OPENFABRICS
ALLIANCE

Analyzing InfiniBand Packets

OpenFabrics Software User Group Workshop

Qian Liu
QGA2@unh.edu
Advisor: Professor Robert D. Russell

University of New Hampshire

Presentation Overview

- 1. Why analyze IB packets
- 2. How to capture IB packets
- 3. Comparison of IB capture tools
- 4. Our use of the tools to analyze packets

1. Why analyze IB packets

- Protocol study, debug, verification, and research
- Monitor IB network performance
- Analyze inter-packet delay (IPD)
- Observe Flow Control and Congestion Control

2. How to capture IB packets

- ibdump…………………………………..Software package running on nodes

  http://www.mellanox.com/

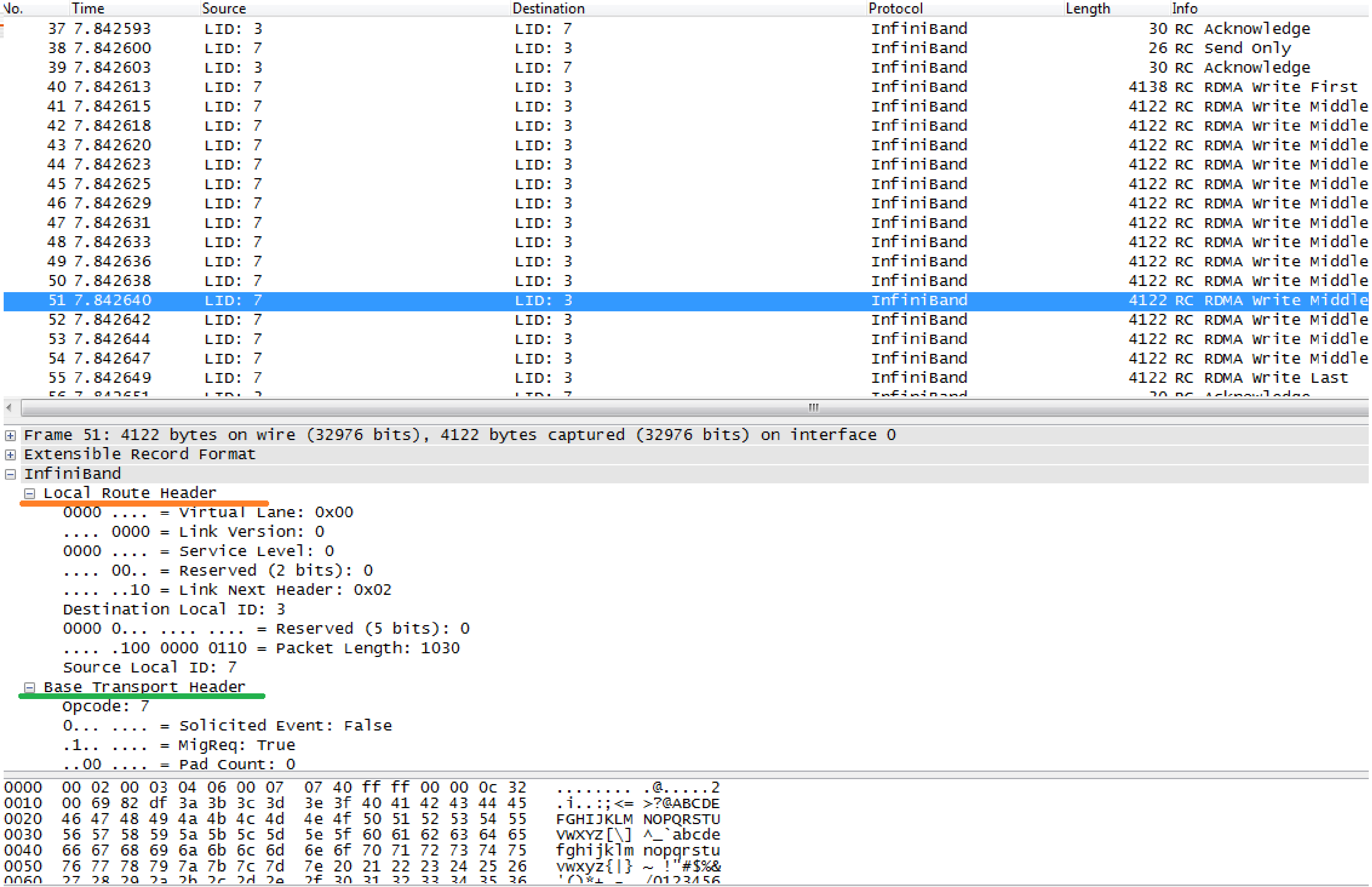
- CatC analyzer……………………….Hardware box inline between ports

  http://www.teledynelecroy.com/

2. How to capture IB packets

- ibdump features
  - Software package freely available from Mellanox Technologies http://downloads.linux.hp.com/downloads/MLNX_OFED/suse/SLES11-SP2/x86_64/2.2_1.0.1/ibdump-2.0.0-8.x86_64.rpm
  - Requires NO physical change to the network
  - Runs on an IB host & Captures packets on an IB interface on that host
  - Works for all IB data rates: SDR, DDR, QDR, FDR10, FDR
  - Dumps a.pcap file which can be loaded by Wireshark http://www.wireshark.com/

Wireshark view of ibdump capture



```
No.  Time       Source    Destination   Protocol     Length  Info
 37 7.842593    LID: 3    LID: 7        InfiniBand       30  RC Acknowledge
 38 7.842600    LID: 7    LID: 3        InfiniBand       26  RC Send Only
 39 7.842603    LID: 3    LID: 7        InfiniBand       30  RC Acknowledge
 40 7.842613    LID: 7    LID: 3        InfiniBand     4138  RC RDMA Write First
 41 7.842615    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 42 7.842618    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 43 7.842620    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 44 7.842623    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 45 7.842625    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 46 7.842629    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 47 7.842631    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 48 7.842633    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 49 7.842636    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 50 7.842638    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 51 7.842640    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 52 7.842642    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 53 7.842644    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 54 7.842647    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Middle
 55 7.842649    LID: 7    LID: 3        InfiniBand     4122  RC RDMA Write Last
 56 7.842651    LID: 3    LID: 7        InfiniBand       30  RC Acknowledge
```

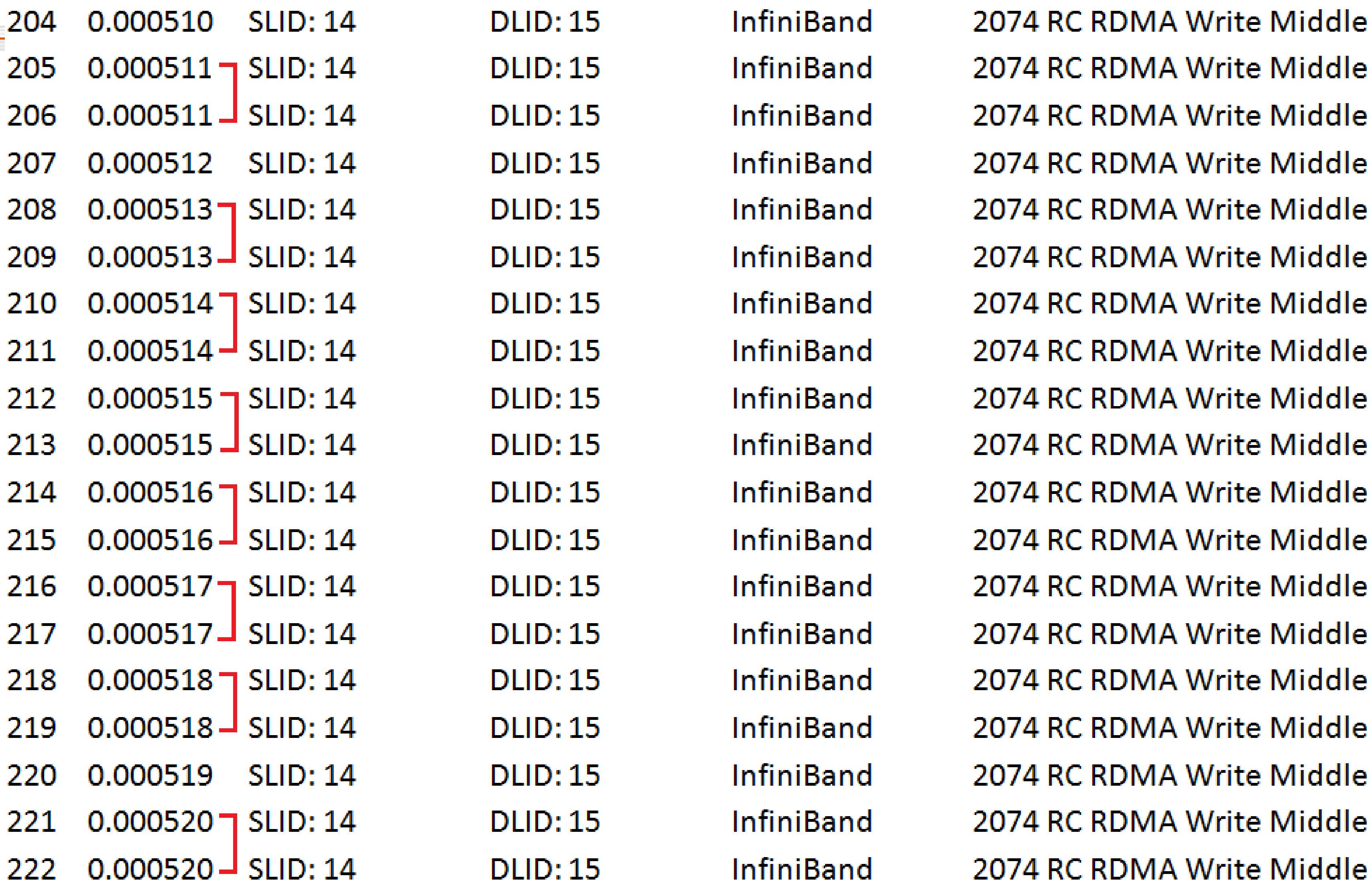
```
⊞ Frame 51: 4122 bytes on wire (32976 bits), 4122 bytes captured (32976 bits) on interface 0
⊞ Extensible Record Format
⊟ InfiniBand
  ⊟ Local Route Header
      0000 .... = Virtual Lane: 0x00
      .... 0000 = Link Version: 0
      0000 .... = Service Level: 0
      .... 00.. = Reserved (2 bits): 0
      .... ..10 = Link Next Header: 0x02
      Destination Local ID: 3
      0000 0... .... .... = Reserved (5 bits): 0
      .... .100 0000 0110 = Packet Length: 1030
      Source Local ID: 7
  ⊟ Base Transport Header
      Opcode: 7
      0... .... = Solicited Event: False
      .1.. .... = MigReq: True
      ..00 .... = Pad Count: 0
```

```
0000  00 02 00 03 04 06 00 07  07 40 ff ff 00 00 0c 32   ........ .@.....2
0010  00 69 82 df 3a 3b 3c 3d  3e 3f 40 41 42 43 44 45   .i..:;<= >?@ABCDE
0020  46 47 48 49 4a 4b 4c 4d  4e 4f 50 51 52 53 54 55   FGHIJKLM NOPQRSTU
0030  56 57 58 59 5a 5b 5c 5d  5e 5f 60 61 62 63 64 65   VWXYZ[\] ^_`abcde
0040  66 67 68 69 6a 6b 6c 6d  6e 6f 70 71 72 73 74 75   fghijklm nopqrstu
0050  76 77 78 79 7a 7b 7c 7d  7e 20 21 22 23 24 25 26   vwxyz{|} ~ !"#$%&
0060  27 28 29 2a 2b 2c 2d 2e  2f 30 31 32 33 34 35 36   '()*+,-. /0123456
```

ibdump

- ibdump limitations
  - Cannot capture Flow Control Packets (FCP)
  - Packets may get lost if the data rate is high, e.g. FDR (56Gbits/s)
  - Works only on Mellanox HCAs
  - Doesn’t work between switches because it is software running on nodes
  - Max capture size depends on the available host RAM or Disk space
  - Inaccurate packet timestamps (in microsecond) (show this next)

Inaccurate microsecond timestamps in ibdump



```
204   0.000510   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
205   0.000511   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
206   0.000511   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
207   0.000512   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
208   0.000513   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
209   0.000513   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
210   0.000514   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
211   0.000514   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
212   0.000515   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
213   0.000515   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
214   0.000516   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
215   0.000516   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
216   0.000517   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
217   0.000517   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
218   0.000518   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
219   0.000518   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
220   0.000519   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
221   0.000520   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
222   0.000520   SLID: 14     DLID: 15     InfiniBand     2074 RC RDMA Write Middle
```

2. How to capture IB packets

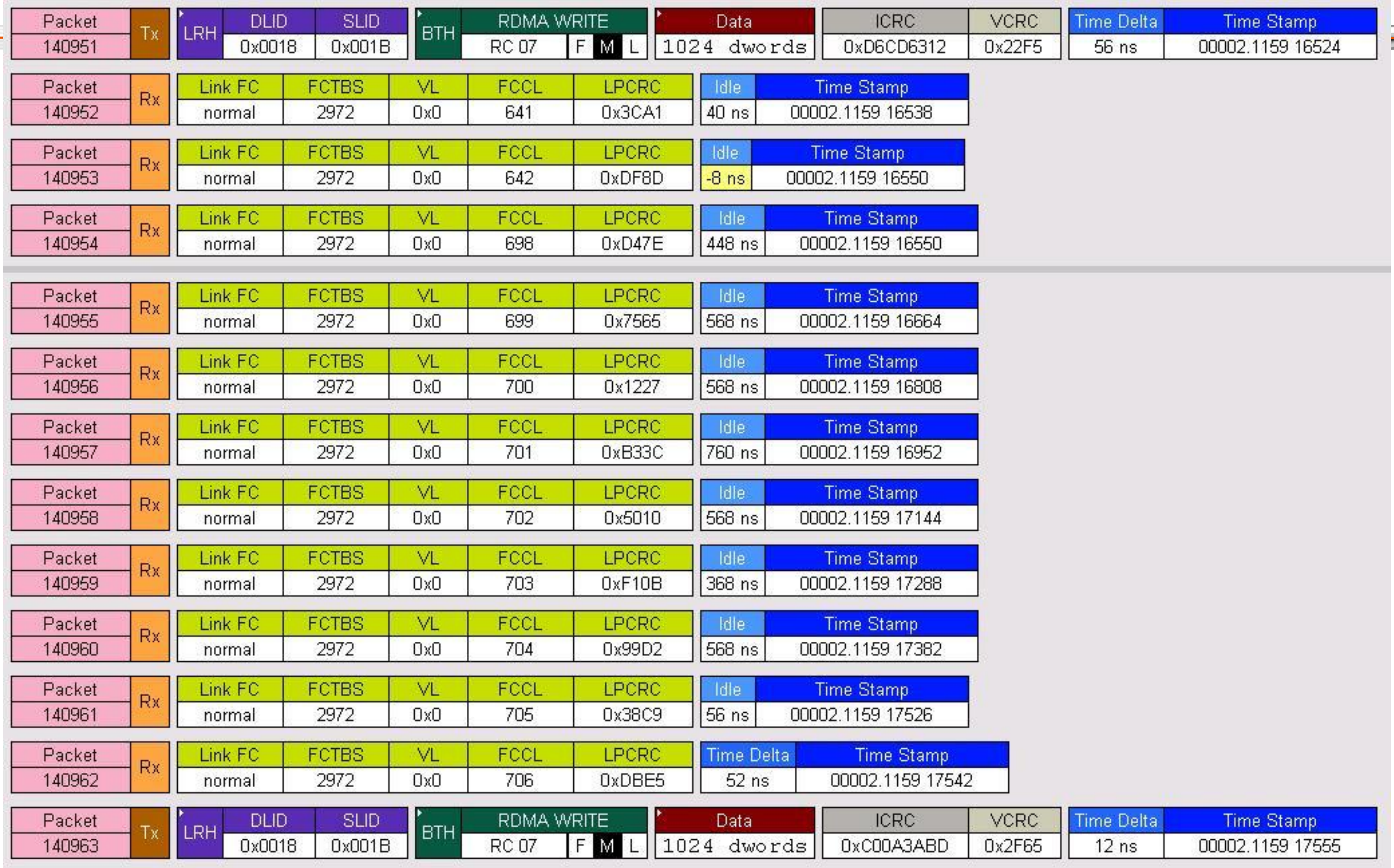
- CatC analyzer features
  - Hardware analyzer from LeCroy

    https://www.teledynelecroy.com
  - Must be physically placed into an IB link between two IB ports
  - Dumps an .ibt file which can be loaded by its IBTracer software
  - Works only for SDR (8Gbits/s) data rate
  - Works for any type of IB HCAs and switches
  - Accurate packet timestamps (in nanosecond)
  - Captures ALL packets on the link, including Flow Control Packets (FCP)

CatC analyzer

- Captures packets passing through it in both directions

CatC analyzer Capture



Packet			DLID	SLID		RDMA WRITE	Data	ICRC	VCRC	Time Delta	Time Stamp
140951	Tx	LRH	0x0018	0x001B	BTH	RC 07 F M L	1024 dwords	0xD6CD6312	0x22F5	56 ns	00002.1159 16524

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
140952	Rx	normal	2972	0x0	641	0x3CA1	40 ns	00002.1159 16538
140953	Rx	normal	2972	0x0	642	0xDF8D	-8 ns	00002.1159 16550
140954	Rx	normal	2972	0x0	698	0xD47E	448 ns	00002.1159 16550
140955	Rx	normal	2972	0x0	699	0x7565	568 ns	00002.1159 16664
140956	Rx	normal	2972	0x0	700	0x1227	568 ns	00002.1159 16808
140957	Rx	normal	2972	0x0	701	0xB33C	760 ns	00002.1159 16952
140958	Rx	normal	2972	0x0	702	0x5010	568 ns	00002.1159 17144
140959	Rx	normal	2972	0x0	703	0xF10B	368 ns	00002.1159 17288
140960	Rx	normal	2972	0x0	704	0x99D2	568 ns	00002.1159 17382
140961	Rx	normal	2972	0x0	705	0x38C9	56 ns	00002.1159 17526

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Time Delta	Time Stamp
140962	Rx	normal	2972	0x0	706	0xDBE5	52 ns	00002.1159 17542

Packet			DLID	SLID		RDMA WRITE	Data	ICRC	VCRC	Time Delta	Time Stamp
140963	Tx	LRH	0x0018	0x001B	BTH	RC 07 F M L	1024 dwords	0xC00A3ABD	0x2F65	12 ns	00002.1159 17555

CatC analyzer

- CatC analyzer limitations
  - Only works for SDR (8Gbits/s) data rate
  - 2GB recording capacity
  - Doesn’t dump in .pcap format, so its capture file cannot use Wireshark

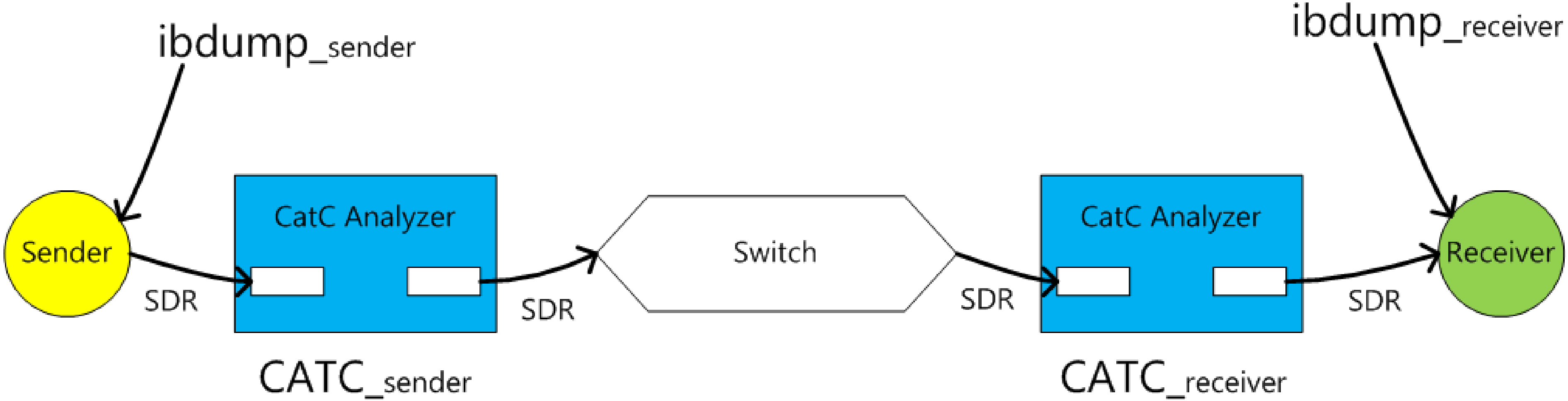
3. Comparison between ibdump & CatC analyzer captures

First Experiment

One data source is sending 128Mi bytes (MTU = 2k, 65536 packets), by using RDMA_WRITE, to the receiver via a MLNX SX6036 switch.

Because there is no competing flow, therefore, there should be no congestion on the link.

ibdump on both sides are running at the same time



3. Comparison between ibdump & CatC analyzer captures

First Experiment

Transferring data packets on a SDR (8Gbits/s) link with no congestion,

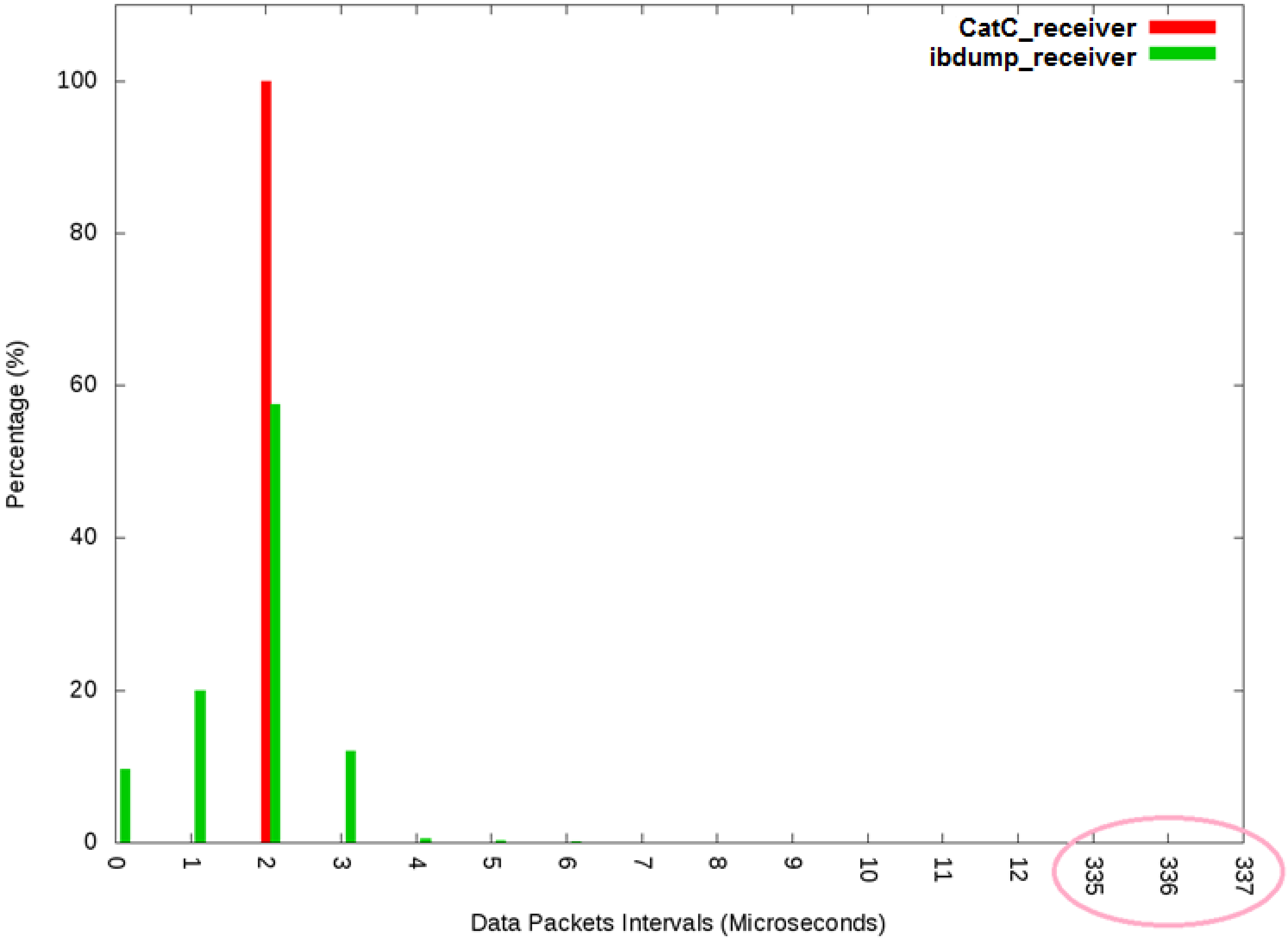
if each data packet has 2048 bytes payload (MTU is 2k),

The inter-packet time should be around:

2048 bytes * 8 / (8Gbits/s) = 2 us

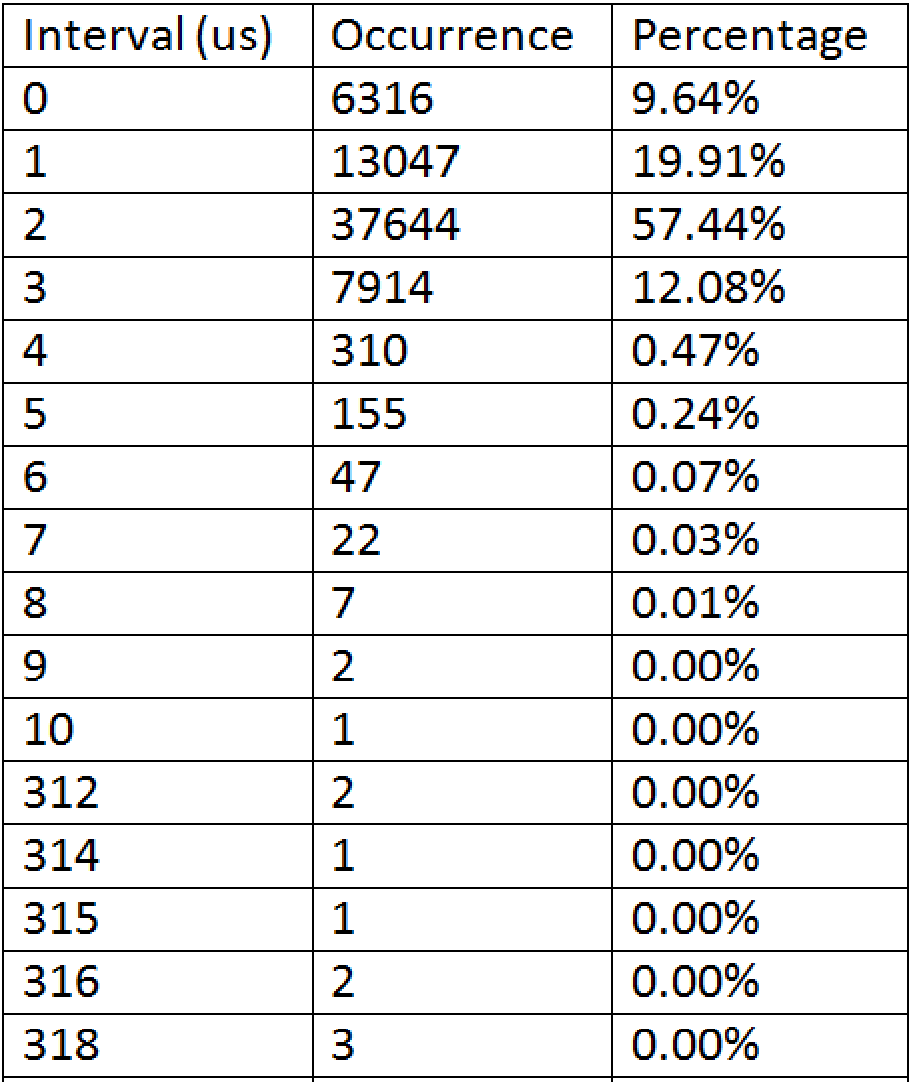
3. Comparison between ibdump & CatC captures on the receive side

First Experiment

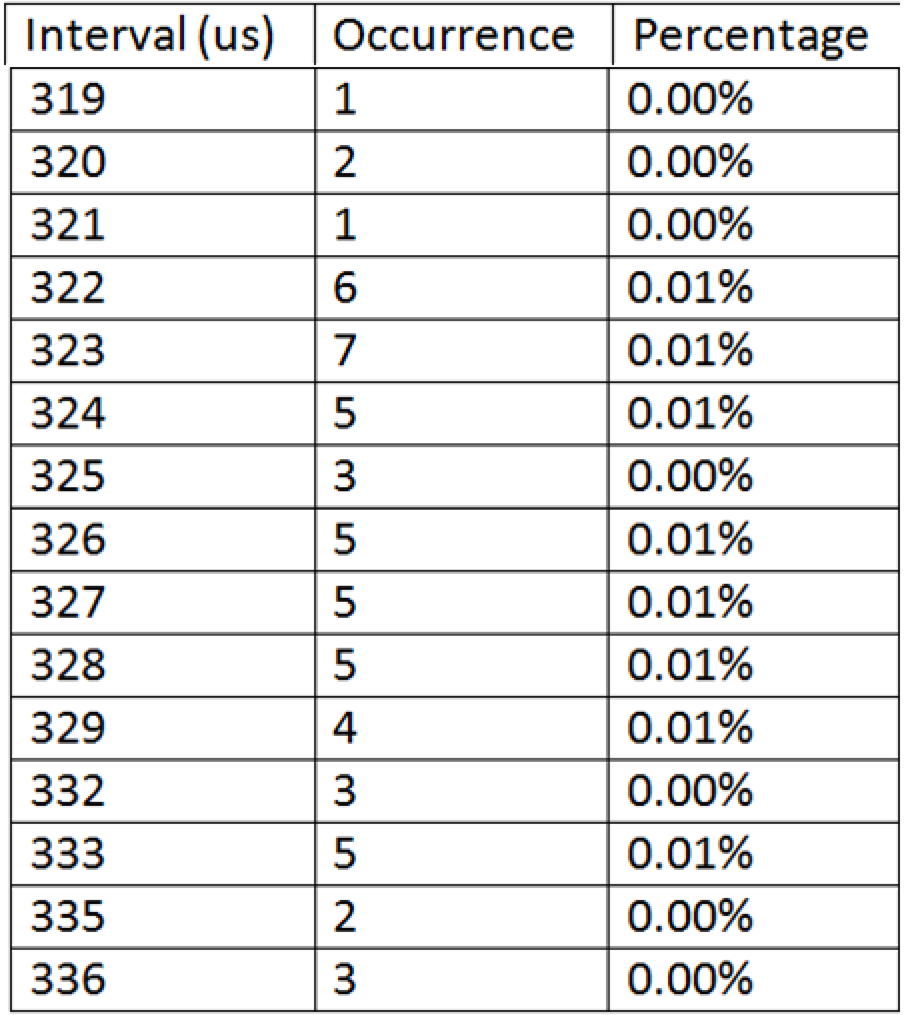


3. ibdump_receiver raw data

First Experiment



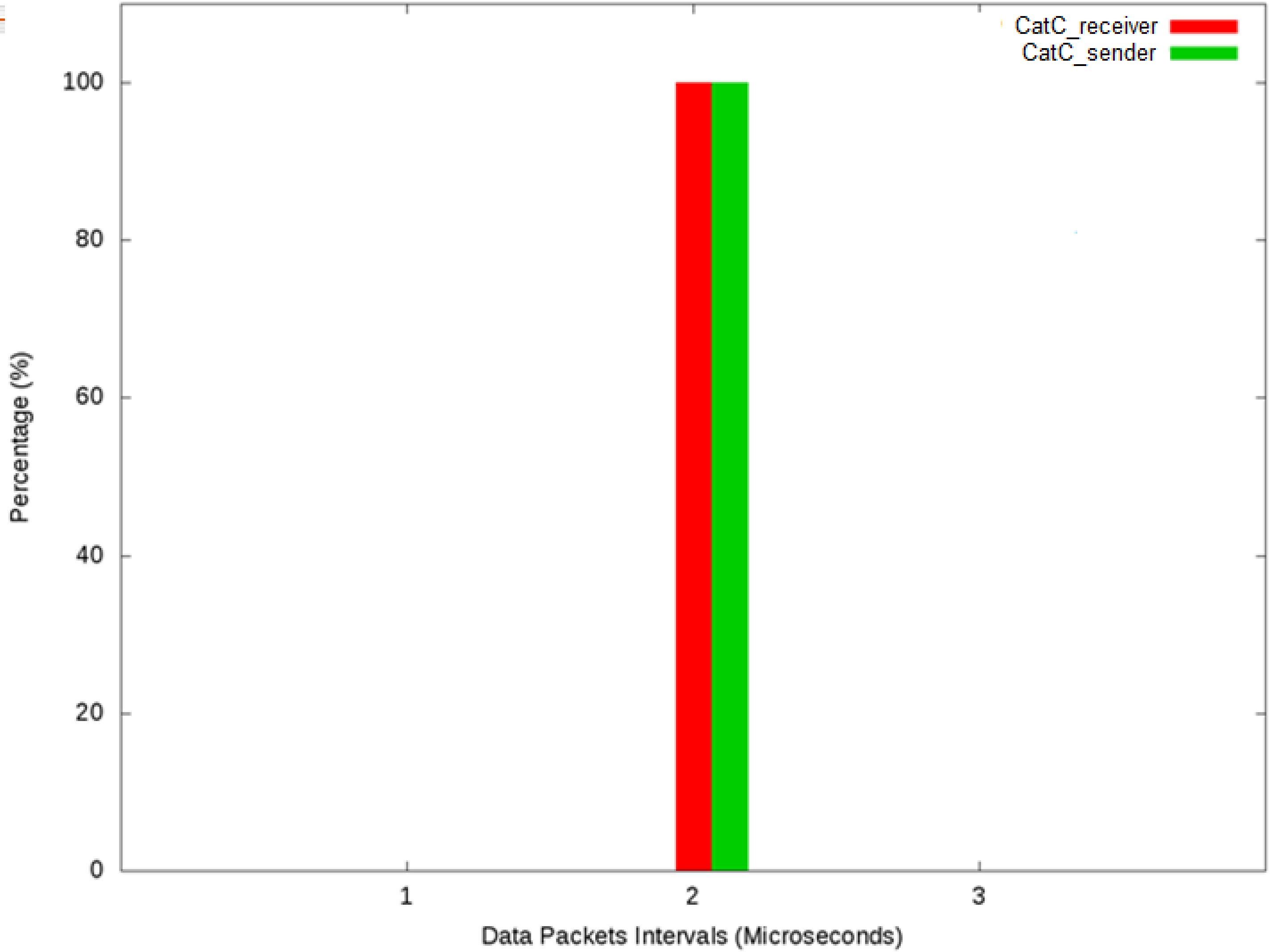
Interval (us)	Occurrence	Percentage
0	6316	9.64%
1	13047	19.91%
2	37644	57.44%
3	7914	12.08%
4	310	0.47%
5	155	0.24%
6	47	0.07%
7	22	0.03%
8	7	0.01%
9	2	0.00%
10	1	0.00%
312	2	0.00%
314	1	0.00%
315	1	0.00%
316	2	0.00%
318	3	0.00%



Interval (us)	Occurrence	Percentage
319	1	0.00%
320	2	0.00%
321	1	0.00%
322	6	0.01%
323	7	0.01%
324	5	0.01%
325	3	0.00%
326	5	0.01%
327	5	0.01%
328	5	0.01%
329	4	0.01%
332	3	0.00%
333	5	0.01%
335	2	0.00%
336	3	0.00%

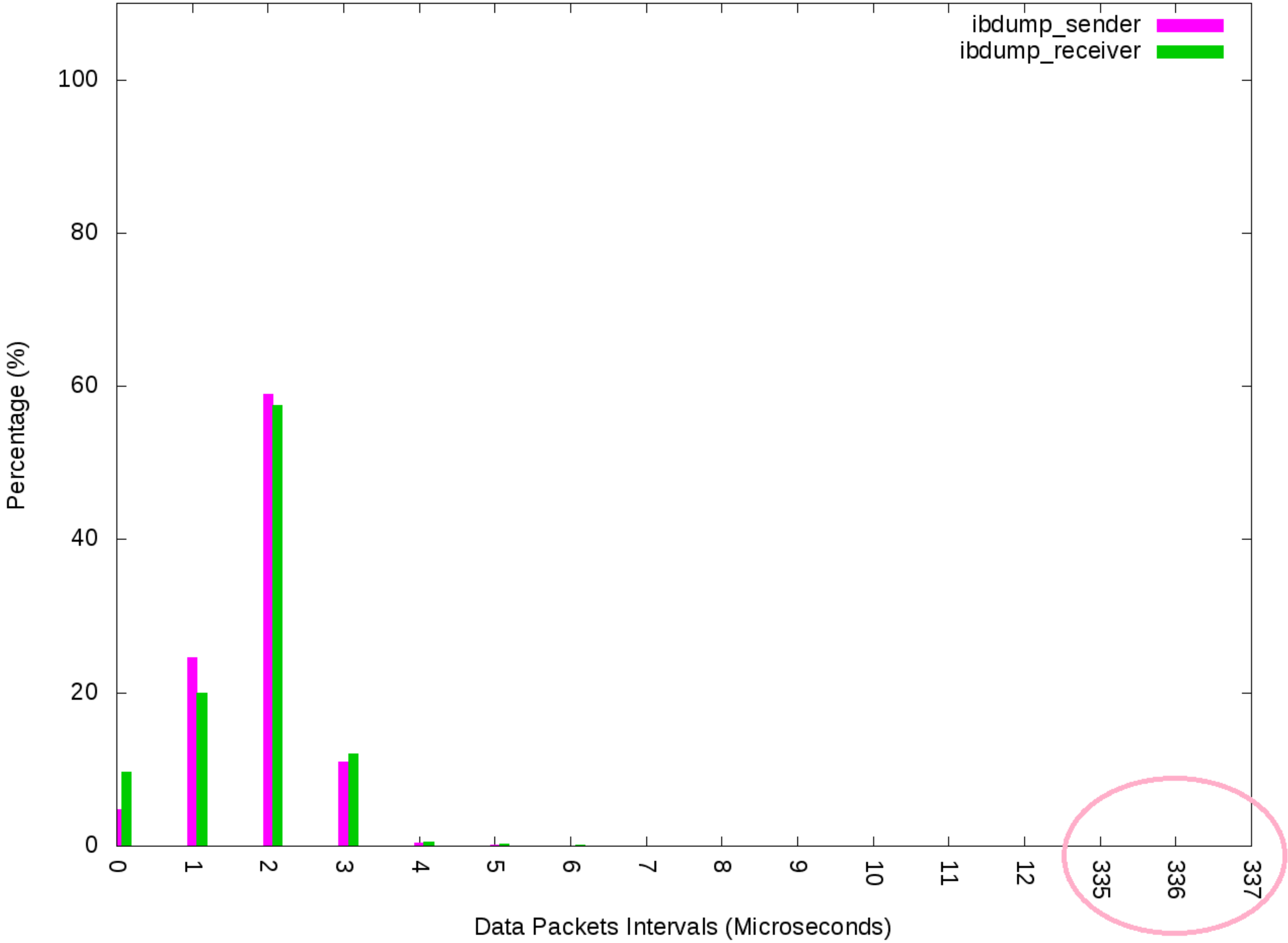
Comparison of CatC analyzer captures on both sides

First Experiment



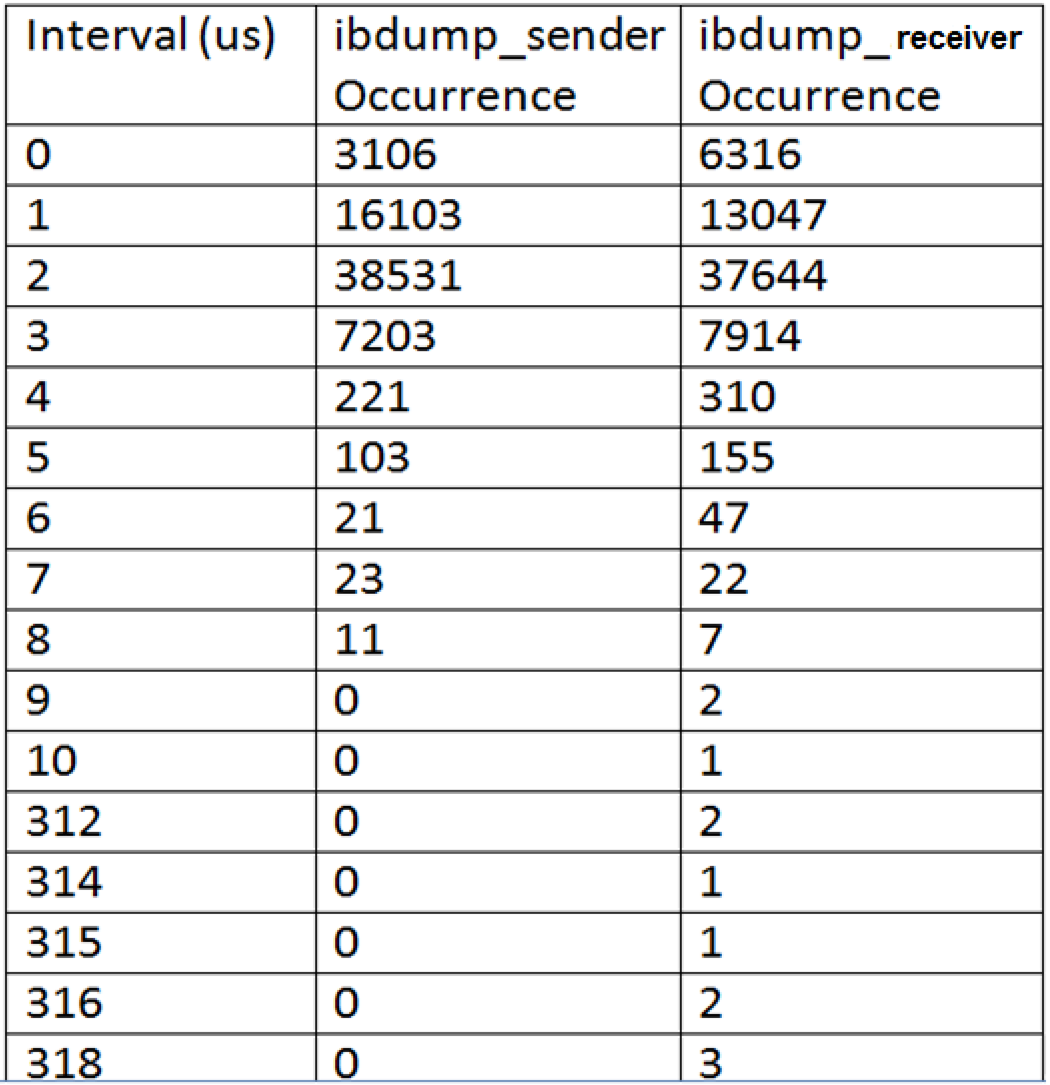
Comparison of ibdump captures on both sides

First Experiment

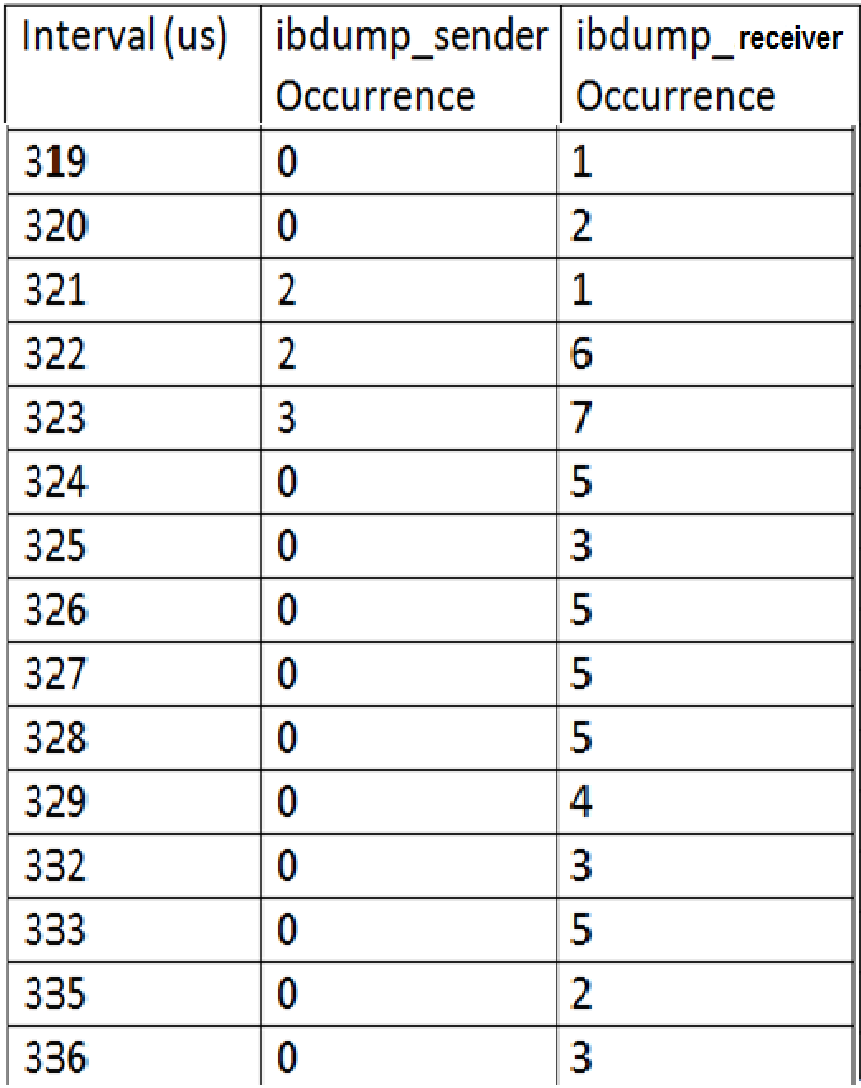


ibdump_sender and ibdump_receiver raw data

First Experiment



Interval (us)	ibdump_sender Occurrence	ibdump_receiver Occurrence
0	3106	6316
1	16103	13047
2	38531	37644
3	7203	7914
4	221	310
5	103	155
6	21	47
7	23	22
8	11	7
9	0	2
10	0	1
312	0	2
314	0	1
315	0	1
316	0	2
318	0	3



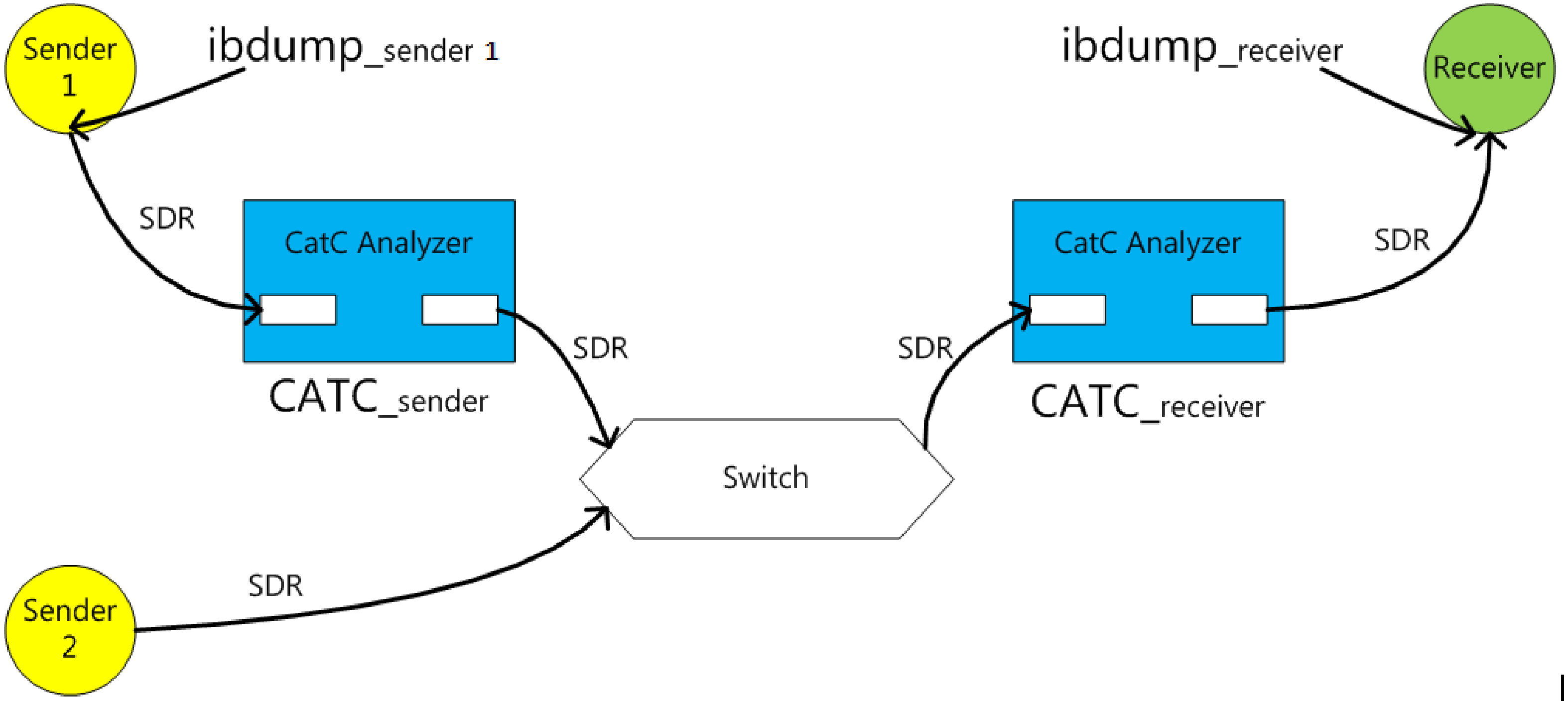
Interval (us)	ibdump_sender Occurrence	ibdump_receiver Occurrence
319	0	1
320	0	2
321	2	1
322	2	6
323	3	7
324	0	5
325	0	3
326	0	5
327	0	5
328	0	5
329	0	4
332	0	3
333	0	5
335	0	2
336	0	3

3. Comparison between ibdump & CatC analyzer captures

Second Experiment

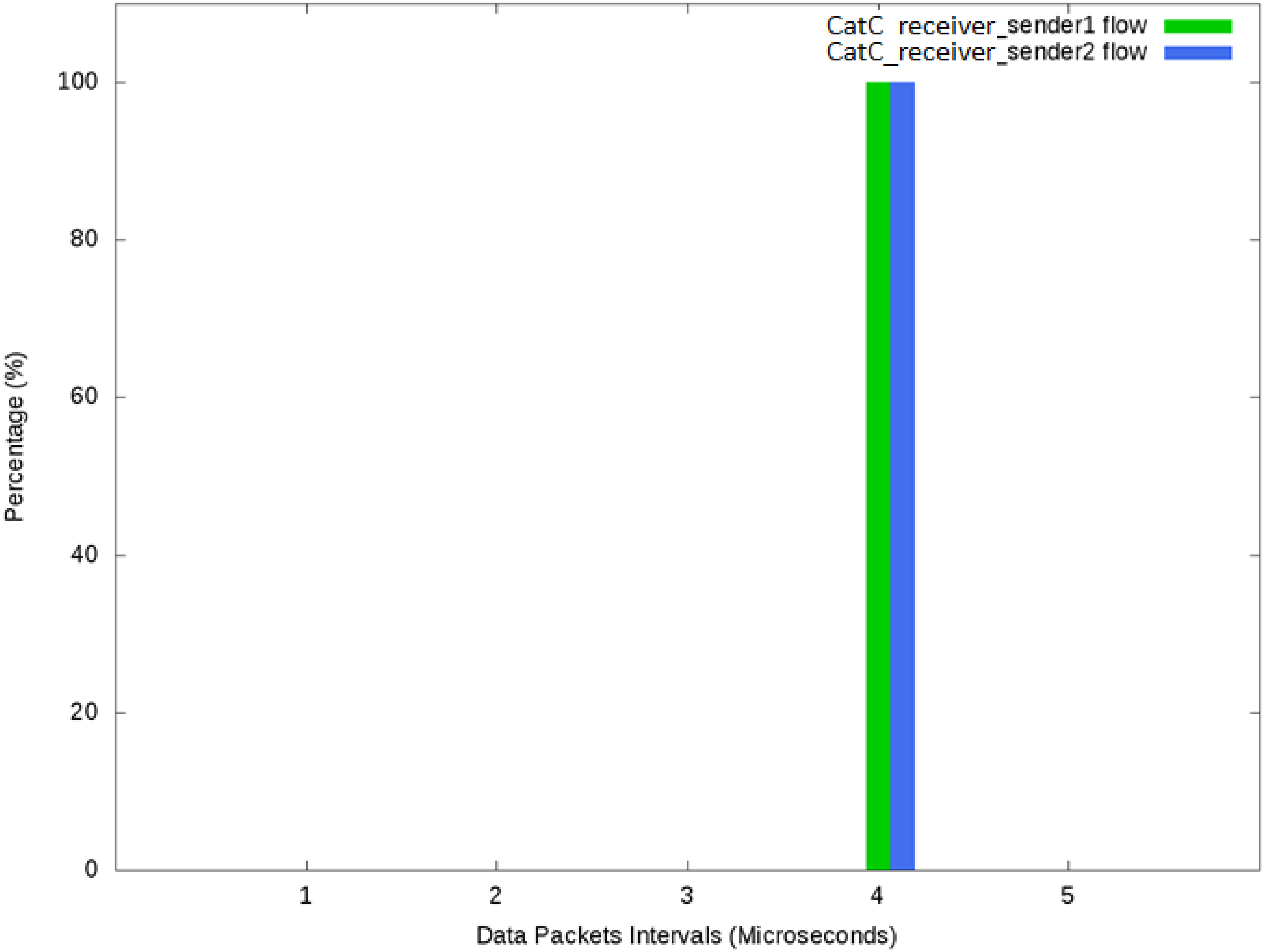
Two data sources, each is sending 128Mi bytes, by using RDMA_WRITE, to the single receiver via a MLNX SX6036 switch.

The expected inter-packet interval from the same source should be 4 us



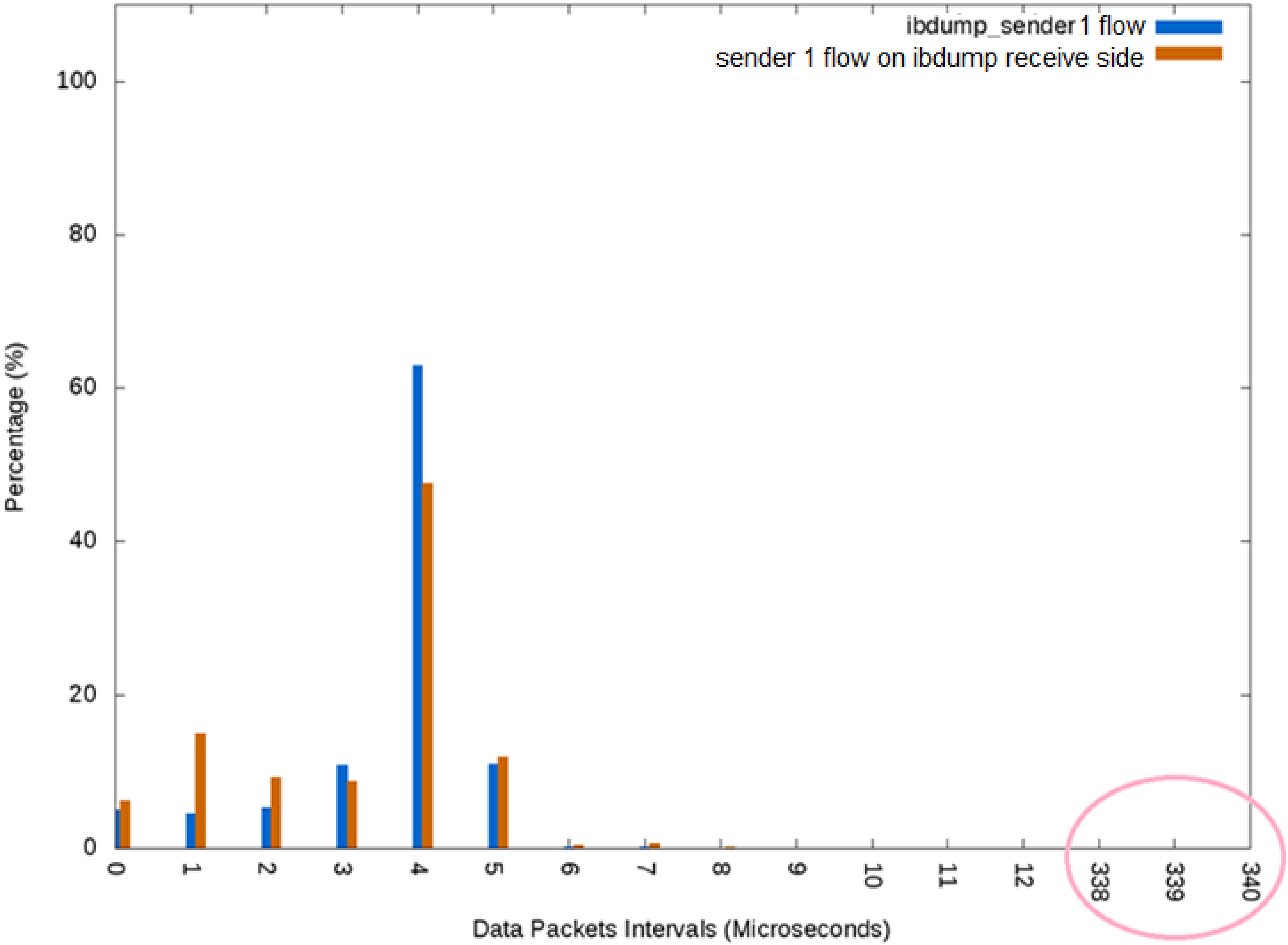
Comparison of two sender flows on CatC receive side

Second Experiment



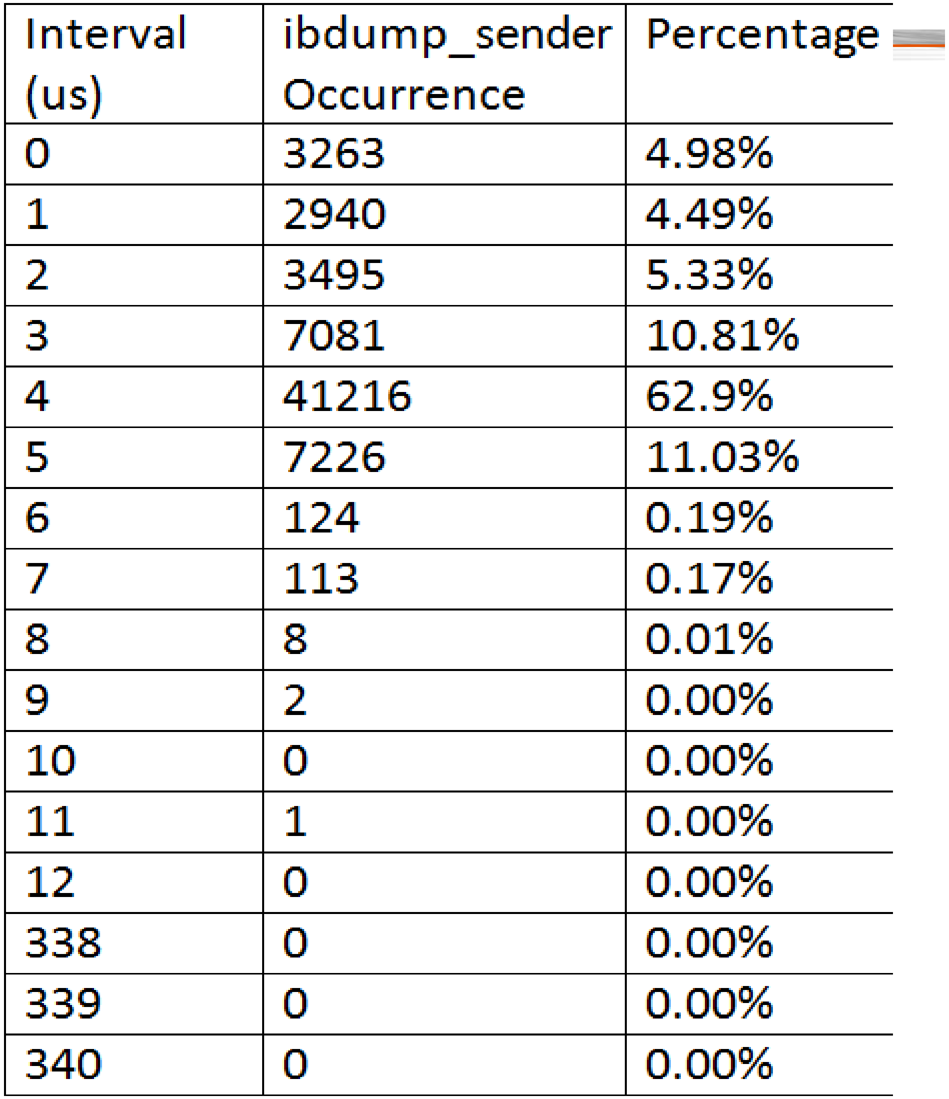
Comparison of ibdump sender 1 flow on both sides

Second Experiment

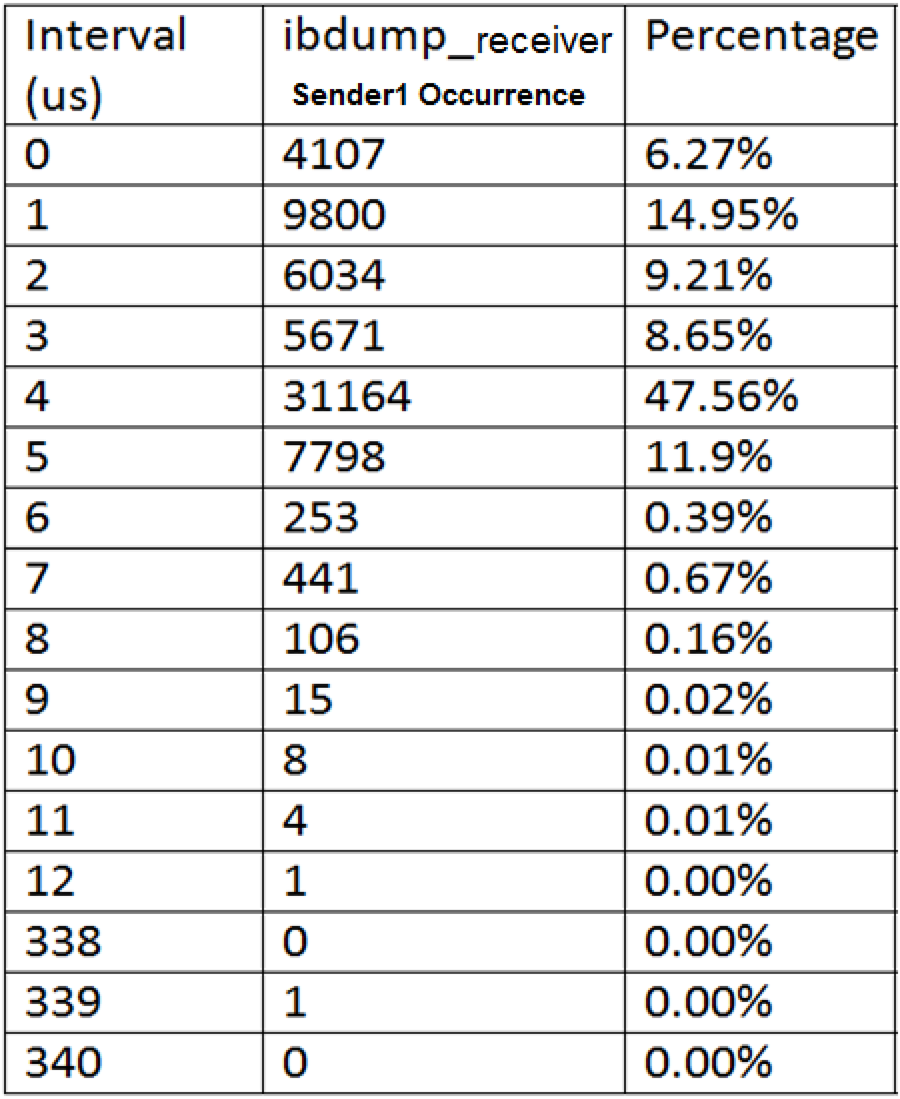


ibdump sender 1 flow raw data on both sides

Second Experiment



Interval (us)	ibdump_sender Occurrence	Percentage
0	3263	4.98%
1	2940	4.49%
2	3495	5.33%
3	7081	10.81%
4	41216	62.9%
5	7226	11.03%
6	124	0.19%
7	113	0.17%
8	8	0.01%
9	2	0.00%
10	0	0.00%
11	1	0.00%
12	0	0.00%
338	0	0.00%
339	0	0.00%
340	0	0.00%



Interval (us)	ibdump_receiver Sender1 Occurrence	Percentage
0	4107	6.27%
1	9800	14.95%
2	6034	9.21%
3	5671	8.65%
4	31164	47.56%
5	7798	11.9%
6	253	0.39%
7	441	0.67%
8	106	0.16%
9	15	0.02%
10	8	0.01%
11	4	0.01%
12	1	0.00%
338	0	0.00%
339	1	0.00%
340	0	0.00%

4. Our use of the tools to analyze packets

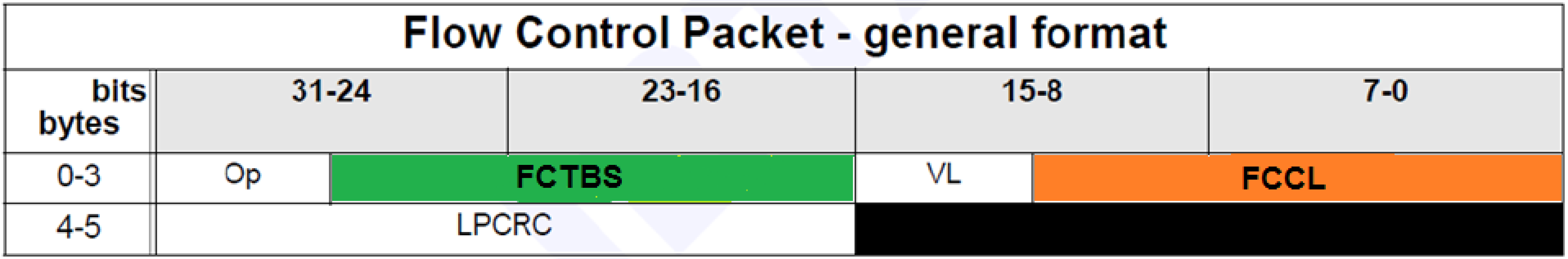
- 4.1 Flow Control mechanism
- 4.2 Study of the switch buffer size
- 4.3 Study of the tick value

4.1 Flow Control Mechanism

- InfiniBand – Link Layer Flow Control (FC) mechanism
- IB sender will NOT send data packets unless it knows for sure that the other side of the physical link has enough buffer to hold the data
- Flow Control Packets (FCPs) are used to report the available buffer space
- Only CatC analyzer can capture FCPs

4.1 Flow Control Mechanism

- FCP format



Flow Control Packet - general format

bits bytes	31-24		23-16		15-8		7-0	
0-3	Op	FCTBS			VL	FCCL		
4-5	LPCRC							

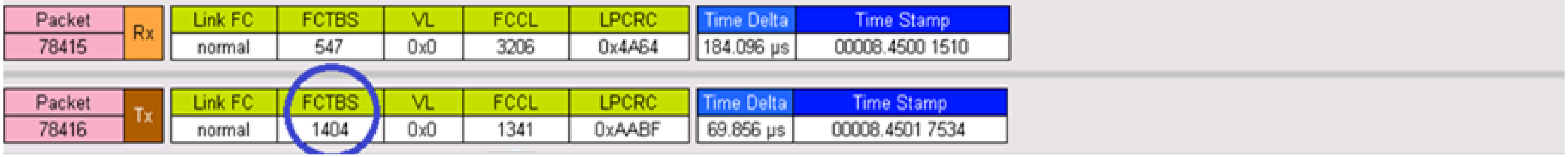
- If A sends a FCP to B, then
  - **FCTBS**: total blocks A has sent to B since link initialization
  - **FCCL**: the **sum** of the total blocks A has received from B, plus the available buffer space in A's receive buffer
  - Both numbers are increasing monotonically, modulo 4096
  - One block is 64 bytes of buffer space

4.1 Flow Control Mechanism

- Experiment:
  - A sender is sending 128Mi bytes of data to a receiver, using RDMA_WRITE
  - MTU = 2k, 65536 data packets
  - Each packet is at least 2048 + 8 + 12 + 6 = 2074 bytes.
  - Each packet occupies $\left\lceil \frac{2074}{64} \right\rceil$ = 33 FC blocks

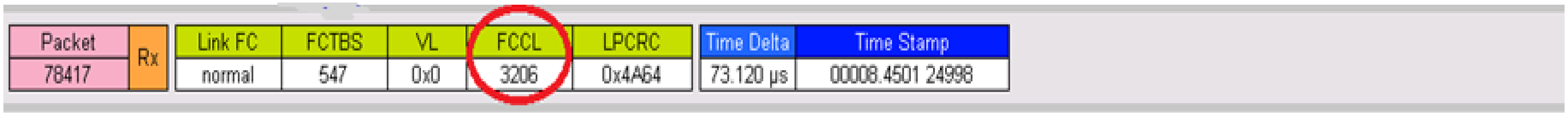
4.1 Flow Control Mechanism

- Starting FCCL/FCTBS before A (Tx) sends data packets to B (Rx)



Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Time Delta	Time Stamp
78415	Rx	normal	547	0x0	3206	0x4A64	184.096 µs	00008.4500 1510
78416	Tx	normal	1404	0x0	1341	0xAABF	69.856 µs	00008.4501 7534

A has sent 1404 blocks to B

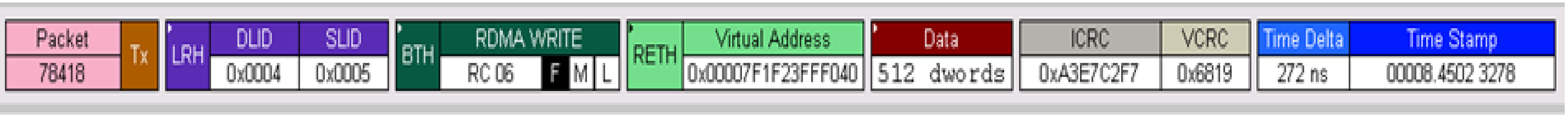


Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Time Delta	Time Stamp
78417	Rx	normal	547	0x0	3206	0x4A64	73.120 µs	00008.4501 24998

A receives a FCP from B, in which the FCCL value is 3206

3206 = total blocks B has received from A + the available receive buffer space in B

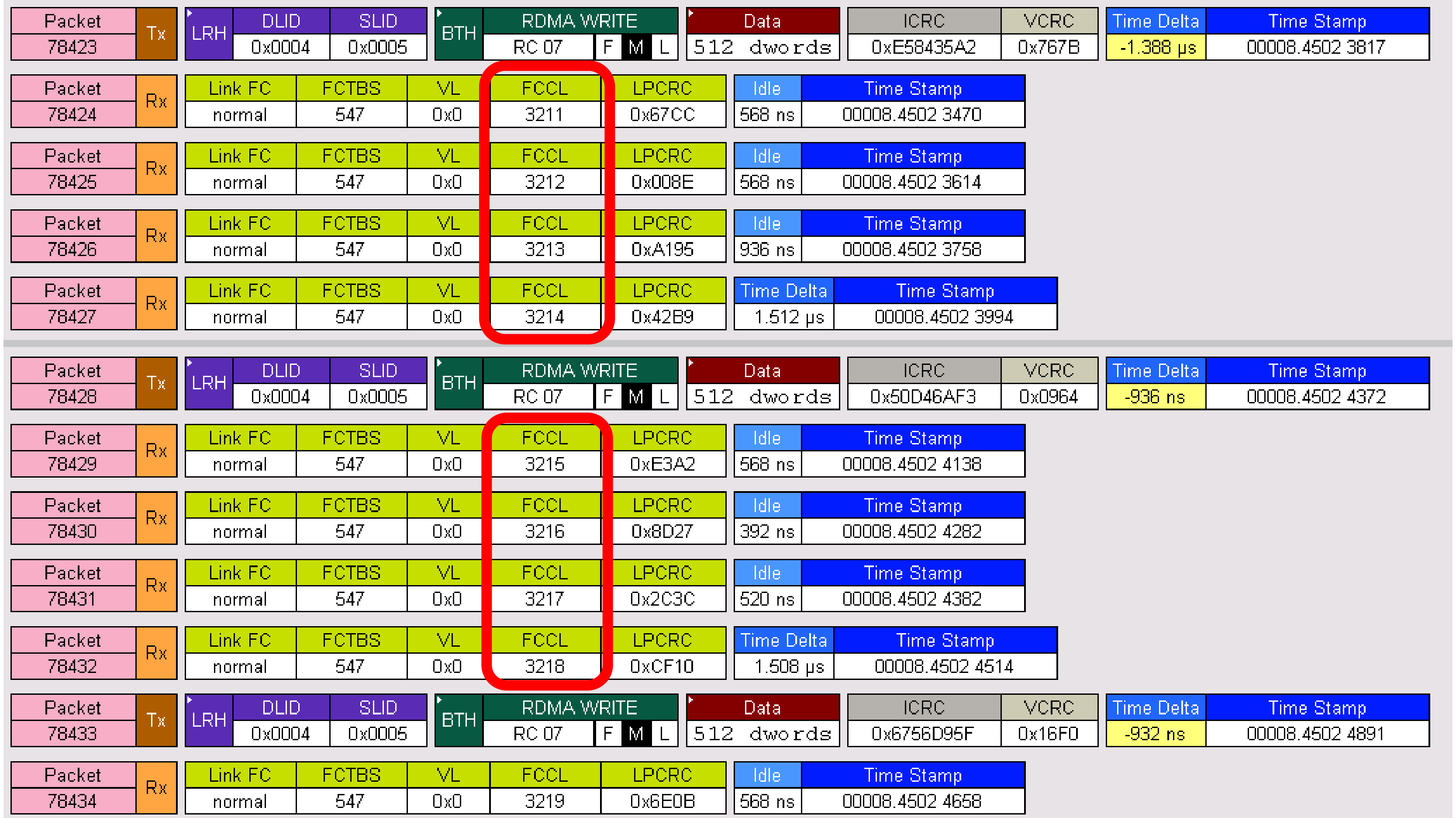
3206 – 1404 >> 33, based on this calculation, A is able to send a data packet



Packet		LRH	DLID	SLID	BTH	RDMA WRITE				RETH	Virtual Address	Data	ICRC	VCRC	Time Delta	Time Stamp
78418	Tx		0x0004	0x0005		RC 06	F	M	L		0x00007F1F23FFF040	512 dwords	0xA3E7C2F7	0x6819	272 ns	00008.4502 3278

4.1 Flow Control Mechanism

- FCCL value update -> means one or more blocks are released in B's receive buffer



Packet		LRH	DLID	SLID	BTH	RDMA WRITE				Data	ICRC	VCRC	Time Delta	Time Stamp
78423	Tx		0x0004	0x0005		RC 07	F	M	L	512 dwords	0xE58435A2	0x767B	-1.388 μs	00008.4502 3817

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78424	Rx	normal	547	0x0	3211	0x67CC	568 ns	00008.4502 3470
78425	Rx	normal	547	0x0	3212	0x008E	568 ns	00008.4502 3614
78426	Rx	normal	547	0x0	3213	0xA195	936 ns	00008.4502 3758

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Time Delta	Time Stamp
78427	Rx	normal	547	0x0	3214	0x42B9	1.512 μs	00008.4502 3994

Packet		LRH	DLID	SLID	BTH	RDMA WRITE				Data	ICRC	VCRC	Time Delta	Time Stamp
78428	Tx		0x0004	0x0005		RC 07	F	M	L	512 dwords	0x50D46AF3	0x0964	-936 ns	00008.4502 4372

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78429	Rx	normal	547	0x0	3215	0xE3A2	568 ns	00008.4502 4138
78430	Rx	normal	547	0x0	3216	0x8D27	392 ns	00008.4502 4282
78431	Rx	normal	547	0x0	3217	0x2C3C	520 ns	00008.4502 4382

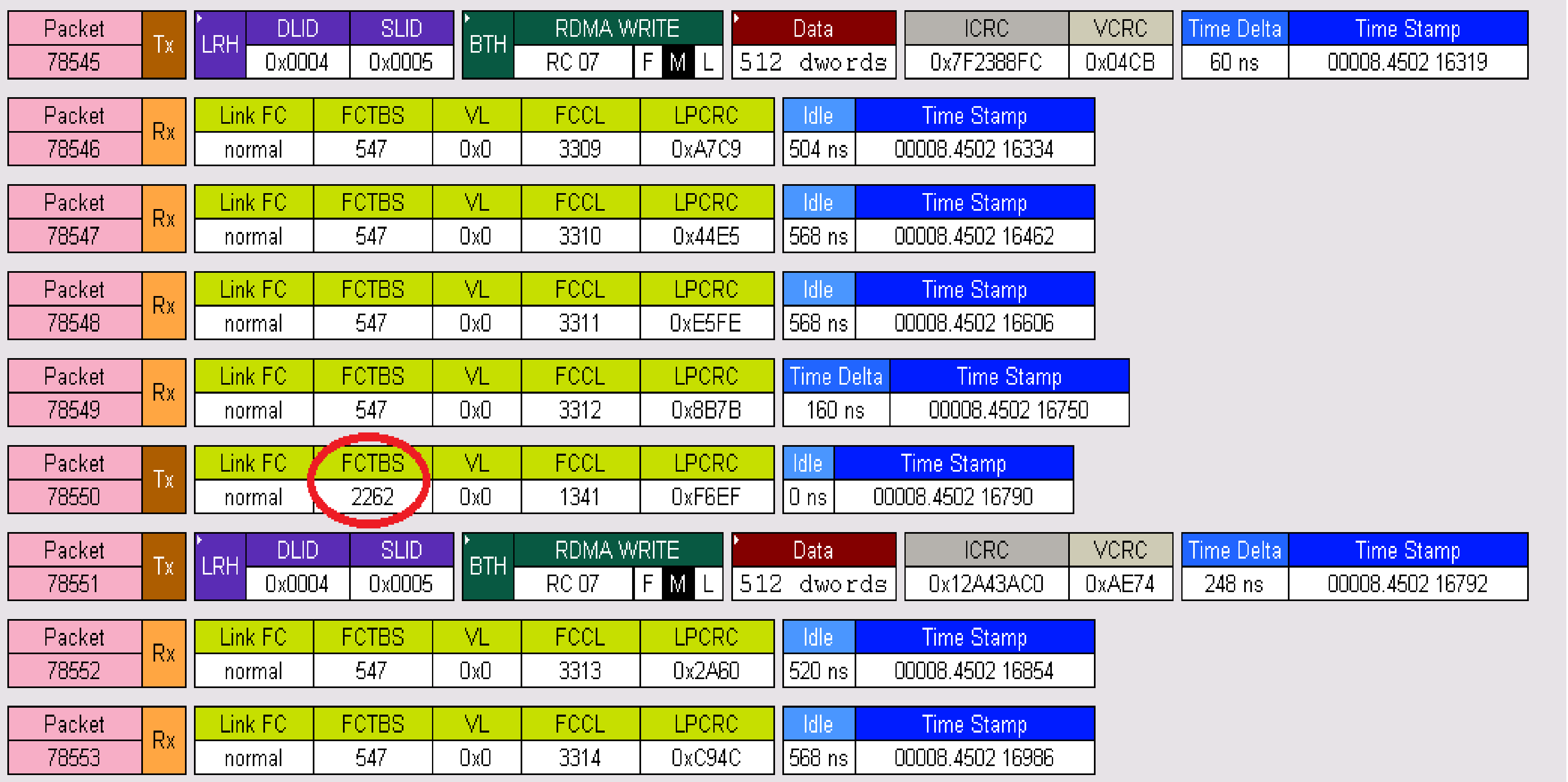
Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Time Delta	Time Stamp
78432	Rx	normal	547	0x0	3218	0xCF10	1.508 μs	00008.4502 4514

Packet		LRH	DLID	SLID	BTH	RDMA WRITE				Data	ICRC	VCRC	Time Delta	Time Stamp
78433	Tx		0x0004	0x0005		RC 07	F	M	L	512 dwords	0x6756D95F	0x16F0	-932 ns	00008.4502 4891

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78434	Rx	normal	547	0x0	3219	0x6E0B	568 ns	00008.4502 4658

4.1 Flow Control Mechanism

- FCTBS value update



Packet		LRH	DLID	SLID	BTH	RDMA WRITE				Data	ICRC	VCRC	Time Delta	Time Stamp
78545	Tx		0x0004	0x0005		RC 07	F	M	L	512 dwords	0x7F2388FC	0x04CB	60 ns	00008.4502 16319

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78546	Rx	normal	547	0x0	3309	0xA7C9	504 ns	00008.4502 16334

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78547	Rx	normal	547	0x0	3310	0x44E5	568 ns	00008.4502 16462

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78548	Rx	normal	547	0x0	3311	0xE5FE	568 ns	00008.4502 16606

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Time Delta	Time Stamp
78549	Rx	normal	547	0x0	3312	0x8B7B	160 ns	00008.4502 16750

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78550	Tx	normal	2262	0x0	1341	0xF6EF	0 ns	00008.4502 16790

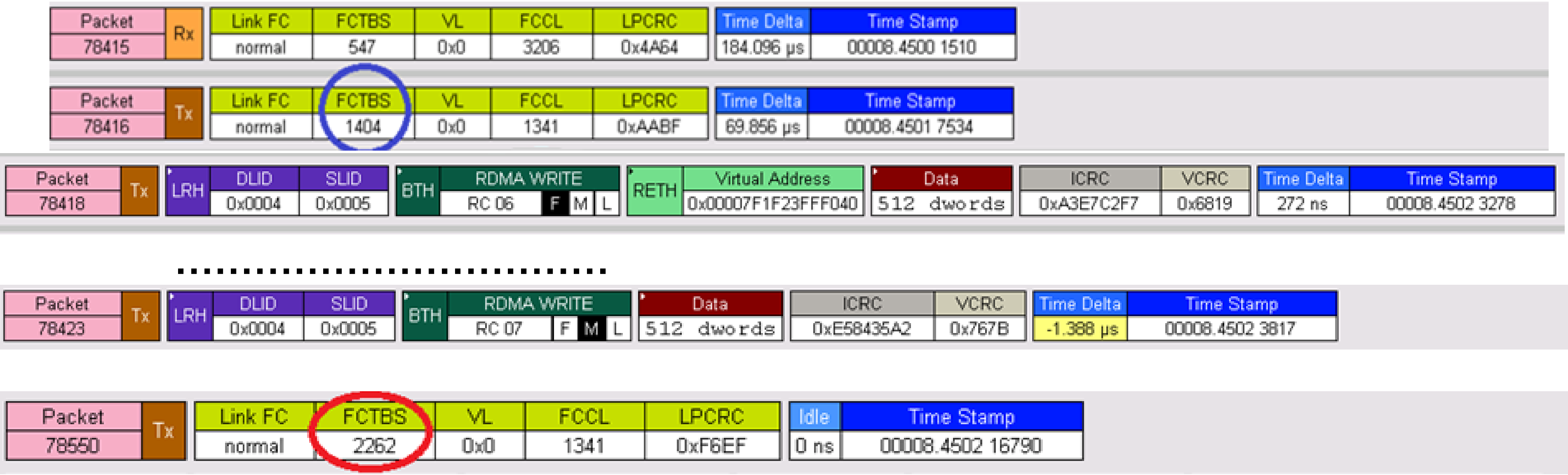
Packet		LRH	DLID	SLID	BTH	RDMA WRITE				Data	ICRC	VCRC	Time Delta	Time Stamp
78551	Tx		0x0004	0x0005		RC 07	F	M	L	512 dwords	0x12A43AC0	0xAE74	248 ns	00008.4502 16792

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78552	Rx	normal	547	0x0	3313	0x2A60	520 ns	00008.4502 16854

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78553	Rx	normal	547	0x0	3314	0xC94C	568 ns	00008.4502 16986

4.1 Flow Control Mechanism

- Before A sends data packets to B, the starting FCTBS value is 1404



Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Time Delta	Time Stamp
78415	Rx	normal	547	0x0	3206	0x4A64	184.096 µs	00008.4500 1510

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Time Delta	Time Stamp
78416	Tx	normal	1404	0x0	1341	0xAABF	69.856 µs	00008.4501 7534

Packet		LRH	DLID	SLID	BTH	RDMA WRITE	RETH	Virtual Address	Data	ICRC	VCRC	Time Delta	Time Stamp
78418	Tx		0x0004	0x0005		RC 06 F M L		0x00007F1F23FFF040	512 dwords	0xA3E7C2F7	0x6819	272 ns	00008.4502 3278

....

Packet		LRH	DLID	SLID	BTH	RDMA WRITE	Data	ICRC	VCRC	Time Delta	Time Stamp
78423	Tx		0x0004	0x0005		RC 07 F M L	512 dwords	0xE58435A2	0x767B	-1.388 µs	00008.4502 3817

Packet		Link FC	FCTBS	VL	FCCL	LPCRC	Idle	Time Stamp
78550	Tx	normal	2262	0x0	1341	0xF6EF	0 ns	00008.4502 16790

- The latest FCTBS value is 2262

- (2262 - 1404) / 33 = 26 data packets have been sent from A to B

4.2 Study of switch buffer size

- Object:

  MLNX SX6036 FDR switch

  Use the CatC analyzer to determine the switch buffer size

  Assumption:

  1. input-queued switch
  2. shared buffer per port, divided by the available Virtual Lanes (VLs)

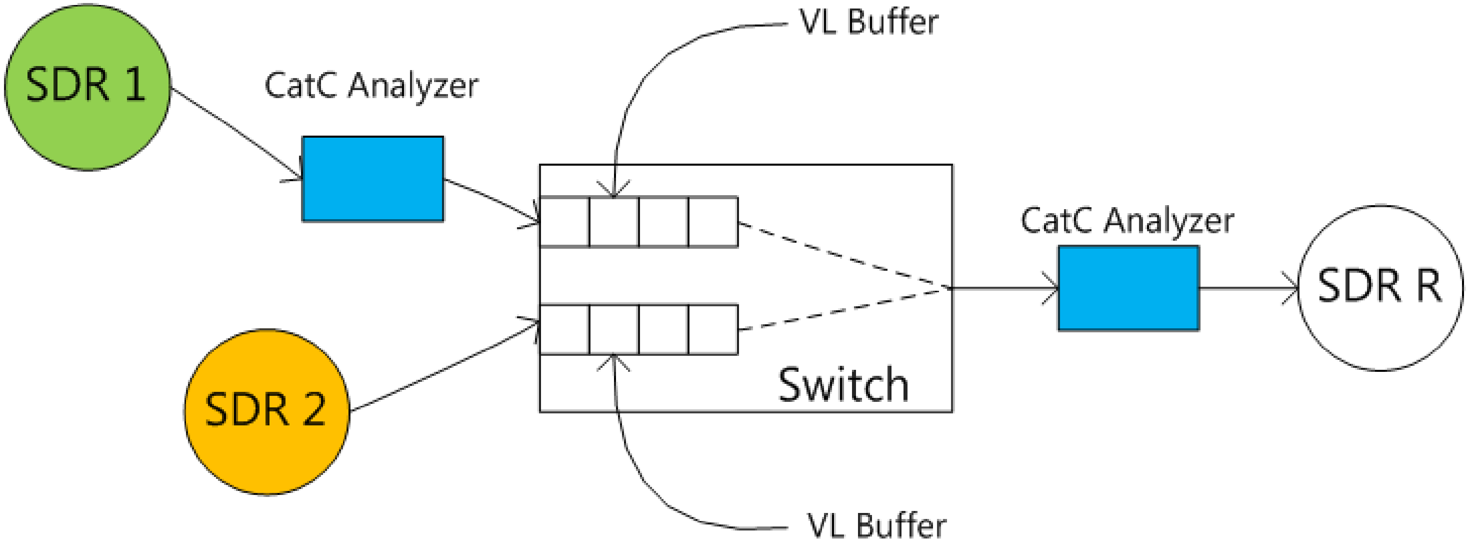
4.2 Study of switch buffer size

The buffer size is an indicator of the latency a program may experience

SDR 1 and SDR 2, two senders are sending data to a SDR receiver

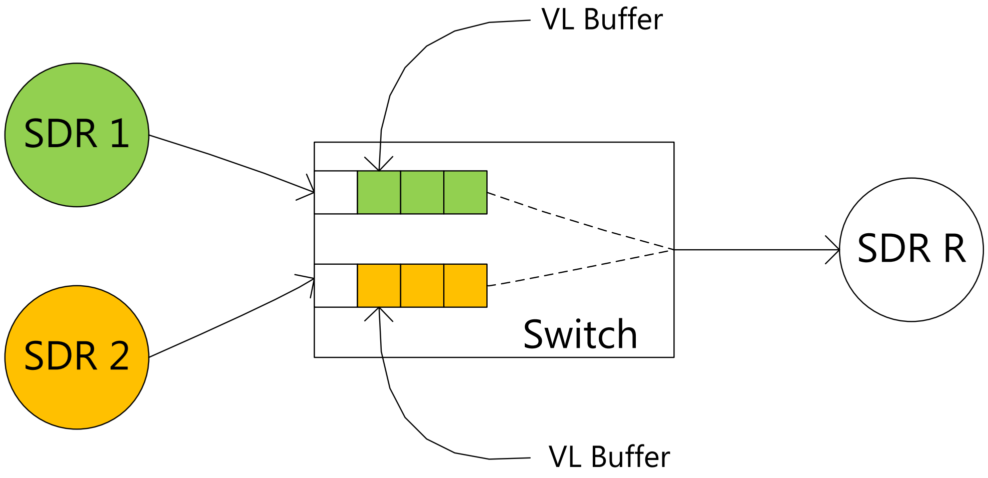
MTU 2k, data transmission is on VL0 (Start SDR 2 later than SDR 1)

1. at the very beginning, each SDR sender can inject packets in 2us
2. when congestion occurs, each SDR sender can only inject packets in 4us



4.2 Study of switch buffer size

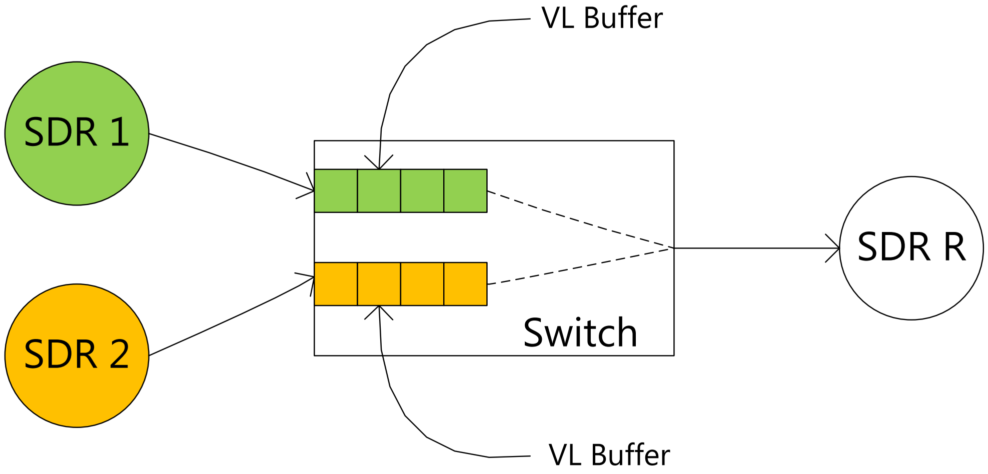
- Buffer space on each port is not full
- Packets can be put in 2us interval



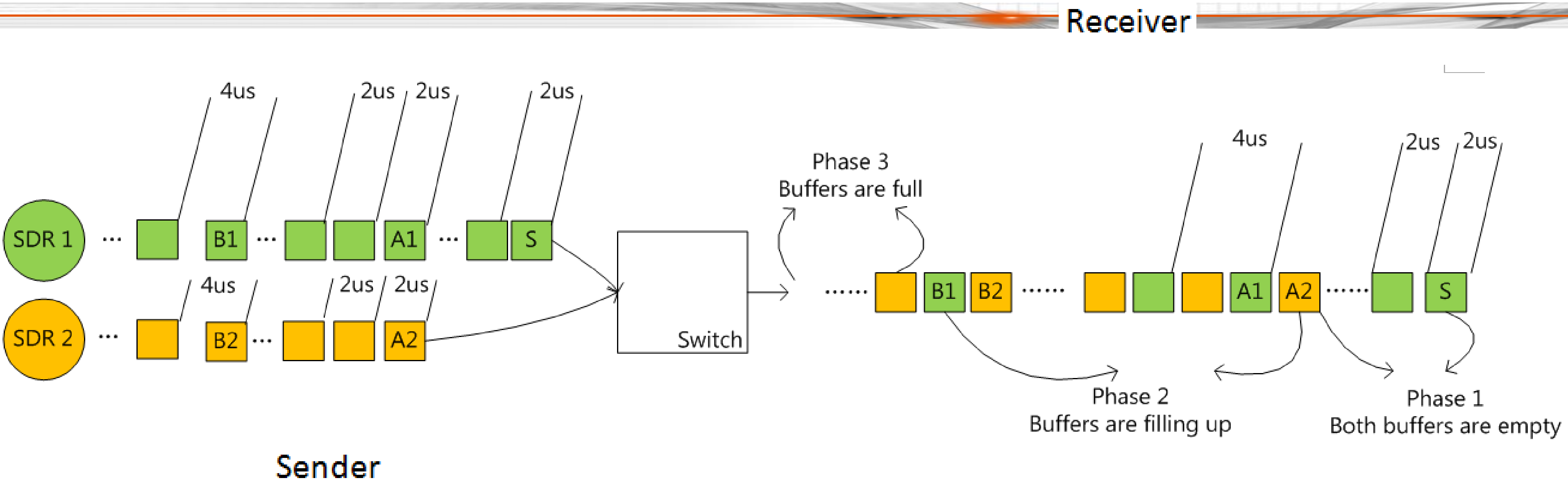
4.2 Study of switch buffer size

- Buffer space on each port is full

- Senders have to wait until there are enough buffer space on switch port to hold the data packets



4.2 Study of switch buffer size



A2: The first data packet of SDR 2 (SDR 2 is started later than SDR 1)

B1: The first SDR 1 data packet whose inter-packet interval on its sending side is 4us

4.2 Study of switch buffer size

On Mellanox SX6036 switch,

By counting the number of the green packets in the 2nd phase,

the determined switch input VL buffer space is around 32Ki bytes.

With configuration of 4 VLs, 4 * 32Ki = 128Ki bytes for each input port

4.3 Study of tick value

Congestion Indicator (counter) **PortXmitWait:**

Port counter that is used to indicate the "number of **ticks** during which selected port had data to transmit but none was sent during the entire tick either because of insufficient credits or due to lack of arbitration"

4.3 Study of tick value

PortXmitWait:

What is the tick?

Tick indicates the node’s sampling clock interval:

encoding value * symbol time

symbol time:

the time required to transmit an 8 bit data quantity onto a physical lane (SDR symbol time 4ns)

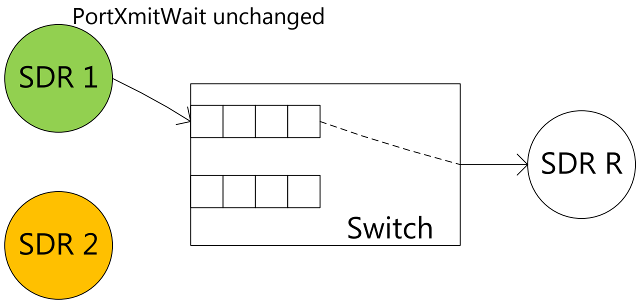
encoding value:

multiple of the symbol time. 1 ~ 256

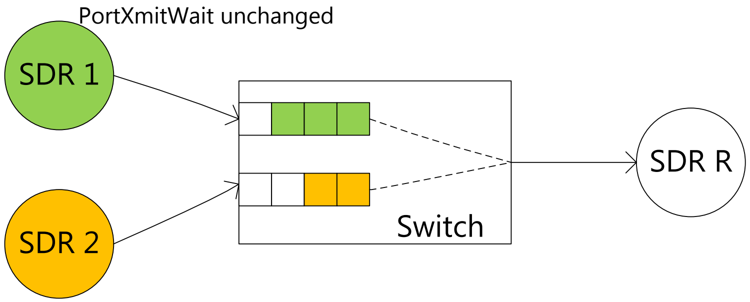
```
# perfquery –c LID Port_Number
```

4.3 Study of tick value

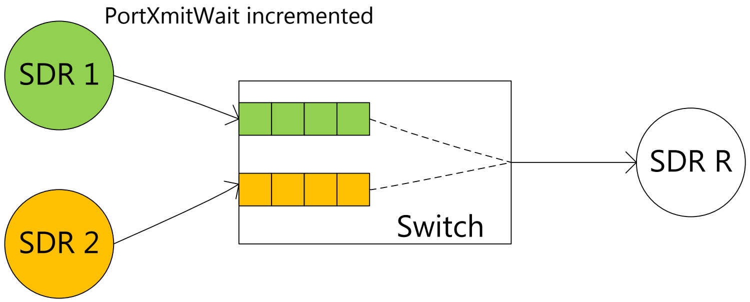
1) Both buffers are empty



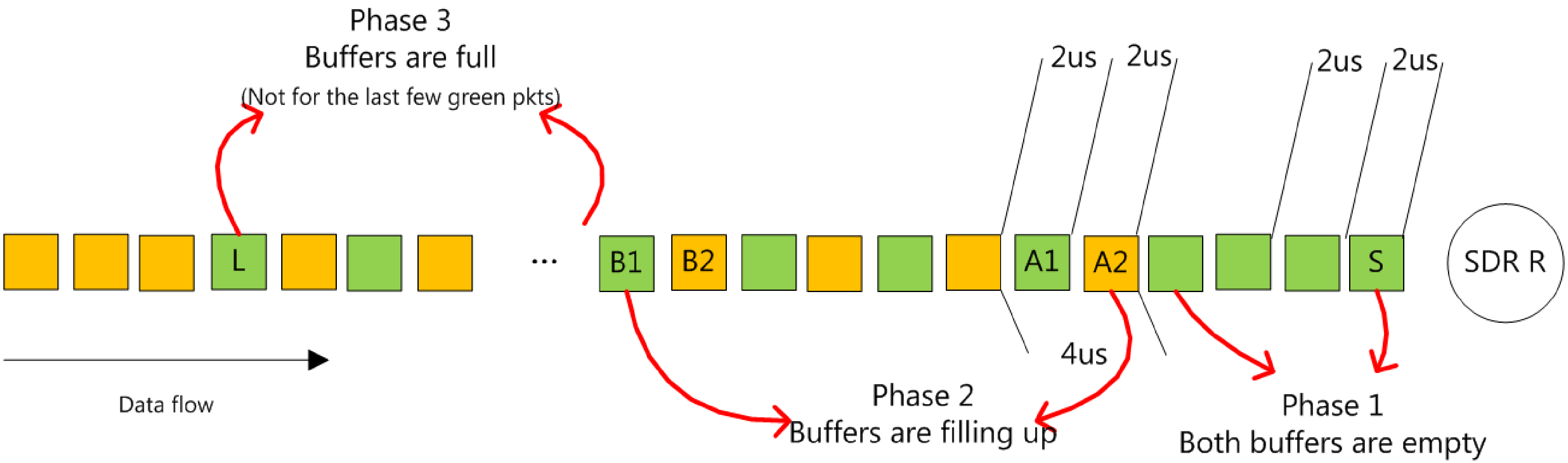
2) Buffers are filling up



3) Buffers are full



4.3 Study of tick value



A2: Time when SDR R starts receiving packets from both competing flows

B1: Time when the inter-packet intervals on each sender side go up to 4us

L: Time when SDR R receives the last SDR 1 data packet

4.3 Study of tick value

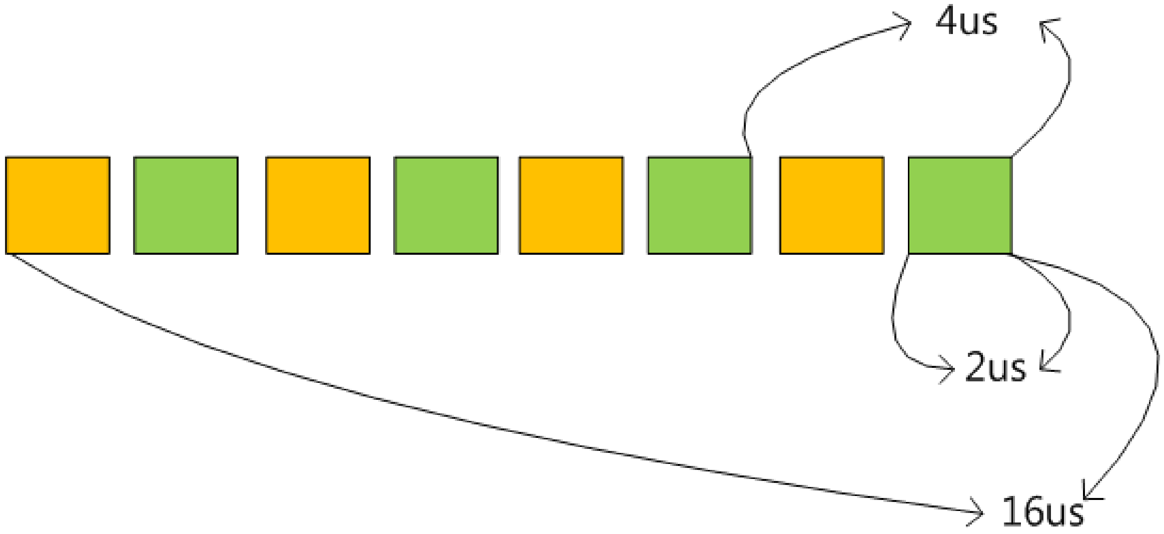
- *Tick* =

$$\frac{\mathit{Congestion\ Duration\ from\ the\ Point\ B1\ to\ the\ Point\ L}}{\mathit{PortXmitWait\ value\ increase\ in\ the\ time\ period\ (Point\ B1 \sim point\ L\ )}}$$

- Duration of the Congestion = $\mathrm{TIME}_{\mathrm{B1\text{-}L}}$ - $\mathrm{TIME}_{\mathrm{regular}}$

4.3 Study of tick value

- *Congestion time*



MLNX MT26428 QDR CA

encoding value = 31 = 0x1F

```
# perfquery –c  LID 1
Tick………………………..0x1F
```

Acknowledgement

I would like to thank for their support

- My advisor, Professor Robert D. Russell

- National Science Foundation Grant OCI-1127228

- Software Forge, Inc. -- for the loan of the CatC analyzers

- University of New Hampshire InterOperability Lab (UNH IOL)

Thank You

OPENFABRICS ALLIANCE

OpenFabrics Software User Group Workshop