generation and capture with IPG control and capture mechanisms.](image-url)
Packet Generation

- **SoNIC allows precise control of IPGs**

Interpacket delays (ns)

CDF

SoNIC

Sniffer 10G

Specialized NIC

Higher variance

SoNIC

Zero variance!!!

9Gbps, IPD = 13992 bits (1357ns)
Packet Capture

- *SoNIC captures what is sent*

![Diagram showing interpacket delays and packet capture]

- **Interpacket delays (ns)**
  - 9Gbps, IPD = 13992 bits (1357ns)
Covert Timing Channel

• Embedding signals into interpacket gaps.
  – Large gap: ‘1’
  – Small gap: ‘0’

• Covert timing channel by modulating IPGs at 100ns
  • Overt channel at 3 Gbps
  • Covert channel at 250 kbps
  • Over 4-hops with < 1% BER
Covert Timing Channel

- **Modulating IPGS at 100ns scale (=128 /I/s)**

![Graph showing CDF, Interpacket delays, and BER](image)
Contributions

• Network Research
  – Unprecedented access to the PHY with commodity hardware
  – A platform for cross-network-layer research
  – Can improve network research applications

• Engineering
  – Precise control of interpacket gaps (delays)
  – Design and implementation of the PHY in software
  – Novel scalable hardware design
  – Optimizations / Parallelism

• Status
  – Measurements in large scale: DCN, GENI, 40 GbE
Conclusion

• Precise Realtime Software Access to the PHY
• Commodity components
  – An FPGA development board, Intel architecture
• Network applications
  – Network measurements
  – Network characterization
  – Network steganography
• Webpage: http://sonic.cs.cornell.edu
  – SoNIC is available Open Source.
Thank you

Demo tonight!

http://sonic.cs.cornell.edu