Analytical Performance Evaluation of IPv6 and IPv4 Over 10 Gigabit Ethernet and InfiniBand using IPoIB

Eric Gamess
School of Computer Science
Central University of Venezuela
Caracas, Venezuela

Humberto Ortiz-Zuazaga
Department of Computer Science
University of Puerto Rico
San Juan, Puerto Rico

***Abstract*—IPv6 is the response to the shortage of IPv4 addresses. It was defined almost twenty years ago by the IETF as a replacement of IPv4, and little by little, it is becoming more preponderant as the Internet protocol. The growth of Internet has led to the development of high performance networks. On one hand, Ethernet has evolved significantly and today it is common to find 10 Gigabit Ethernet networks in LANs. On the other hand, another approach for high performance networking is based on RDMA (Remote Direct Memory Access) which offers innovative features such as kernel bypass, zero copy, offload of splitting and assembly of messages in packets to the CAs (Channel Adapters), etc. InfiniBand is currently the most popular technology that implements RDMA. It uses verbs instead of sockets and a big effort of the community is required to port TCP/IP software to InfiniBand, to take advantage of its benefits. Meanwhile, IPoIB (IP over InfiniBand) is a protocol that has been proposed and permits the execution of socket-based applications on top of InfiniBand, without any change, at the expense of performance. In this paper, we make a performance evaluation of IPv6 and IPv4 over 10 Gigabit Ethernet and IPoIB. Our results show that 10 Gigabit Ethernet has a better throughput than IPoIB, especially for small and medium payload sizes. However, as the payload size increases, the advantage of 10 Gigabit Ethernet is reduced in comparison to IPoIB/FDR. With respect to latency, IPoIB did much better than 10 Gigabit Ethernet. Finally, our research also indicates that in a controlled environment, IPv4 has a better performance than IPv6.**

***Keywords*—IPv4; IPv6; Performance Evaluation; InfiniBand; IP over InfiniBand; 10 Gigabit Ethernet; Benchmarking Tools**

I. INTRODUCTION

Internet Protocol version 6 (IPv6) [1][2][3][4] is around for some years now. It is a solution to the unexpected dramatic growth of the Internet, which is facing the exhaustion of available IPv4 addresses. This new version of IP has 128-bit addresses, while IPv4 is limited to 32-bit addresses. Furthermore, IPv6 adds many improvements in areas such as routing, multicasting, security, mobility, and network auto-configuration. According to IPv6 statistics made by Google [5], more than 10% of the users that require services from this company do it with an IPv6 connection. Cisco Systems [6] is gathering and publishing information about IPv6 deployment in the world. As reported by the recollected data, Belgium is the country with the highest IPv6 deployment in the world, with more than 55%. In the USA, the deployment is around 44%.

RDMA (Remote Direct Memory Access) communications differ from normal IP communications because they bypass kernel intervention in the communication process. That is with RDMA, the CA (Channel Adapter) directly places the application's buffer into packets on sending, and the content of the packets into the application's buffer on reception, without any intervention of the CPU. This allows a much better communication system with zero copy. Moreover, the CA also manages the splitting and assembly of messages into packets in RDMA, while IP fragmentation and TCP segmentation are in charge of the CPU in typical IP communications. As a result, RDMA provides high throughput and low latency while incurring a minimal amount of CPU load.

Recently, three major RDMA fabric technologies have emerged: InfiniBand [7][8][9], RoCE [10][11] (RDMA over Converged Ethernet), and iWARP [12][13] (internet Wide Area RDMA Protocol). InfiniBand seems to have the major acceptance of these three technologies, hence many manufacturers are offering a wide variety of products (CAs and switches), especially for the fields of HPC (High Performance Computing) and Data Centers. InfiniBand defines its own stack of protocols. Moreover, it does not use sockets as TCP/IP applications do, and is based on "verbs". To date, just few applications have been ported to verbs to work on top of InfiniBand. It is more than likely that it will take time to adapt popular socket-oriented applications to verbs. Hence, the IETF (Internet Engineering Task Force) has proposed a new protocol called IP over InfiniBand (IPoIB) [14][15][16] to run existing TCP/IP applications in an InfiniBand network without any changes.

In this work, we make a performance evaluation of 10 Gigabit Ethernet and IP over InfiniBand, where the former is the new de facto technology for local area network. We report the throughput and latency obtained at the level of UDP and TCP, for IPv6 and IPv4, when varying the payload size. To do so, we use famous benchmarking tools of the field of networking.

The rest of this paper is organized as follows: we discuss related work in Section II. A survey of InfiniBand and IPoIB is made in Section III. Section IV presents the testbed for our experiments, and some benchmarking tools for point-to-point network evaluation are introduced in Section V. The results of our network performance evaluation is presented and discussed in Section VI. Finally, Section VII concludes the paper and gives directions for future work in this area.

II. Related Work

In the field of the assessment of the performance of IPv6 and IPv4, there are several studies that evaluate their capacities based on benchmarking tools, with different operating systems and network technologies. Narayan, Shang, and Fan [17][18] studied the performance of TCP and UDP traffic with IPv6 and IPv4 on a Fast Ethernet LAN, using various distributions of Windows and Linux. A similar study was conducted by Kolahi et al. [19], where the TCP throughput of Windows Vista and Windows XP was compared using IPv6 and IPv4, also on a Fast Ethernet LAN. A comparison of the network performance between Windows XP, Windows Vista, and Windows 7 was conducted by Balen, Martinovic, and Hocenski [20], under IPv6 and IPv4. Their testbed consisted of two computers connected through a point-to-point link with Gigabit Ethernet. The authors of [21] assessed the throughput of UDP and TCP over IPv6 and IPv4 for Windows XP and Windows 7 in a point-to-point network, where the two end-nodes were connected by a Gigabit Ethernet link. Soorty and Sarkar [22][23] evaluated UDP over IPv6 and IPv4, using different modern operating systems. In [22], the computers of the testbed were running Windows 7, Windows Server 2008, Ubuntu Server 10.04, and Red Hat Enterprise Server 5.5. In [23], they used Ubuntu Server 10.04 and Red Hat Enterprise Server 5.5. For both cases [22][23], the network between the end-nodes also consisted of a back-to-back Gigabit Ethernet connection. The performance of the IP protocols has also been compared in wireless networks [24].

Some other efforts are more focused on modeling the performance of IPv6 and IPv4. An upper bound model to compute TCP and UDP throughput for IPv6 and IPv4 in Ethernet networks was presented by Gamess and Surós [25]. They compared the performance of various operating systems (Windows XP, Solaris 10, and Debian 3.1) with this upper bound, using a point-to-point network with Ethernet and Fast Ethernet technologies. Gamess and Morales [26] developed models to compute the throughput and latency of IPv6 and IPv4 in Ethernet LANs. They validated the proposed models doing experiments in Ethernet and Fast Ethernet networks, where the end-nodes were connected through a chain of routers (from 0 to 5 routers).

As far of InfiniBand is concerned, just a few works have been done. Cohen [27] did a low-level evaluation of InfiniBand (Send/Receive and RDMA operations) in a back-to-back connection between two end-nodes, i.e. a fabric without InfiniBand switches. Latency, throughput, and CPU load were reported by the author. In [28], Rashti and Afsahi evaluated three network technologies (10-Gigabit iWARP, 4X SDR InfiniBand, and Myrinet-10G) at the user-level and MPI [29] (Message Passing Interface) layer. The authors of [30] evaluated 4X FDR InfiniBand and 40GigE RoCE on HPC and cloud computing systems. They did some basic network level characterizations of performance, but most of the work is done with MPI point-to-point and collective communication benchmarks. In [31], Sur, Koop, Chai, and Panda did a network-level performance evaluation of the Mellanox ConnectX architecture on multi-core platforms. They evaluated low-level operations such as RDMA Write and RDMA Read, as well as high level applications as a whole.

As discussed in this section, many works have been done to evaluate the performance of IPv6 and IPv4 over Ethernet, Fast Ethernet, and Gigabit Ethernet. InfiniBand has also been assessed in a few studies, at low-level (Send/Receive, RDMA Read, and RDMA Write operations) and MPI level. To the best of our knowledge, this effort is the first one that compares the performance of IPv6 and IPv4 over 10 Gigabit Ethernet and InfiniBand using IPoIB.

III. A Survey of InfiniBand and IPoIB

In this section, we briefly introduce InfiniBand and IPoIB. We describe key concepts that can significantly help for the understanding of this research work.

A. Introduction to InfiniBand

InfiniBand [7][8][9] defines the notion of QPs (Queue Pairs) which consists of two queues: a SQ (Send Queue) and a RQ (Receive Queue). At the transport layer of the OSI model, InfiniBand offers several transport services which include: RC (Reliable Connection) and UD (Unreliable Datagram). In the RC transport service, a QP-to-QP connection must be established between the two RC QPs before transmission. It is a point-to-point connection, hence the involved QPs can only send packets to each other and receive packets from each other. An Ack/Nak mechanism permits the requester logic (QP SQ) to verify that all the packets are delivered to the responder (QP RQ). In the UD transport service, there is no initial connection setup with the remote QP prior to sending or receiving messages. It is not a QP-to-QP connection, hence the QPs can send and receive packets to and from any potential remote QPs.

InfiniBand has several speed grades known as: SDR (Simple Data Rate), DDR (Double Data Rate), QDR (Quadruple Data Rate), FDR (Fourteen Data Rate), and EDR (Enhanced Data Rate). SDR, DDR, and QDR use 8B/10B encoding, i.e., 10 bits carry 8 bits of data. In other words, the data rate in 80% of the signal rate. FDR and EDR use the more efficient 64B/66B encoding. Table I shows the signal rate and data rate achieved by InfiniBand, depending on the width of the link (1X, 4X, 8X, or 12X). The non-shaded rows represent the signal rate, while the shaded rows correspond to the data rate.

TABLE I. Signal and Data Rates Achieved by InfiniBand in Gbps

	SDR	DDR	QDR	FDR	EDR
1X	2.5	5.0	10.0	14.0625	25.78125
	2.0	4.0	8.0	13.64	25.0
4X	10.0	20.0	40.0	56.25	103.125
	8.0	16.0	32.0	54.54	100.0
8X	20.0	40.0	80.0	112.50	206.25
	16.0	32.0	64.0	109.09	200.00
12X	30.0	60.0	120.0	168.75	309.375
	24.0	48.0	96.0	163.63	300.00

B. Introduction to IPoIB

InfiniBand provides “verbs” to do low level IOs, but till date, very few applications have been developed with them. Hence, a mechanism is required to run TCP/IP on top of InfiniBand. The role of IPoIB [14][15][16] (IP over InfiniBand) is to provide an IP network emulation layer on top of InfiniBand networks, allowing the numerous existing socket-based applications to run over InfiniBand networks

unmodified. As a drawback, the performance of those applications will be considerably lower than if they were directly written to use RDMA communications natively, since they do not benefit from typical features offered by InfiniBand (kernel bypass, zero copy, splitting and assembly of messages to packets in the CAs, etc). However, for the users, IPoIB is a tradeoff between running their favorite socket-oriented applications without having to wait for the port to verbs and loosing part of the high performance of this emerging technology.

Linux has a module, called "ib_ipoib", for implementing IPoIB. This module creates a virtual NIC (ib0, ib1, ib2, etc) for each InfiniBand port on the system, which makes an HCA (Host Control Adapter) act like an ordinary NIC. IPoIB has two modes of operation: datagram mode [15] and connected mode [16]. In datagram mode the UD transport service is used, while the connected mode is based on the RC transport service. By default, IPoIB on Linux is configured in datagram mode. However, it is easy to switch between modes using the simple commands of Fig. 1. Line 01 shall be used to switch to connected mode, while Line 02 can be entered to switch to datagram mode.

```
01: echo connected > /sys/class/net/ib0/mode
02: echo datagram  > /sys/class/net/ib0/mode
```

Fig. 1. Switching between Datagram and Connected Modes

C. IPoIB in Connected Mode in a Unique Subnet Fabric

In the connected mode of IPoIB, the RC transport service is used. Hence a private connection must be established prior to the exchange of packets. Communication Management encompasses the protocols and mechanisms used to establish, maintain, and release channels for the RC transport service. The connection is established by exchanging three packets: ConnectRequest, ConnectReply, and ReadyToUse. Once the channel is created, the IPoIB packets can be sent. To close the connection, two packets must be sent: DisconnectRequest and DisconnectReply. Fig. 2 shows an IPoIB packet in connected mode in a fabric with a unique subnet. It is composed of three headers, the IPv6 or IPv4 packet per se, and two CRCs (Invariant CRC and Variant CRC).

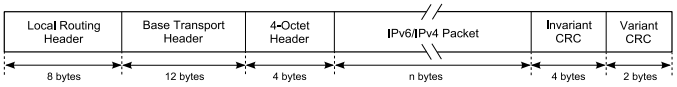

Fig. 2. IPoIB Packet in Connected Mode in a Fabric with a Unique Subnet

The first header, known as LRH (Local Routing Header), is shown in Fig. 3. It corresponds to the data-link layer of the OSI model. LNH (Link Next Header) is a 2-bit field that indicates the next header. It must be $(10)_2$ in a fabric with a unique subnet to inform that the next header is BTH (Base Transport Header). "Packet Length" is an 11-bit field, and its value shall equal the number of bytes in all the fields starting with the first byte of the LRH header and ending with the last byte of the Invariant CRC, inclusive, divided by 4. The Layer-2 address of the destination port is specified as "Destination Local Identifier" or DLID. The LIDs (Local Identifiers) are unique within a subnet and are assigned by the Subnet Manager during the initial startup or the reconfiguration of InfiniBand devices. The Layer-2 address of the port that injected the packet into the subnet is specified as "Source Local Identifier" or SLID.

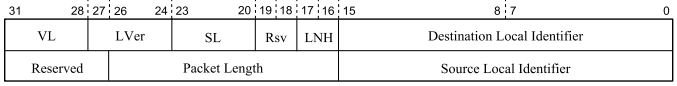

Fig. 3. Local Routing Header

The second header, known as BTH (Base Transport Header), is depicted in Fig. 4. It corresponds to the transport layer of the OSI model. Since RC is reliable, there is an Ack/Nak mechanism. The 1-bit A field of BTH is a request for the responder to schedule an acknowledgment for the packet. PSN (Packet Sequence Number) is a 24-bit field to identify the position of a packet within a sequence of packets. In that way, the responder can verify that all requested packets are received in order, and are only processed once. The 24-bit field called "Destination QP" identifies the receiving QP. Unlike IP, where the source and destination ports are present in all the segments sent by TCP, in InfiniBand just the destination QP is transported by a packet in the RC transport service. The source QP is not required, since RC is connection-oriented and both sides of the communication must keep information of the state of the connection, with includes the remote QP. That is, sending the source QP will be redundant and InfiniBand opts to save bandwidth by not transporting it.

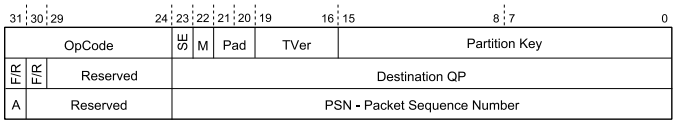

Fig. 4. Base Transport Header

The third header, known as "4-Octet Header", is shown in Fig. 5. The 16-bit field called "Type" is used to specify the type of payload (0x0800, 0x86DD, 0x0806, 0x8035 for IPv4, IPv6, ARP, and RARP, respectively).

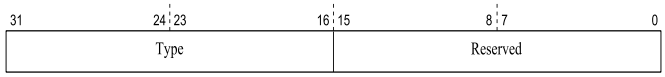

Fig. 5. 4-Octet Header

After these three headers (LRH, BTH, and 4-Octet header), the payload of the IPoIB packet can be either an IPv6 or an IPv4 packet, starting with its respective IPv6 or IPv4 header, and followed by its own payload. Finally, the IPoIB packet is finished with the two CRCs (Invariant CRC and Variant CRC) managed by InfiniBand.

D. IPoIB in Datagram Mode in a Unique Subnet Fabric

In the datagram mode of IPoIB, the UD transport service is used. Hence, there is no private connection between the requester and the responder. That is, it is not a QP-to-QP connection and no Ack/Nak mechanism is available. Even though each packet contains a sequential PSN (see Fig. 4), it is not meaningful because the entire message is encapsulated in a single packet. Fig. 6 shows an IPoIB packet in datagram mode in a fabric with a unique subnet.

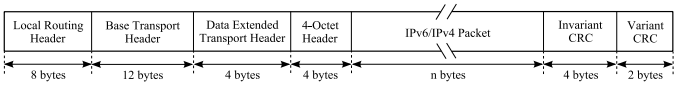

Fig. 6. IPoIB Packet in Datagram Mode in a Fabric with a Unique Subnet

It is similar to the one of the connected mode (see Fig. 2), with an additional header called DETH (Data Extended Transport Header).

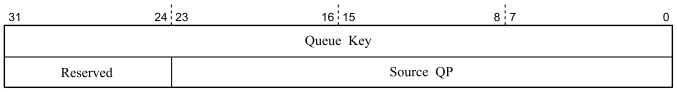

Fig. 7. Datagram Extended Transport Header

Fig. 7 shows the DETH header. Since the service is not oriented to connection, many remote devices can send packets to a local UD queue pair. Due to this, the sending QP is required in a packet and is specified in the 24-bit field called "Source QP". It is used by the receiver as the destination QP for response packets. The 32-bit field called "Queue Key" should be used for authentication to authorize access to the destination queue. The responder compares this field with the destination's QP key. Access will be allowed only if they are equal.

E. IPoIB in a Fabric with Multiple Subnets

In the case of an InfiniBand fabric with several subnets, routers are required to connect the subnets together. In this case, in addition to Layer-2 addresses which are LIDs (SLID and DLID in the LRH header as shown in Fig. 3), Layer-3 addresses are required. In InfiniBand, these addresses are called GIDs (Global Identifiers) and are 128-bit long and similar to IPv6 addresses. They are constructed by concatenating a 64-bit GID prefix with a EUI-64 (64-bit Extended Unique Identifier), where the latter is assigned by the manufacturer. Each subnet must have its own 64-bit GID prefix and is generally set by the network administrator.

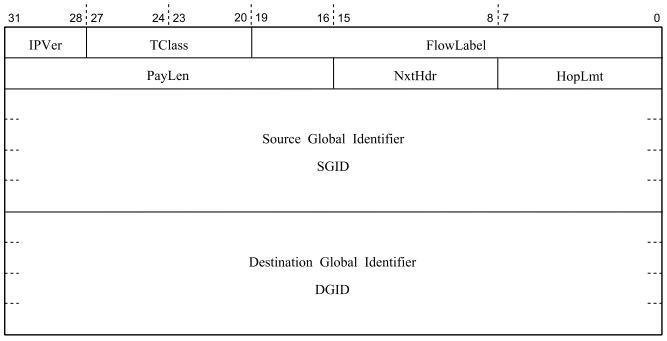

Fig. 8. Global Routing Header

An IPoIB packet traveling a fabric with several subnets is similar to the one of Fig. 2 in connected mode and Fig. 6 in datagram mode, with an additional header, called GRH (Global Routing Header), as shown in Fig. 8. GRH corresponds to the network layer of the OSI model and is placed between the LRH (data-link layer header) and the BTH (transport layer header). The SGID (Source Global Identifier) field corresponds to the GID of the port which injected the packet into the network. The DGID (Destination Global Identifier) identifies the GID for the port which will extract the packet from the network.

F. InfiniBand MTU

The IBTA (InfiniBand Trade Association) defines the following MTUs: 256, 512, 1024, 2048, or 4096 bytes. Messages must be segmented into packets for transmission according to the PMTU. Segmentation of messages into packets on transmission and reassembly on reception are provided by CAs (Channel Adapters) at the end-nodes.

IV. Testbed for Our Experiments

For our experiments, the testbed was based on a cluster with end-nodes that were running CentOS v6.6. As shown in Fig. 9, the cluster was made of four end-nodes, one InfiniBand switch (SW1), and one 10 Gigabit Ethernet switch (SW2). The InfiniBand switch was a Mellanox Technologies SX6012, with 12 QSFP ports that support full-duplex signal rate of 56 Gbps (FDR). It is a managed switch that can be administered through the CLI (Command Line Interface) and SNMP, and also offers IPMI (Intelligent Platform Management Interface) support. It was running Mellanox MLNX-OS version 3.4.2008 as operating system. The 10 Gigabit Ethernet switch was a Cisco Catalyst 4500-X with 16 ports (10 Gigabit Ethernet SFP+/SFP ports).

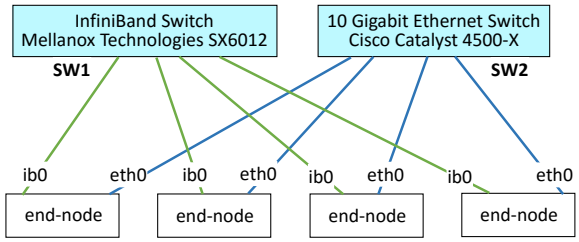

Fig. 9. Testbed for our Experiments

The connection between the end-nodes and the InfiniBand switch was based on 4X-width cables, while we used twinax cables between the end-nodes and the 10 Gigabit Ethernet switch. The InfiniBand fabric was configured with a 2048-byte MTU. The end-nodes had the following characteristics:

- Processors: 2 16-core Intel Xeon E5-2630 v3 at 2.4 GHz
- RAM: 64 GiB – 4 x 16 GiB DIMM (DDR4 2133 MHz)
- HCA: Mellanox Technologies single-port MT27500 ConnectX-3
- NIC: dual port NetXtreme II BCM57810 10 Gigabit Ethernet
- Hard Disk: Seagate ST1000NM0033 (1 TB, 7200 RPM, 128 MB Cache, SATA 6.0 Gb/s) for a local installation of the operating system (CentOS v6.6)
- Remote Management: IPMI.

It is worth clarifying that the InfiniBand network was composed of a single subnet, that is, the IPoIB packets did not have the GRH header (see Fig. 8) and were as shown in Fig. 2 and Fig. 6, for connected and datagram modes, respectively. Moreover, InfiniBand allows the Subnet Manager to be run in an end-node or in a switch. For our experiments, we chose to run it in a end-node.

V. Benchmarking Tools Used in Our Experiments

Many socket-based benchmarking tools have been proposed for network performance evaluation at the level of UDP and TCP. Unfortunately, not all of them support IPv6. Netperf is a benchmarking tool that can be used to measure various aspects of networking performance. It has support for IPv6 and IPv4. Its primary focus is on bulk data transfer (TCP_STREAM, UDP_STREAM, etc) and request/response performance (TCP_RR and UDP_RR) using either TCP or UDP. It is designed around the basic client/server model. In the TCP_STREAM test, a constant bitrate of data is transferred from the client (netperf) to the server (netserver), and the actual throughput is reported as the result. It is worth mentioning that the reported throughput is equal to the maximum throughput, since Netperf saturates the communication link. The UDP_STREAM test is similar to the TCP_STREAM test, except that UDP is used as the transport protocol rather than TCP. In the TCP_RR test, a fixed quantity of data is exchanged by TCP between the client (netperf) and the server (netserver) a number of times, and the benchmark reports the transaction rate which is the number of complete round-trip transactions per second. The UDP_RR is very much the same as the TCP_RR test, except that UDP is used rather than TCP.

Since Netperf does not report the latency, we developed our own benchmarking tool using the C programming language for IPv6 and IPv4. The benchmark is based on the client/server model. Basically, an UDP datagram or TCP segment with a fixed payload length is exchanged between the client and the server a number of times. We take a timestamp before and after the interchange. The difference of the timestamps is divided by the number of times the message was sent and received, and by 2 to get the average latency.

VI. Results and Analytical Comparison

In this section, we do several experiments to measure the performance of IPv6 and IPv4, in our testbed, with 10 Gigabit Ethernet and IPoIB. All the throughput measurements were done with Netperf. Regarding the latency assessments, we used the benchmarking tool that we developed. Also, it is important to clarify that each experiment was repeated several times, and the result that we report is an average, for a better consistency.

A. Experiments when Changing the IPoIB Mode

The objective of these first experiments is to compare the performance achieved by IPoIB in datagram and connected modes, for IPv6 and IPv4. For these performance tests, we chose FDR for the signal rate of InfiniBand.

Table II shows the results obtained for the UDP throughput when varying the payload size from 4 to 32,768 bytes. We did not take biggest UDP payload sizes since we were limited by the IPv4 maximum packet size. These experiments indicate that the throughput for the datagram mode is higher than the one of the connected mode. Also, it is noticeable that IPv4 has a better throughput than IPv6.

TABLE II. UDP Throughput in Mbps for Ipv6 and Ipv4 Over Ipoib/FDR in Datagram and Connected Modes

Payload Size	Datagram Mode		Connected Mode	
	IPv6	IPv4	IPv6	IPv4
4	13.21	16.41	10.15	11.43
8	25.22	34.63	19.46	21.09
16	54.51	66.10	43.37	44.21
32	119.04	134.78	86.10	88.78
64	255.12	271.18	175.48	178.67
128	513.77	544.76	352.92	356.77
256	902.85	1,064.14	429.53	432.51
512	2,137.53	2,162.18	723.49	725.36
1,024	3,096.52	3,107.23	1,145.08	1,152.42
2,048	3,805.11	3,851.54	1,411.72	1,429.32
4,096	4,408.75	4,583.80	1,502.55	1,514.37
8,192	6,101.41	6,352.89	3,402.70	3,418.76
16,384	6,414.42	7,925.14	6,301.15	7,643.12
32,768	7,438.32	9,721.07	7,030.24	7,777.30

Table III shows the results obtained for the UDP latency when varying the payload size from 4 to 32,768 bytes. We did not take biggest UDP payload sizes since we were limited by the IPv4 maximum packet size. These experiments indicate that the latency for the datagram mode is lower than the one of the connected mode. Also, it is noticeable that IPv4 has a better latency than IPv6.

TABLE III. UDP Latency in Microseconds for Ipv6 and Ipv4 Over Ipoib/FDR in Datagram and Connected Modes

Payload Size	Datagram Mode		Connected Mode	
	IPv6	IPv4	IPv6	IPv4
4	10.92	7.37	11.25	7.38
8	11.05	7.41	11.31	7.42
16	11.23	7.45	11.47	7.49
32	11.51	7.51	11.72	7.55
64	11.75	7.53	11.89	7.58
128	12.02	7.55	12.15	7.62
256	12.40	7.62	12.45	7.65
512	12.53	7.75	12.57	7.76
1,024	12.72	7.83	12.82	7.89
2,048	15.01	11.17	15.42	11.31
4,096	19.47	14.76	19.77	15.85
8,192	21.32	18.37	22.63	18.47
16,384	31.58	23.19	34.92	31.52
32,768	48.32	34.98	50.32	45.74

According to these first experiments, the IPoIB datagram mode seems to have a better performance than the IPoIB connected mode. Hence, the rest of our experiments were done with the datagram mode.

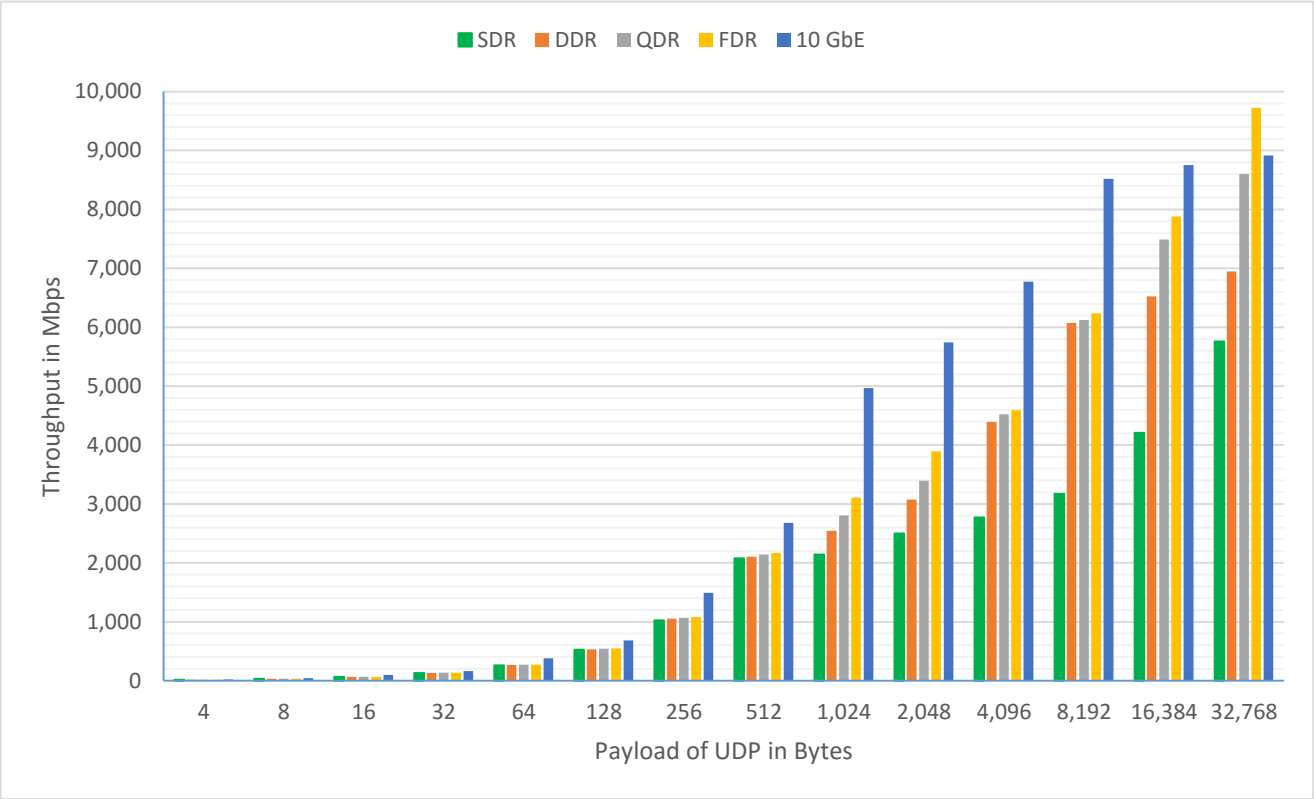

Fig. 10. Throughput for UDP/IPv4 over 10 GbE and IPoIB in Datagram Mode when Varying the Signal Rate of InfiniBand

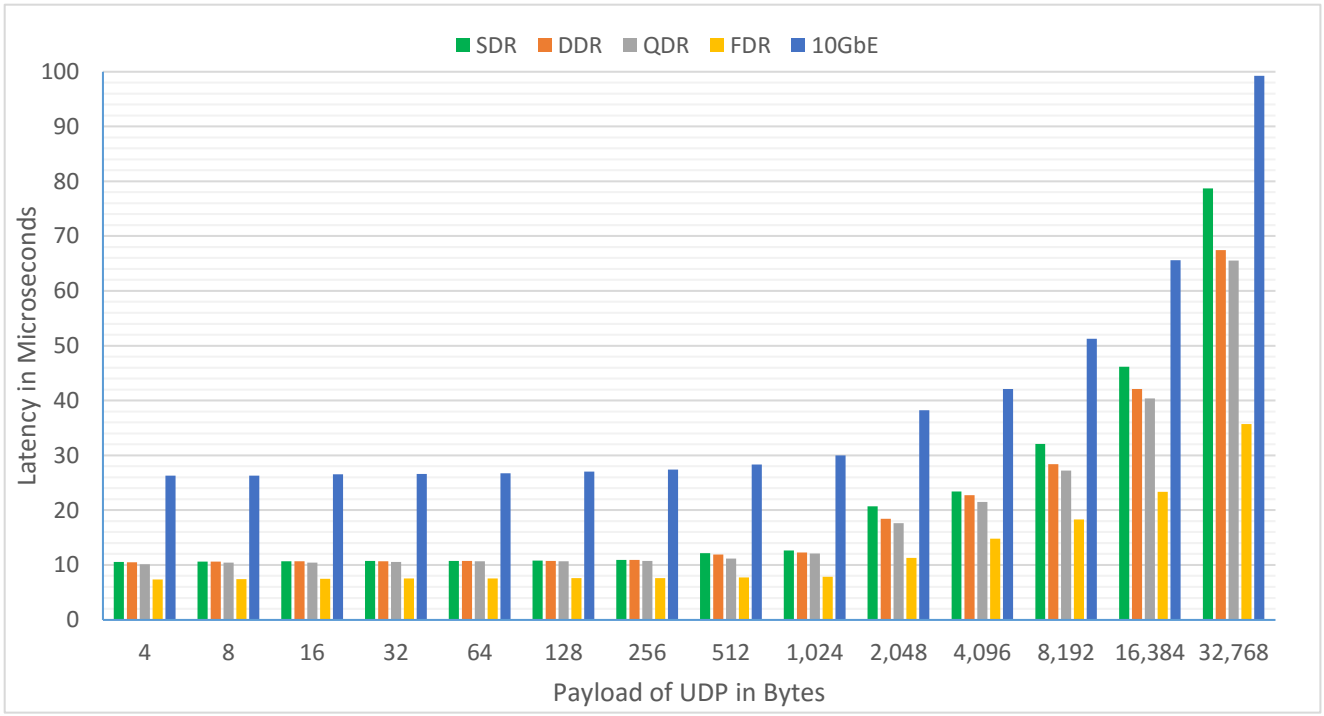

Fig. 11. Latency for UDP/IPv4 over 10 GbE and IPoIB in Datagram Mode when Varying the Signal Rate of InfiniBand

B. *Experiments when Changing the Signal Rate in UDP/IPv4*

In these experiments, we compare the throughput and latency obtaining between two end-nodes, for UDP/IPv4, for both 10 Gigabit Ethernet and IPoIB when changing the signal rate. In the case of 10 Gigabit Ethernet (10GbE), we took the default configuration. Meanwhile for IPoIB, we chose the datagram mode in the end-nodes (see Fig. 1) and we had to

adjust the signal rate (SDR, DDR, QDR, and FDR) in the interfaces of the InfiniBand switch.

Fig. 10 depicts the throughput obtained when varying the UDP payload size from 4 to 32,768 bytes. For each value of the UDP payload size, there are five bars. The first four bars represent the SDR, DDR, QDR, and FDR throughputs for IPoIB, respectively. The last and fifth bar is for 10GbE. Our experiments indicate that the throughput of 10GbE outperforms the ones of IPoIB, except for very large payload sizes (e.g. 32,768 bytes) where IPoIB in FDR has the best throughput. As far of InfiniBand is concerned, the results of the throughput are as expected, that is SDR has the lowest one, while FDR has the biggest one.

Fig. 11 shows the latency obtained when varying the UDP payload size from 4 to 32,768 bytes. For each value of the UDP payload size, there are five bars. The first four bars represent the SDR, DDR, QDR, and FDR latencies for IPoIB, respectively. The last and fifth bar is for 10GbE. From our experiments, we can see that 10GbE has the biggest latency. Also, it is worth to point out that for each technology, the latency does not vary for small UDP payload sizes, and a difference can be noticed when the size is greater than or equal to 256 bytes. With respects to InfiniBand, the results of the latency are as expected, that is SDR has the biggest one while FDR has the lowest one.

C. Experiments for UDP/IPv6 and UDP/IPv4

In these experiments, we compare the throughput and latency obtaining between two end-nodes for UDP/IPv6 and UDP/IPv4, over 10GbE and IPoIB/FDR. In the case of IPoIB, we chose the datagram mode in the end-nodes (see Fig. 1). Whereas for 10GbE, we took the default configuration.

Table IV shows the results of the throughput when varying the UDP payload size from 4 to 32,768 bytes. We can see that 10GbE has the best throughput for almost all the payload sizes, except for very large payload sizes (e.g. 32,768 bytes) where the performance of IPoIB/FDR is better. Also, IPv4 exceeds IPv6 for both technologies.

TABLE IV. UDP THROUGHPUT IN MBPS FOR IPV6 AND IPV4 OVER 10GBE AND IPOIB/FDR

Payload Size	10GbE		IPoIB/FDR	
	IPv6	IPv4	IPv6	IPv4
4	14.38	20.15	13.21	16.41
8	26.97	45.22	25.22	34.63
16	57.68	100.46	54.51	66.10
32	112.65	164.86	109.04	134.78
64	272.96	383.41	255.12	271.18
128	531.80	703.52	513.77	544.76
256	910.42	1,492.17	902.85	1,064.14
512	2,254.57	2,723.55	2,137.53	2,162.18
1,024	3,490.73	4,967.32	3,096.52	3,107.23
2,048	4,319.39	5,742.73	3,805.11	3,851.54
4,096	5,107.52	6,784.56	4,408.75	4,583.80
8,192	6,208.20	8,522.13	6,101.41	6,352.89
16,384	6,532.70	8,796.20	6,414.42	7,925.14
32,768	6,754.10	8,893.05	7,438.32	9,721.07

Table V gives the results of the latency when varying the UDP payload size from 4 to 32,768 bytes. We can see that IPoIB/FDR has the lowest latency for all the payload sizes. Also, it is worth pointing out that the latency of IPv4 is under the one of IPv6.

TABLE V. UDP LATENCY IN MICROSECONDS FOR IPV6 AND IPV4 OVER 10GBE AND IPOIB/FDR

Payload Size	10GbE		IPoIB/FDR	
	IPv6	IPv4	IPv6	IPv4
4	31.17	26.28	10.92	7.37
8	31.32	26.30	11.05	7.41
16	31.56	26.57	11.23	7.45
32	31.71	26.59	11.51	7.51
64	31.93	26.70	11.75	7.53
128	32.46	27.01	12.02	7.55
256	32.82	27.42	12.40	7.62
512	33.41	28.30	12.53	7.75
1,024	35.72	30.02	12.72	7.83
2,048	46.17	38.39	15.01	11.17
4,096	50.62	42.08	19.47	14.76
8,192	60.58	51.25	21.32	18.37
16,384	78.32	65.59	31.58	23.19
32,768	114.41	99.23	48.32	34.98

D. Experiments for TCP/IPv6 and TCP/IPv4

In these experiments, we compare the throughput and latency obtaining between two end-nodes for TCP/IPv6 and TCP/IPv4, over 10GbE and IPoIB/FDR. In the case of IPoIB, we chose the datagram mode in the end-nodes (see Fig. 1). Meanwhile for 10GbE, we took the default configuration.

Table VI shows the results of the throughput when varying the TCP payload size from 4 to 32,768 bytes. We can see that 10GbE has the best throughput for almost all the payload sizes, except for very large payload sizes (e.g. 32,768 bytes) where the performance of IPoIB/FDR is better. Also, IPv4 exceeds IPv6 for both technologies.

TABLE VI. TCP THROUGHPUT IN MBPS FOR IPV6 AND IPV4 OVER 10GBE AND IPOIB/FDR

Payload Size	10GbE		IPoIB/FDR	
	IPv6	IPv4	IPv6	IPv4
4	13.72	19.62	12.39	15.28
8	24.32	39.45	23.41	32.73
16	55.15	73.93	50.73	64.23
32	109.46	142.52	105.47	130.94
64	258.72	297.58	241.69	265.44
128	510.60	613.78	497.08	540.98
256	904.78	1,176.34	897.43	1,061.54
512	2,231.78	2,296.75	2,005.81	2,157.76
1,024	3,482.23	3,984.84	3,087.36	3,102.42
2,048	4,315.17	5,654.91	3,798.34	3,845.72
4,096	5,098.92	6,575.67	4,401.43	4,575.59
8,192	6,201.53	8,343.26	6,098.52	6,345.74
16,384	6,528.38	8,576.45	6,407.31	7,917.27
32,768	6,742.57	8,744.21	7,429.50	9,714.57

Table VII gives the results of the latency when varying the TCP payload size from 4 to 32,768 bytes. We can see that IPoIB/FDR has the lowest latency for all the payload sizes. Also, it is worth pointing out that the latency of IPv4 is under the one of IPv6.

TABLE VII. TCP LATENCY IN MICROSECONDS FOR IPV6 AND IPV4 OVER 10GBE AND IPOIB/FDR

Payload Size	10GbE		IPoIB/FDR	
	IPv6	IPv4	IPv6	IPv4

4	32.24	30.02	17.02	14.32
8	32.51	30.62	17.14	14.41
16	32.76	31.04	17.50	14.58
32	32.87	31.46	17.72	14.65
64	33.36	31.82	18.01	14.72
128	33.75	32.17	18.15	14.81
256	34.63	32.90	18.28	14.88
512	35.27	33.42	18.42	14.92
1,024	37.94	35.63	18.82	15.26
2,048	52.47	50.79	21.18	18.41
4,096	53.02	51.82	26.24	23.52
8,192	61.23	53.48	34.51	31.02
16,384	79.17	67.97	40.26	36.41
32,768	114.98	103.77	55.73	53.91

VII. Conclusions and Future Work

In this research work, we analyzed the performance of IPv6 and IPv4 over 10 Gigabit Ethernet and IPoIB. For small and medium IPv6 and IPv4 packets, our experiments showed that the throughput of 10 Gigabit Ethernet is over the one shown by IPoIB/FDR and the differences are significant. However, as the size of the UDP and TCP payload increases, IPoIB/FDR improves its performance and finally outperforms 10 Gigabit Ethernet. Regarding latency, IPoIB/FDR does better than 10 Gigabit Ethernet for all the UDP and TCP payload sizes. Additionally, our experiments showed that the performance of IPv4 is over the performance of IPv6, however, the differences are small and are mostly likely due to the IP headers, 20 bytes in IPv4 and 40 bytes in IPv6, resulting in a higher transmission time for IPv6.

As future work, we are planning to develop some mathematical models to represent the maximum throughput and the minimum latency that can be achieved by different transport services of InfiniBand, in connections between two end-nodes with zero or more intermediate switches between them. Another direction of research that we also want to explore is the performance evaluation of parallel file systems over InfiniBand (e.g. Lustre [32] and NFS over RDMA).

Acknowledgment

We want to thank the NSF (National Science Foundation) which partially supported this research under grant number 1010094 through the EPSCoR Track 2 program. We also express our gratitude for all the valuable assistance and comments that we received throughout this project from José Bonilla and Ramón Sierra of the High Performance Computer Facility of the University of Puerto Rico, and José Muñoz and Osvaldo Casiano of the Resource Center for Science and Engineering of the University of Puerto Rico.

References

[1] J. Davies, Understanding IPv6, 3rd edition, Microsoft Press, June 2012.

[2] J. Pyles, J. Carrell, and E. Tittel, Guide to TCP/IP: IPv6 and IPv4, 5th edition, Cengage Learning, June 2016.

[3] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460. December 1998.

[4] M. Dooley and T. Rooney, IPv6 Deployment and Management, 1st edition, Wiley-IEEE Press, May 2013.

[5] Google IPv6, https://www.google.com/intl/en/ipv6.

[6] 6lab - The Place to Monitor IPv6 Adoption, http://6lab.cisco.com.

[7] InfiniBand Trade Association, InfiniBand Architecture Specification Volume 1, Release 1.3, March 2015.

[8] InfiniBand Trade Association, InfiniBand Architecture Specification Volume 2, Release 1.3, March 2015.

[9] MindShare and T. Shanley, InfiniBand Network Architecture, 1st edition, Addison Wesley, November 2002.

[10] InfiniBand Trade Association, Supplement to InfiniBand Architecture Specification Volume 1, Release 1.2.1, RDMA over Converged Ethernet (RoCE), Annex A16, April 2010.

[11] InfiniBand Trade Association, Supplement to InfiniBand Architecture Specification Volume 1, Release 1.2.1, RoCEv2, Annex A17, September 2014.

[12] Mellanox, RoCE vs. iWARP Competitive Analysis, White Paper, August 2015.

[13] Chelsio Communications, The Case Against iWARP, 2015.

[14] V. Kashyap, IP over InfiniBand (IPoIB) Architecture, RFC 4392, April 2006.

[15] J. Chu and V. Kashyap, Transmission of IP over InfiniBand (IPoIB), RFC 4391, April 2006.

[16] V. Kashyap, IP over InfiniBand: Connected Mode, RFC 4755, December 2006.

[17] S. Narayan, P. Shang, and N. Fan, "Performance Evaluation of IPv4 and IPv6 on Windows Vista and Linux Ubuntu," in proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC'09), Wuhan, China, April 2009.

[18] S. Narayan, P. Shang, and N. Fan, "Network Performance Evaluation of Internet Protocols IPv4 and IPv6 on Operating Systems," in proceedings of the 6th International Conference on Wireless and Optical Communications Networks (WOCN'09), Cairo, Egypt, April 2009.

[19] S. S. Kolahi, B. K. Soorty, Z. Qu, and N. Chand, "Performance Analysis of IPv4 and IPv6 on Windows Vista and Windows XP over Fast Ethernet in Peer-peer LAN," in proceedings of the 3rd International Conference on New Technologies, Mobility and Security (NTMS'09). Cairo, Egypt, December 2009.

[20] J. Balen, G. Martinovic, and Z. Hocenski, "Network Performance Evaluation of Latest Windows Operating Systems," in proceedings of the 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), vol. 7, Split, Croatia, September 2012.

[21] P. Jain, S. Singh, G. Singh, and C. Goel, "Performance Comparison of IPv4 and IPv6 using Windows XP and Windows 7 over Gigabit Ethernet LAN," International Journal of Computer Applications, vol. 43, no. 16, April 2012.

[22] B. Soorty and N. Sarkar, Evaluating IPv6 in Peer-to-Peer Gigabit Ethernet for UDP using Modern Operating Systems, in proceedings of the 2012 IEEE Symposium on Computers and Communications (ISCC 2012), Cappadocia, Turkey, July 2012.

[23] B. Soorty and N. Sarkar, "UDP-IPv6 Performance in Peer-to-Peer Gigabit Ethernet using Modern Windows and Linux Systems," International Journal of Computer and Information Technology (IJCIT), vol. 3, no. 3, pp. 496-502, May 2014.

[24] H. Fahmy and S. Ghoneim, "Performance Comparison of Wireless Networks over IPv6 and IPv4 under Several Operating Systems," in proceedings of the IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS'13), Abu Dhabi, UAE, December 2013, pp. 670–673.

[25] E. Gamess and R. Surós, "An Upper Bound Model for TCP and UDP Throughput in IPv4 and IPv6," Journal of Network and Computer Applications, vol. 31, no. 4, pp. 585-602, November 2008.

[26] E. Gamess and N. Morales, "Modeling IPv4 and IPv6 Performance in Ethernet Networks," International Journal of Computer and Electrical Engineering, vol. 3, no. 2, pp. 282-288, April, 2011.

[27] A. Cohen, A Performance Analysis of 4X InfiniBand Data Transfer Operations, in proceedings of the 2003 International Parallel and Distributed Processing Symposium (IPDPS 2003), Nice, France, April 2003.

[28] M. Rashti and A. Afsahi, 10-Gigabit iWARP Ethernet: Comparative Performance Analysis with InfiniBand and Myrinet-10G, in proceedings of the 2007 IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007), Long Beach, CA, USA, March 2007.

[29] Message Passing Interface Forum, MPI: A Message-Passing Interface Standard, Version 3.1, High Performance Computing Center Stuttgart, June 2015.

[30] J. Vienne, J. Chen, Md. Wasi-ur-Rahman, N. Islam, H. Subramoni, and D. Panda, Performance Analysis and Evaluation of InfiniBand FDR and 40GigE RoCE on HPC and Cloud Computing Systems, in proceedings of the Symposium on High-Performance Interconnects, Santa Clara, CA, USA, August 2012.

[31] S. Sur, M. Koop, L. Chai, and D. Panda, Performance Analysis and Evaluation of Mellanox ConnectX InfiniBand Architecture with Multi-Core Platforms, in proceedings of the 15th Annual IEEE Symposium on High-Performance Interconnects (HOTI 2007), Stanford, CA, USA, August 2007.

[32] Y. Wang, Y. Lu, C. Qiu, P. Gao, and J. Wang, Performance Evaluation of a InfiniBand-based Lustre Parallel File System, in proceedings of the 2011 2nd International Conference on Challenges in Environmental Science and Computer Engineering (CESCE 2011), Haikou, China, December 2011.