University of Castilla-La Mancha

A publication of the

Department of Computer Science

A Strategy to Compute the InfiniBand Arbitration Tables

by

Francisco J. Alfaro, José L. Sánchez, José Duato

Technical Report **#DIAB-01-02-20** October 2001

Part of this work has been submitted to International Parallel and Distributed Processing Symposium (IPDPS'2001)

DEPARTAMENTO DE INFORMÁTICA
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE CASTILLA-LA MANCHA
Campus Universitario s/n
Albacete - 02071 - Spain
Phone +34.967.599200, Fax +34.967.599224

A Strategy to Compute the InfiniBand Arbitration Tables *

Francisco J. Alfaro, José L. Sánchez

Dept. de Informática
Escuela Politécnica Superior
Universidad de Castilla-La Mancha
02071- Albacete, Spain
{falfaro, jsanchez}@info-ab.uclm.es

José Duato

Dept. de Informática de
Sistemas y Computadores
Universidad Politécnica de Valencia
46071- Valencia, Spain
jduato@gap.upv.es

Abstract

The InfiniBand Architecture (IBA) is a new industry-standard architecture for server I/O and interprocessor communication. InfiniBand is very likely to become the de facto standard in a few years. It is being developed by the InfiniBand^{SM} Trade Association (IBTA) to provide the levels of reliability, availability, performance, scalability, and quality of service (QoS) necessary for present and future server systems.

The provision of QoS in data communication networks is currently the focus of much discussion and research in industry and academia. IBA enables QoS support with some mechanisms. In this paper, we examine these mechanisms and describe a way to use them. We propose a traffic segregation strategy based on mean bandwidth requirements. Moreover, we propose a very effective strategy to compute the virtual lane arbitration tables for IBA switches. We evaluate our proposal with different network topologies. Performance results show that, with a correct treatment of each traffic class in the arbitration of the output port, every traffic class meets its QoS requirements.

1 Introduction

Input-output (I/O) buses have become the bottleneck for disk accesses, especially in high-performance servers. While buses have the major advantage of simplicity, and have served the industry well up to this point, bus-based I/O systems do not use their underlying electrical technology well enough to provide data transfer bandwidth out of a system to devices.

Despite recent upgrades, the most popular I/O bus (PCI bus) offers limited bandwidth, concurrency and reliability, and this limitation is unacceptable for a lot of actual applications and server systems. A single device failure can inhibit the correct operation of the bus itself, causing all the devices on the bus to become unavailable.

Several external interconnects, such as Fiber Channel, have been used to overcome these difficulties. However, they still enter the processing complex through an industry-standard bus, making it impossible to avoid the bottlenecks and low availability characteristic of standard I/O buses.

This was the primary reason why a group of the most important companies joined together to develop a standard for communication between processing nodes and I/O devices as well as

*This work was partly supported by the Spanish CICYT under Grant TIC2000-1151-C07

for interprocessor communication, known as InfiniBand [7]. The InfiniBand^{SM} Trade Association (IBTA) is a group of 200 or more companies founded in August 1999 to develop IBA. Membership is also open to Universities, research laboratories, and others. The IBTA is lead by a Steering Committee whose members come from Dell, Compaq, HP, IBM, Intel, Microsoft, and Sun, co-chaired by IBM and Intel. Sponsor companies are 3Com, Cisco Systems, Fujitsu-Siemens, Hitachi, Adaptec, Lucent Technologies, NEC, and Nortel Networks.

The first specification of the InfiniBand Architecture (release 1.0) was published in October 2000 [5], and more than 200 companies are currently supporting the InfiniBand initiative.

On the other hand, most of the current networking products have tried to achieve maximum throughput and minimum latency, leaving aside other aspects like guarantee of bandwidth, maximum latency deadline, interarrival delays, etc [10]. Many current applications need those characteristics that not all the current networks are able to provide. The current network that can best provide QoS is probably ATM [3, 4]. However, ATM focuses more upon wide area networks, and it can only be used with great difficulty in LAN environments. It is not suitable for connections between a processor and its devices since it introduces significant overhead.

The Internet Engineering Task Force (IETF) is currently in the process of developing an architecture for providing QoS on the Internet. This effort is referred to as Differentiated Services [1].

Therefore, it would be important for InfiniBand to be able to satisfy both the applications that only need minimum latency, and also those different applications that need other characteristics to satisfy its QoS requirements. InfiniBand provides a series of mechanisms that, when properly used, are able to provide QoS to the applications. These mechanisms are mainly the segregation of traffic according to categories and the arbitration of the output ports according to an arbitration table that can be configured to give priority to the packets with the most need for QoS.

InfiniBand distinguishes up to a maximum of 16 service levels, but it does not specify which characteristics will bear the traffic of those service levels. Nor does it specify how the arbitration table should be filled in, as it only specifies the form that this table should have, leaving the implementation of the table to the manufacturers or users consideration, according to the characteristics that they want to obtain.

Recently, an overview of a possible implementation of DiffServ over IBA has been described in [8]. In this study the traffic is classified in several categories and the author proposes that the arbitration tables of InfiniBand should deal with each category in a different way, but no attempt is made to indicate how to fill in those tables.

In this paper, we propose a classification of the different traffic types with QoS needs, based on the proposal made in [8], as well as a strategy to compute the arbitration tables for the IBA switch ports to obtain the QoS required by the applications.

The structure of the paper is as follows: Section 2 presents a summary of the general aspects in the specifications of InfiniBand. In Section 3, we explain the most important mechanisms that InfiniBand provides to support QoS. In Section 4, we present our proposal, and its performance is evaluated in Section 5. Finally, some conclusions are given and future work is proposed.

2 InfiniBand

The InfiniBand Architecture (IBA) Specification describes a System Area Network (SAN) for connecting multiple independent processor platforms (i.e. host processor nodes), I/O platforms and I/O devices. The IBA SAN is a communications and management infrastructure supporting both I/O and interprocessor communications for one or more computer systems (see Figure 1). The

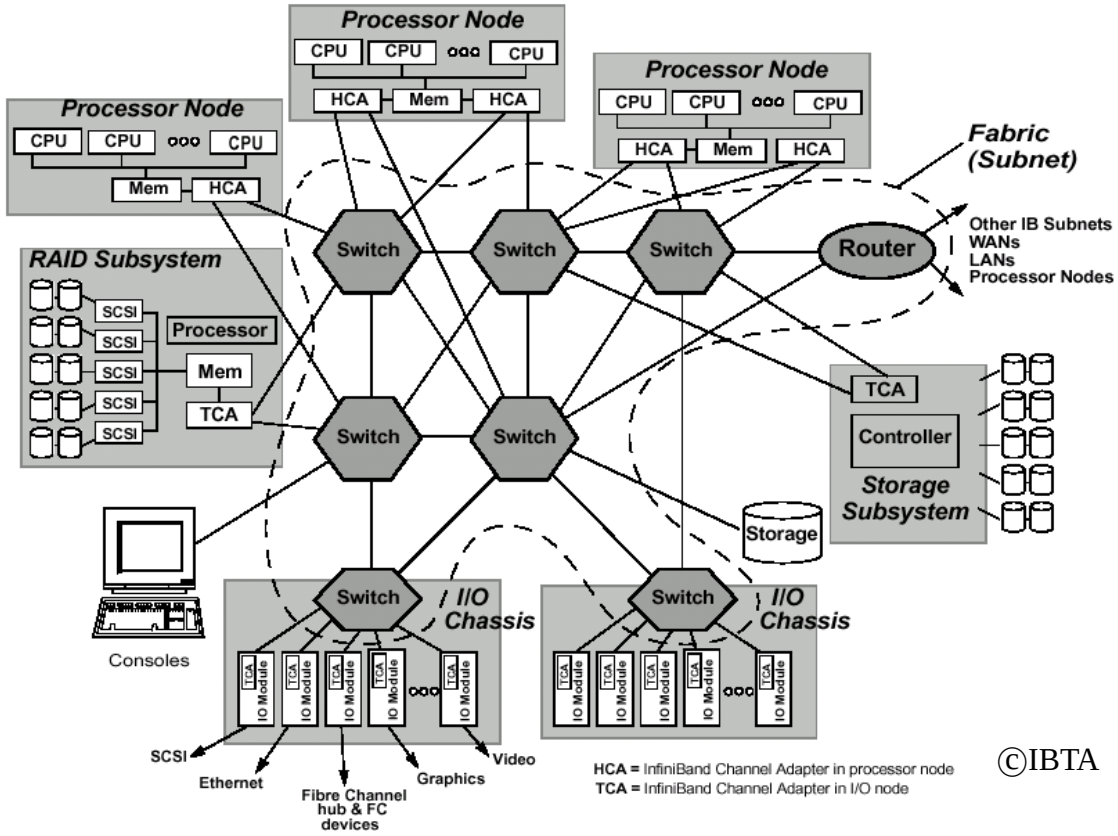

Figure 1: IBA System Area Network.

architecture is independent of the host operating system and processor platform.

IBA is designed around a switch-based interconnect technology with high-speed point-to-point links. An IBA network is divided into subnets interconnected by routers, each subnet consisting of one or more switches, processing nodes and I/O devices. Routing in IBA subnets is distributed, based on forwarding tables stored in each switch. IBA supports any topology defined by the user, including irregular ones, in order to provide flexibility and incremental expansion capability.

Processing nodes (either single processor or symmetric multiprocessor (SMPs)) are directly attached to a switch through a Host Channel Adapter (HCA). I/O devices can be attached to a switch through a Target Channel Adapter (TCA). While IBA Specification describes the behavior for HCA by IBA verbs, IBA does not specify the semantics of the consumer interface for a TCA. The IBA verbs are features that are defined to be available to host programs.

IBA links are bidirectional point-to-point communication channels, and may be either copper cable, optical fiber or printed circuit on a backplane. The signaling rate on the links is 2.5 GHz in the 1.0 release, the later releases possibly being faster. The physical links may be used in parallel to achieve greater bandwidth. Currently, IBA defines three link bit rates. The lowest one is 2.5 Gbps and is referred to as 1x (only one link at 2.5 GHz). Other link rates are 10 Gbps (referred to as 4x) and 30 Gbps (referred to as 12x) that correspond to 4-bit wide and 12-bit wide links, respectively. The width or widths that will be supported by a link is vendor-specific.

IBA switches route messages from their source to their destination based on forwarding tables that are programmed with forwarding information during initialization and network modification. The forwarding table can be linear, specifying an output port for each possible destination address up to a switch-specific limit, indexed by that address; or random, initialized by storing {destination, output port} pairs. The number of ports of a switch is vendor-specific, but is limited to 256 ports.

Switches can be cascaded to form large networks. Switches may also optionally support multicast routing.

Routing between different subnets (across routers) is done on the basis of a Global Identifier (GID) 128 bits long, modeled over IPv6 addresses. On the other hand, the addressing used by switches are Local Identifiers (LID) which allow 48K endnodes on a single subnet, the remaining 16K LID addresses being reserved for multicast.

Messages are segmented into packets for transmission on links and through switches. The packet size is such that after headers are considered, the Maximum Transfer Unit (MTU) of data may be 256 bytes, 1KB, 2KB or 4KB. Each packet, even those for unreliable datagrams, contains two separate CRCs, one covering data that cannot change, and another one covering data that changes in switches or routers, and it is recomputed.

The IBA transport mechanisms provide several types of communication services between endnodes. These types are connections or datagrams and both could be reliable (acknowledged) or unreliable. Obviously, for supporting QoS the applications must use reliable connections in order to be able to do resource allocation.

IBA management is defined in terms of managers and agents. While managers are active entities, agents are passive entities that respond to messages from managers. Every IBA subnet must contain a single master subnet manager, residing on an endnode or a switch that discovers and initializes the network.

The interested reader is referred to the InfiniBand Specifications [5] for more details on InfiniBand. Other interesting papers that are good summaries of the official specifications are [9, 2].

3 IBA support for QoS

In this section we are going to explain the mechanisms that IBA provides for supporting QoS. Basically, IBA has three mechanisms that permit QoS to be supported: service levels, virtual lanes, and virtual lane arbitration for transmission over links.

3.1 Service Levels

IBA defines a maximum of 16 service levels (SLs), but it does not specify what characteristics the traffic of each service level should have. Therefore, it depends on the implementation or the administrator how to distribute the different existing traffic types among the SLs.

By allowing the traffic to be segregated by category, we will be able to distinguish between packets from different SLs and to give them a different treatment based on their needs.

3.2 Virtual Lanes

IBA ports support virtual lanes (VLs), providing a mechanism for creating multiple virtual links within a single physical link. A VL represents a set of transmit and receive buffers in a port (Figure 2). IBA ports have to support a minimum of two and a maximum of 16 virtual lanes ($VL_0 \ldots VL_{15}$). All ports support VL_{15}, which is reserved exclusively for subnet management, and must always have priority over data traffic in the other VLs. The number of VLs used by a port is configured by the subnet manager. Since systems can be constructed with switches supporting different numbers of VLs, packets are marked with a Service Level (SL), and a relation between SL and VL is established at the input of each link by means of a SL to VL Mapping Table.

Each VL must be an independent resource for flow control purposes. When more than two lanes are implemented, the priorities of the data lanes are defined by VL arbitration tables.

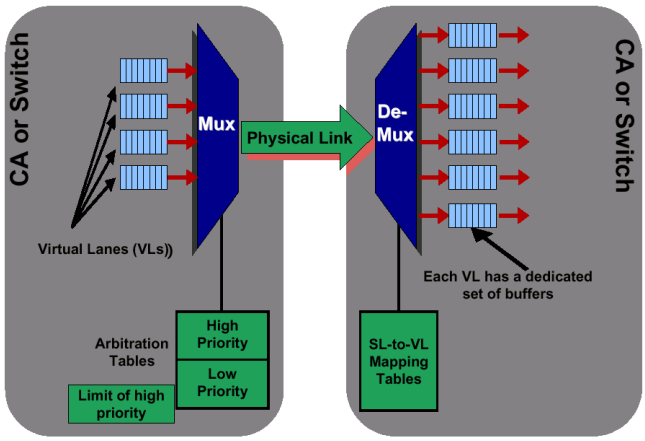

Figure 2: Operation of Virtual Lanes in a physical link.

3.3 Virtual Lane Arbitration

When more than two VLs are implemented, the priorities of the data lanes are defined by the VLArbitrationTable. This is for both HCA and TCA as well as for switches. This arbitration is only for data VLs, because VL_{15}, which transports control traffic, always has priority over any other VL.

The structure of the VLArbitrationTable is shown in Figure 3. Each VLArbitrationTable has two tables, one for delivering packets from high priority VLs and another one for low priority VLs. However, IBA does not specify what is high and low priority. The arbitration tables implement weighted round-robin arbitration within each priority level. Up to 64 table entries are cycled through, each one specifying a VL and a weight, which is the number of units of 64 bytes to be sent from that VL. This weight must be in the range of 0 to 255, and is always rounded up as a whole packet.

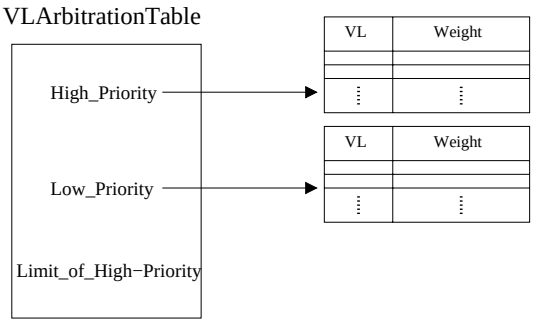

Figure 3: VLArbitrationTable structure.

A LimitOfHighPriority value specifies the maximum number of high priority packets that can be sent before a low priority packet is sent. More specifically, the VLs of the High_Priority table can transmit $LimitOfHighPriority \times 4096$ bytes before a packet from the Low_Priority table could be transmitted. If no high priority packets are ready for transmission at a given time, low priority packets can also be transmitted.

4 Our Proposal

In this section, we propose a classification of traffic and a way of filling in the VLArbitrationTable for each output port.

4.1 Traffic Classification

In [8], a traffic classification was presented. The following four categories were proposed:

- DBTS: Dedicated Bandwidth Time Sensitive traffic, which requires a given minimum bandwidth and must be delivered within a given latency in order for the data to be useful. Examples of such data streams include video conferencing and interactive audio.
- DB: Dedicated Bandwidth traffic, which requires a given minimum bandwidth but is not particularly sensitive to latency. This includes traffic that must have a bounded latency but where the actual value of the bound is not critical. An example of this class of traffic could be the playback of a video clip.
- BE: Best Effort. This traffic tends to be bursty in nature and largely insensitive to both bandwidth and latency. The majority of traffic in current networks is of this type, like file and printing services or web browsing.
- CH: Challenged. This traffic is intentionally degraded so as not to interfere with best effort traffic. An example of this traffic could be disk backup activities, which we do not want to interfere with other traffic.

We agree with this classification albeit with certain additions. For the DBTS and DB categories, we believe that a segregation of traffic should be made based on its characteristics. We have focused on the mean bandwidth requirements and have classified the connections into several classes according to their mean bandwidth requirements. Another possibility could be to consider the deadline required by DBTS connections besides mean bandwidth. But we have not analyzed this possibility in this paper.

The reason for proposing this classification is that if we put traffic with very different mean bandwidth requirements in the same VL, head-of-line blocking could be frequent [6]. By classifying connections into different categories, and by using appropriate values in the arbitration tables, we will be able to give a more adjusted treatment to each traffic class according to its needs.

Specifically, the categories that we distinguish for DBTS and DB are:

- Connections with very low bandwidth: connections with mean bandwidth requirements lower than or equal to 64 Kbps. An example of this class would be cellular phone conversations.
- Connections with low bandwidth: connections with mean bandwidth higher than 64 Kbps, but lower than or equal to 1.55 Mbps. An example of this class would be a video conference.
- Connections with high bandwidth: connections with mean bandwidth higher than 1.55 Mbps, but lower than or equal to 64 Mbps. An example of this class would be digital television transmission.
- Connections with very high bandwidth: connections with mean bandwidth higher than 64 Mbps. An example of this class would be high-definition television (HDTV) transmission.

Although we have made this classification on the basis of the mean bandwidth requirements alone, we will be able to satisfy the maximum latency deadlines by studying the maximum latency deadline requested during connection establishment and with a correct treatment of each traffic class in the arbitration for each output port.

InfiniBand distinguishes a maximum of 16 SLs and each port can have a maximum of 16 VLs. We can match the four previously established categories for DBTS and DB with InfiniBand SLs and assign a VL to each SL. Thus, we will use 4 VLs for the 4 DBTS SLs, another 4 VLs for the 4 DB SLs, one for BE traffic, one for CH traffic, and finally, one VL reserved by IBA for control traffic. Therefore, it would only be necessary for the ports to have 11 VLs. As 11 VLs are not allowed, 15 data VLs should be implemented. We can expand the previous classification into new categories in order to match the SLs and VLs until completing the total number of implemented VLs. Alternatively, these extra VLs can be used for other purposes [5].

If the ports do not have enough VLs implemented for this classification, we can still map several SLs to a single VL. The criteria for doing this have not been established yet.

4.2 Filling in the Virtual Lane Arbitration Table

Pelissier [8] proposed to use the table of high priority for DBTS traffic and the table of low priority for the rest of the traffic, but he did not specify how to compute the weights for the VLs. We agree with this proposal. Moreover, we propose a strategy to fill in the weights for the arbitration tables. For the sake of clarity, we will assume that the network only has DB traffic and we will see how to fill in the table of low priority when the table of high priority is empty.

The control unit of each switch will be responsible for locally accepting or rejecting the connection requests based on the available local information. This information includes the state of the output links and how much bandwidth they have already reserved. Every time a switch accepts a connection, it will modify the arbitration table to adjust it to the traffic requirements. Note that this strategy agrees with InfiniBand specifications, which indicate that the local agent will be able to do these operations.

Since we assumed that we do not have DBTS traffic, the table of high priority will be empty. As DB traffic requires guaranteed bandwidth, we can split time into frames of fixed duration. Each connection will be assigned a certain number of slots per frame, according to its bandwidth requirements.

According to the InfiniBand specifications, each arbitration table can have a maximum of 64 entries, each one with a maximum weight of 255. Assuming that each slot corresponds to the transmission of 64 bytes, which corresponds to a weight of one in a table entry (according to InfiniBand specifications), the maximum number of slots per frame is 64×255. We are going to compute the weights for each entry of the table with respect to this number. As we will see, this will drastically simplify the computation of table entries when connections are dynamically established.

Note that the important thing is neither the reference point nor the particular value of the weight for a certain table entry but the ratio between weights, which should agree with the ratio between the mean bandwidth requirements of the accepted connections. Therefore, we will compute the weight that should be assigned to a table entry according to the mean bandwidth of the connections serviced by that entry.

However, it should be noted that the InfiniBand arbitration scheme will assign all the bandwidth to the set of established connections, even if these connections requested a lower overall bandwidth. The reason is that once all the virtual channels in the corresponding table of priority have been serviced, the link arbiter will repeat the cycle again. Therefore, all the link bandwidth

will be consumed (assuming that there are packets ready to be transmitted) regardless of the values in the tables of priority. Thus, in order to simplify the computation of the values in the tables of priority, we decided to compute these values as if all the slots were in use (with respect to 64×255 slots per frame, as mentioned above). By doing so, the entries in the tables of priority do not need to be recomputed when a new connection is established because all of them have been computed with respect to the same reference. Note that each connection will receive more bandwidth than requested unless the full link bandwidth has been reserved. But this will happen anyway, regardless of how the values in the entries are computed. Therefore, our methodology has the significant advantage of not requiring recomputing the table entries for previously established connections when a new connection is established, while providing the same QoS guarantees as other methodologies.

As we only have 64 entries in each arbitration table, this could limit the number of connections that can be accepted. A connection requiring very high bandwidth could also need slots in more than one table entry. For this reason, we propose grouping the connections with the same SL into a single table entry until completing the maximum weight for that entry, then moving to another free entry. This way, the number of table entries is not a limitation for the acceptance of new connections, but only the available bandwidth.

The weight must therefore be computed according to the total mean bandwidth requested by the connections serviced by the corresponding table entry, and every time a new connection is accepted, only the table entries used by this connection need to be recomputed. Let us now compute the weight corresponding to a certain table entry.

We know that each frame consists of $64 \times 255 = 16320$ slots of 64 bytes. For the different link rates of InfiniBand we can compute its T_{frame}, that is, the time it would take to transmit a frame assuming that all the table entries have been assigned their maximum value. The values of this time for each link rate can be seen in Table 1.

Speed	Link Rate (Gbps)	T_{frame} (msec)
1x	2.5	3.342336
4x	10	0.835584
12x	30	0.278528

Table 1: Time to send an entire frame for each InfiniBand link rate.

The bandwidth B assigned to a connection is the number of slots assigned to that connection divided by the total number of slots in the frame and multiplied by the link bandwidth, that is, the percentage of the frame assigned to that connection multiplied by the link bandwidth.

$$B = \frac{N_{Slots}}{N_{Slots\ per\ frame}} \times B_{link}$$

So, the number of slots for a connection with mean bandwidth B bps is

$$N_{Slots} = \left\lceil \frac{B}{B_{link}} \times 64 \times 255 \right\rceil$$

where $\lceil\ \rceil$ means to round up (ceiling function). We can see T_{Frame} as

$$T_{Frame} = \frac{64\ entries \times 255\ \frac{units}{entry} \times 64\ \frac{bytes}{unit} \times 8\ \frac{bits}{byte}}{B_{link}}$$

This way, we can compute the number of slots as

$$N_{Slots} = \left\lceil B \times \frac{T_{frame}}{512} \right\rceil$$

In order to guarantee that some bandwidth will be available for best-effort traffic, it is possible to reserve some slots for this traffic class. The amount of reserved bandwidth will depend on system requirements. For the evaluation results presented in this paper, we have assumed that 20% of the bandwidth is reserved for BE traffic. Therefore, when the network starts to establish DB connections, the entries corresponding to this BE traffic will have already been created in the table of low priority. However, for the CH traffic, no bandwidth is reserved, and we only set an entry in the table of low priority with weight equal to one. Therefore, this traffic will only be able to take out one packet in each frame. If we do not set any entry for CH traffic, the packets of this type could never be transmitted, so we must assign an entry to this traffic class, but with the minimum weight. Note that when no other traffic is available, CH traffic will be selected for packet transmission.

When DBTS or DB traffic do not use all their assigned bandwidth, this bandwidth will be used by BE and CH traffic (if there is traffic of these categories available), regardless of their values in the arbitration tables. Note that the values in the arbitration table for BE and CH traffic guarantee a minimum bandwidth, but do not limit the bandwidth available to these categories when no other traffic is available. This is a feature of the weighted round robin arbitration policy.

5 Performance Evaluation

In this section, we evaluate the behavior of our proposal. Instead of analytic modeling, simulation was used. We have developed a flit-level simulator that models the InfiniBand behavior. First, we will explain the network and the traffic models that we have used.

5.1 Network Model

We have used both regular and irregular networks. Regular networks have been chosen with hypercube and mesh topology. Irregular ones have been randomly generated, but always having four ports for interconnection between switches. In all the cases, each switch has 4 ports with one host attached to each of them [11].

To be able to test our proposal, each port has at least 11 VLs, both on the switches and on the hosts. One of them is exclusively dedicated to control traffic and the rest to transmit user data. Data VLs will be used according to the traffic classification seen in section 4.1.

Each VL has its own buffer, both at the input and the output of a switch. The size of these buffers has been assumed to be proportional to the packet sizes we have tried: 256 and 4096 bytes. For these packet sizes, the buffer sizes we have used are 1056 and 16416 bytes, respectively. Therefore, these buffer sizes allow 4 entire packets to be stored.

With respect to network size, we have evaluated networks with sizes ranging from 8 to 64 switches (with 32 to 256 hosts, respectively) and, for all cases, the results show the same tendency. Due to space limitation, we will only include here the results for the networks with 16 switches. We have tried the three link rates of InfiniBand: 1x, 4x and 12x, without any variation in the results.

The crossbar incorporated in the switches is multiplexed among VLs, having the same number of inputs as input ports in the switch. Therefore, for the networks analyzed in this paper, with 8-port switches, all the switches have crossbars of size 8×8. To prevent the crossbar from becoming a bottleneck, the crossbar ports work twice as fast as the links so that they are able to take out

everything that arrives [12]. Thus, for the link rate of 2.5 Gbps, the crossbar ports work at 5 Gbps. Finally, we have assumed that the crossbar has a datapath width of 16 bits which, with the link rates we used, implies that the clock cycle is 3.2 nsec.

The arbitration for an output port introduces some delay. We have assumed that this arbitration, with current technology, takes about 20 nsec, approximately 7 clock cycles.

5.2 Traffic Model

We have used CBR traffic of type DB. In this first approach, we have not considered DBTS traffic, which will be the subject of future research. Although the results for BE and CH traffic are not the focus of this paper, we have reserved slots for these traffic classes in the tables.

CBR traffic is randomly generated among the 4 SLs considered in the subsection 4.1, and requested bandwidth is uniformly distributed across the range for each SL. Specifically, for the four categories defined there, we will establish connections with the following ranges:

- SL 0: connections with very low bandwidth that will have between 8 Kbps and 64 Kbps.
- SL 1: connections with low bandwidth that will have between 64 Kbps and 1.55 Mbps.
- SL 2: connections with high bandwidth that will have between 64 Kbps and 64 Mbps.
- SL 3: connections with very high bandwidth that will have between 64 Mbps and 300 Mbps.

We can observe that the bandwidth range for each SL is very wide, and therefore, packets with very different characteristics are going to coexist in the same buffers and in the same SL.

We will increase the traffic in the network by increasing the number of established connections. These connections are equally distributed among the different SLs defined. Therefore, when the connections have been established, there is approximately the same number of connections for each SL. When establishing a certain number of connections, new connections are set until the required number of established connections is reached. Evidently, when we try to establish many connections, it is a more involved task to establish those connections requiring greater bandwidth. If after a certain number of retries it is not possible to establish more connections for some service level, the establishment period ends. This way, there will be more or less the same number of connections for each SL.

This way of establishing connections has the drawback that there may still be room for more connections of smaller bandwidth in the network. But we will see in the next section that we can achieve quite high loads, thus being able to compare between them, having the same number of connections for each SL. In future, we would also like to vary this aspect.

As already mentioned, packets with 256 and 4096 bytes have been considered. These sizes correspond to minimum and maximum packet size allowable in InfiniBand. The packet headers for InfiniBand are included in these quantities. Thus, for packet size equal to 256 bytes, the header represents a larger overhead for the network, but being smaller also has the advantage that the link takes up less time to transmit a packet.

Once we have established the required number of connections, or the maximum as the case may be, the period of connections establishment finishes and a transient period begins. This transient period lasts until 10000 packets have arrived at their destinations. Once the transient period finishes, the steady state period begins, where we will gather the data to be shown in the next section. The steady state period continues until the connection with a smaller mean bandwidth has received 100 packets.

5.3 Simulation Results

In this section, we show the performance evaluation results. For simulations, we have reserved 20% of link bandwidth for BE traffic. So, for link speed of 2.5 Gbps, the maximum bandwidth available for CBR traffic is 2 Gbps. For each topology tested, we can see in Table 2 the number of established connections for each packet size. Note that different topologies are able to accept a different number of connections but the values for all of them are similar.

	Topology					
	Irregular		Hypercube		Mesh	
Tried	Packet 256	Packet 4096	Packet 256	Packet 4096	Packet 256	Packet 4096
256	256	256	256	256	256	256
512	512	512	512	512	512	512
1024	1024	1024	1024	1024	1024	1024
1536	1536	1536	1536	1536	1536	1536
2048	2048	2048	2048	2048	2048	2048
2560	2560	2560	2560	2560	2518	2560
2816	2623	2631	2598	2679	2518	2598
3072	2623	2631	2598	2679	2518	2598

Table 2: Number of established connections for each network and packet size (in bytes) for different number of tried connections.

Figure 4 shows the accepted load (delivered traffic) for all the topologies and packet sizes. Note that the behavior is similar for all the networks and for both packet sizes. It can be observed that the plots for delivered traffic do not flatten when approaching network saturation. The reason is that only accepted connections are allowed to transmit packets. What flattens is the number of established connections, as shown in Table 2. Moreover, with our proposal, the network can reach a throughput close to 80%, which is the maximum available for this kind of traffic, because the other 20% is reserved for BE traffic. Also, note that for small packet size the network reaches a slightly higher throughput (0.769 vs 0.757 bytes/cycle/host). This is due to the fact that the overhead introduced by packet headers is more important for small packet size and more packets must be sent. Note that connections have been accepted until the payload approaches 80%.

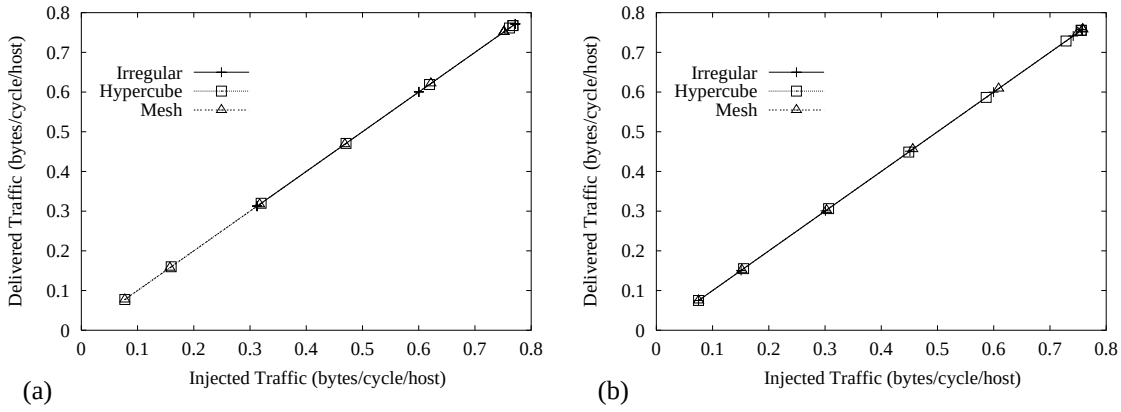

Figure 4: Delivered traffic for irregular and regular networks with (a) small packet size (256 bytes) and (b) large packet size (4096 bytes)

We have also studied how workload is distributed across the host interfaces and switch ports

when the network has a heavy workload. Figure 5 displays the utilizations of host interfaces and switch ports for the irregular network with the maximum number of supported connections. In these figures we plot the utilization for the two packet sizes in two ways: In increasing order of host interface and switch port identifier, and in increasing order of utilization.

For the hypercube network, the plots for host interface and switch port utilization are shown in Figure 6 and for the mesh network in Figure 7. The average utilizations for host interfaces and switch ports are drawn as horizontal lines, and they are very close to the maximum of 80%. Some utilizations exceed 80%, especially for small packets. This is due to the overhead introduced by packet headers. But the important observation is that utilization is very close to the maximum value for almost all the interfaces and ports, thus making a very efficient use of available resources.

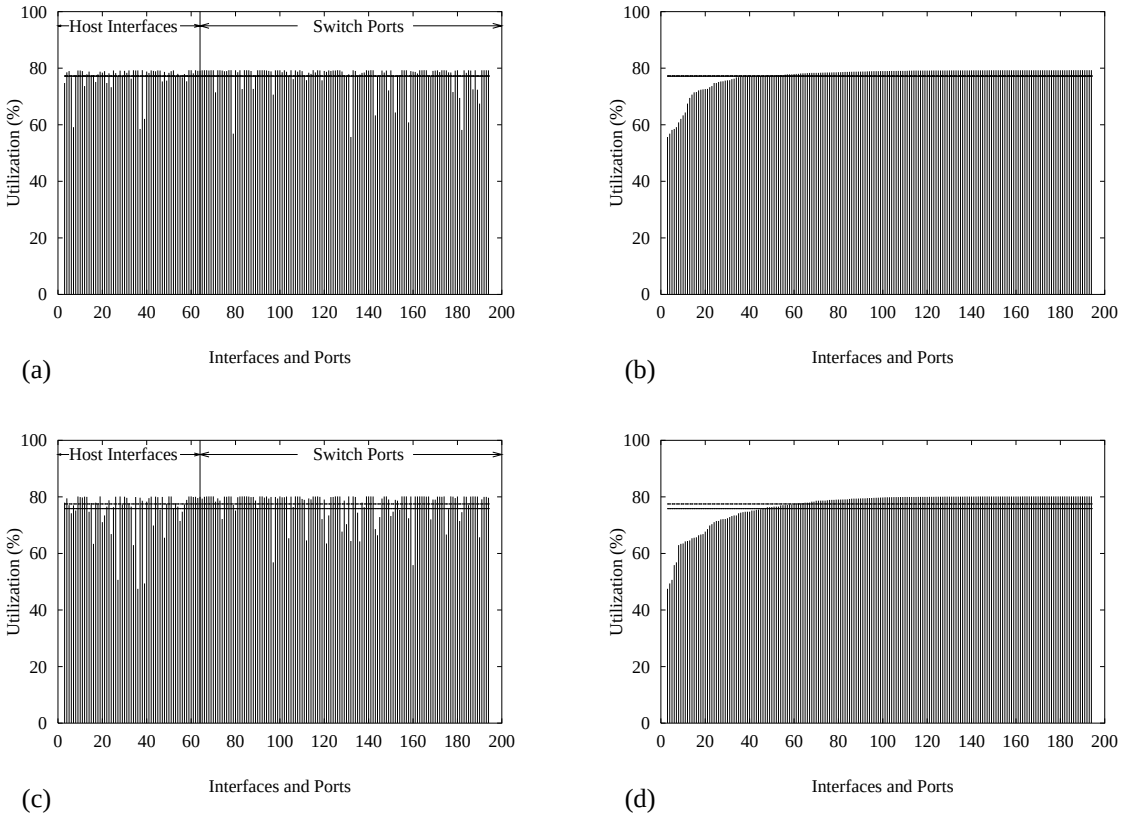

Figure 5: Utilization of switch ports and host interfaces for irregular networks with the maximum number of supported connections for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

In order to get more information about the distribution of packet delay, the percentage of packets whose delays are lower than a set of thresholds have been computed. These percentages have been obtained for the maximum number of accepted connections. These results give an idea of the percentage of packets that will meet a given deadline, and therefore, they measure the provided QoS. The thresholds are different for each type of connection, and they are related to their inter-arrival time (IAT). Therefore, thresholds become tighter as bandwidth requirements increase. The results are presented in Figure 8 for each SL, and for both packet sizes. We can observe that the results are similar for all the networks and packet sizes. In all cases, the packets for very low and low bandwidth connections (SL 0 and SL 1) arrive before their $\frac{IAT}{32}$. For high and very high bandwidth connections (SL 2 and SL 3), this value is too small and packets need some more time, but

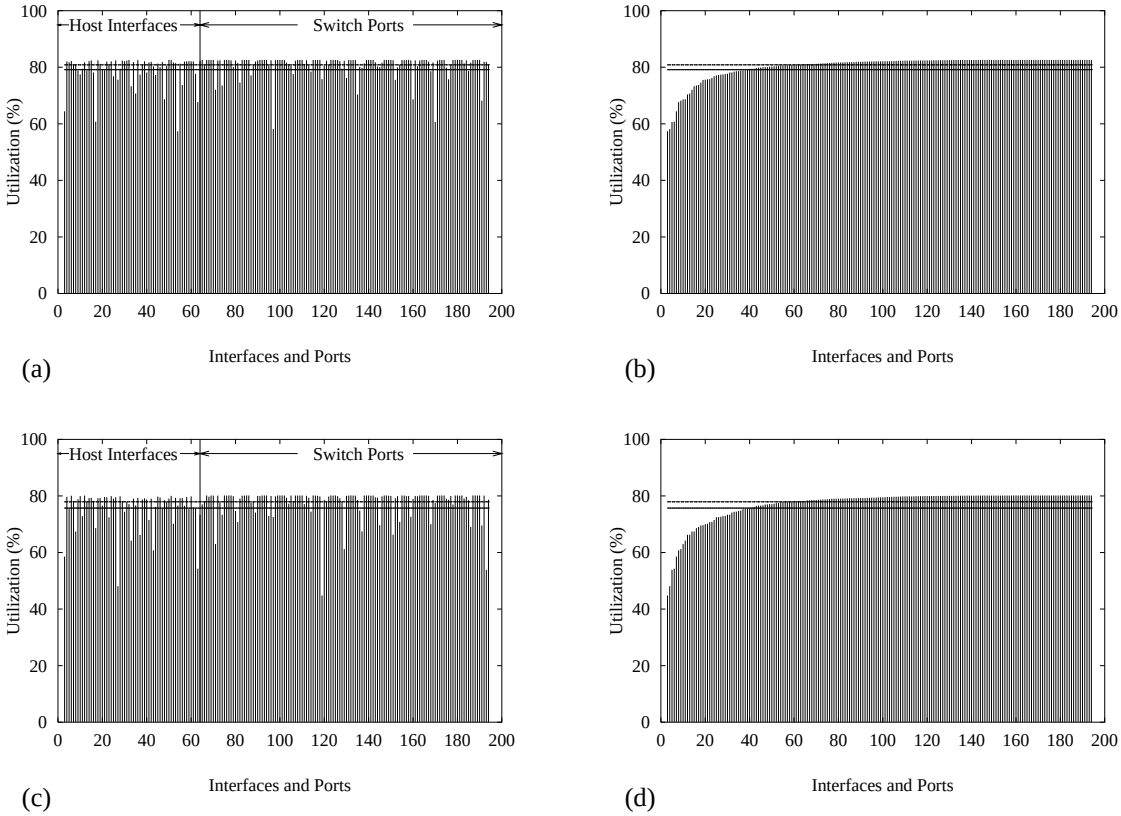

Figure 6: Utilization of switch ports and host interfaces for hypercube network with the maximum number of supported connections for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

in both cases all packets have arrived before a deadline equal to $\frac{3 \times IAT}{4}$. This is a very good results, which shows that our methodology to compute InfiniBand arbitration tables is very effective and provides good QoS support.

Average packet jitter has also been measured. We have established several intervals to compute the jitter and bar graphs plot the percentage of packets received within each interval. These intervals are different for each type of connection, and are related to their IAT. The results are shown for each network and packet size in Figure 9. In all cases, we can observe that packets from SL 0 and SL 1 always arrive with jitter almost equal to zero (in the interval $[\frac{-IAT}{8}, \frac{IAT}{8}]$). For the rest of networks, results are similar.

Finally, we have studied the influence of mixing connections with very different mean bandwidth on the same service level (and therefore, on the same virtual lane). Given a deadline, we have selected the connections that have delivered the lowest and the highest percentage of packets before that deadline. We have selected a very tight deadline so that the percentage of packets meeting the deadline was lower than 100%. In particular we have selected a deadline equal to $\frac{IAT}{2}$. In the Figures these connections will be referred to as the worst and the best connections, respectively. We can see these results for irregular, hypercube and mesh networks in Figures 10, 11 and 12, respectively. The results do not vary significantly, having a similar behavior for all connections within the same service level. Note that even the worst connections for each service level have a deadline lower than their IAT. Therefore, our methodology to compute the weights for the table of priority is very robust, providing good QoS guarantees to all the connections, despite the fact that

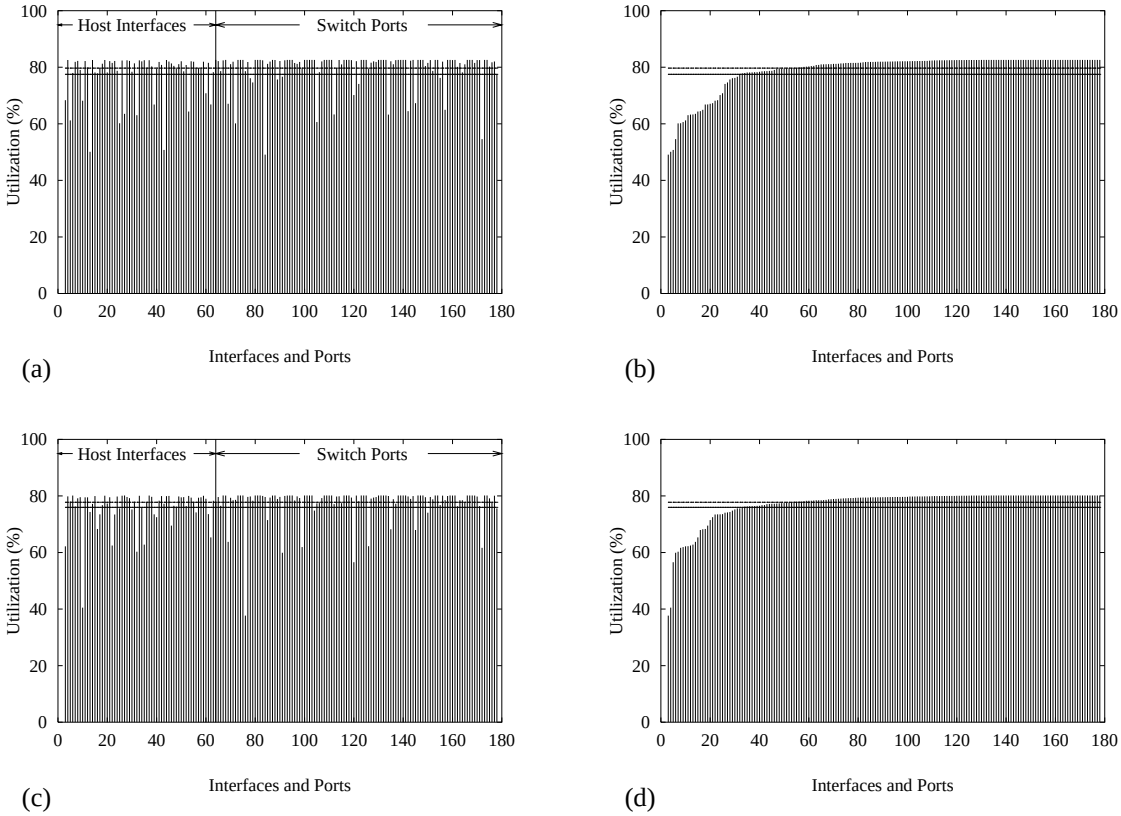

Figure 7: Utilization of switch ports and host interfaces for mesh network with the maximum number of supported connections for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

connections with very different bandwidth requirements share the same resources (virtual lanes).

In Appendix A we have included the results corresponding to 4x (10 Gbps) IBA link rate. As it can be observed, the results are similar to the obtained for 1x IBA link rate. In this appendix, we have also shown the results corresponding to 12x (30 Gbps) IBA link rate. The same tendency can be observed.

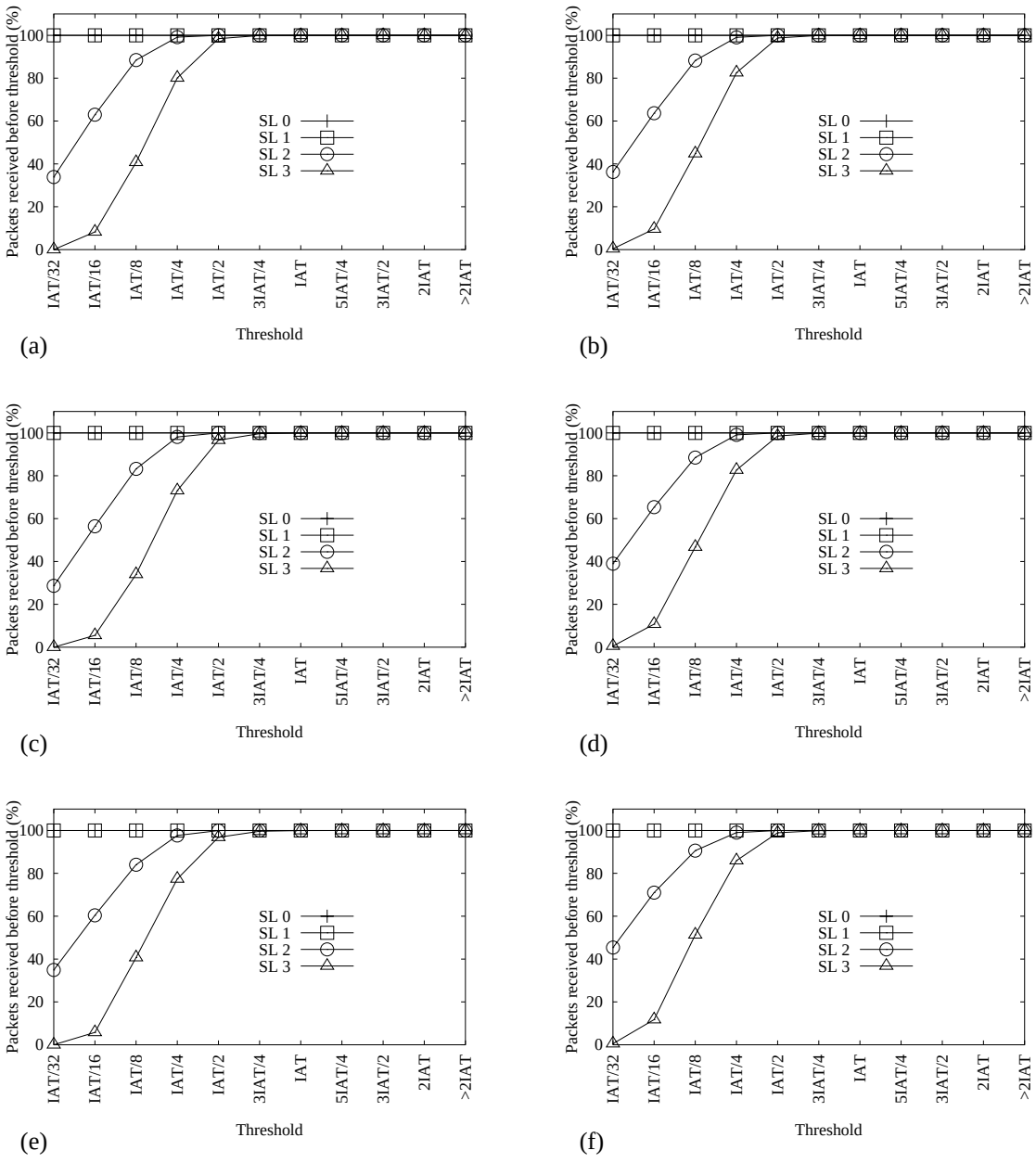

Figure 8: Distribution of packet delay for the maximum number of accepted connections considering (a), (c) and (e) small packet size and (b), (d), (f) large packet size. (a) and (b) show the results for an irregular network, (c) and (d) for a hypercube network and (e) and (f) for a mesh.

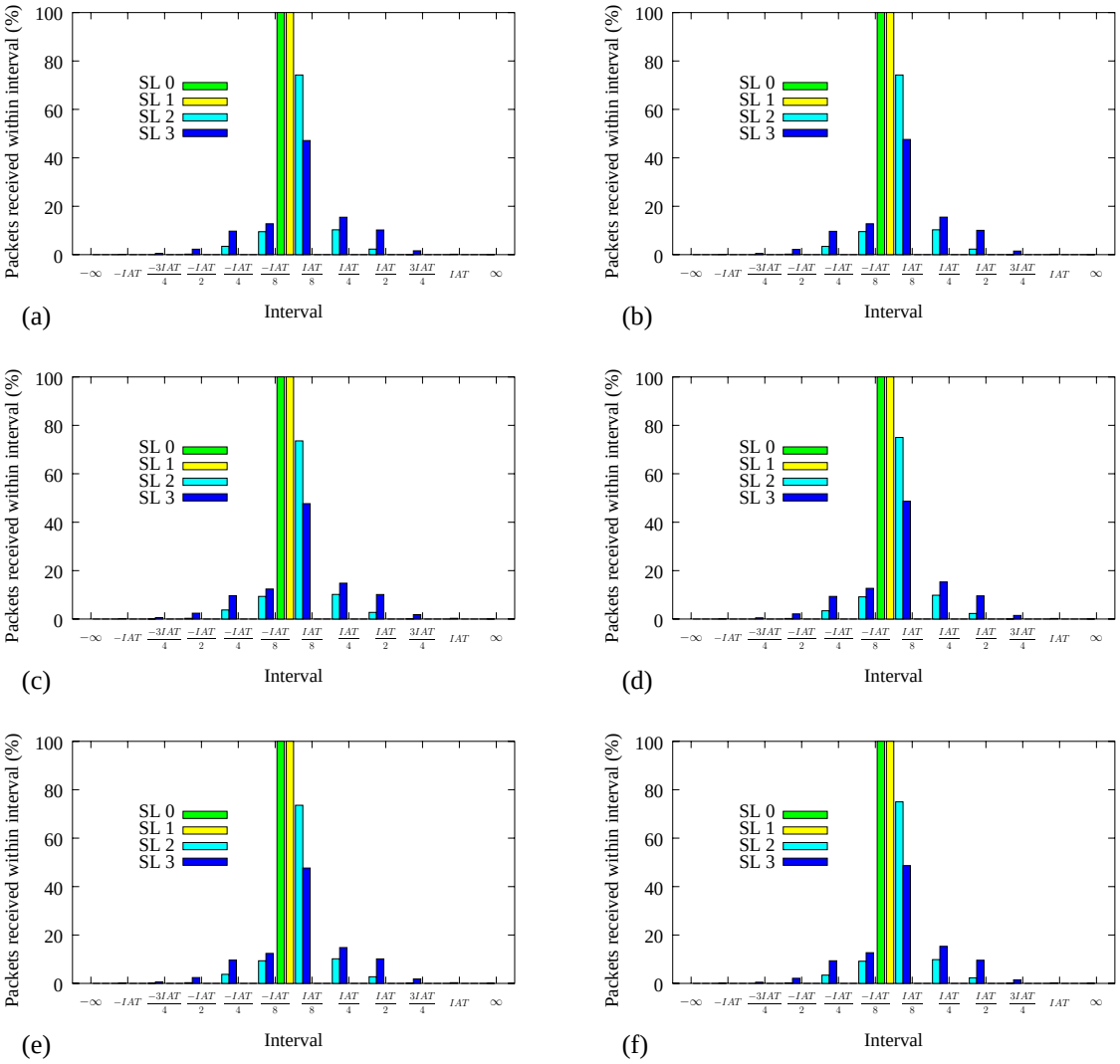

Figure 9: Average packet jitter for (a), (c) and (e) small packet size and (b), (d), (f) large packet size. (a) and (b) are for irregular network, (c) and (d) for hypercube and (e) and (f) for mesh.

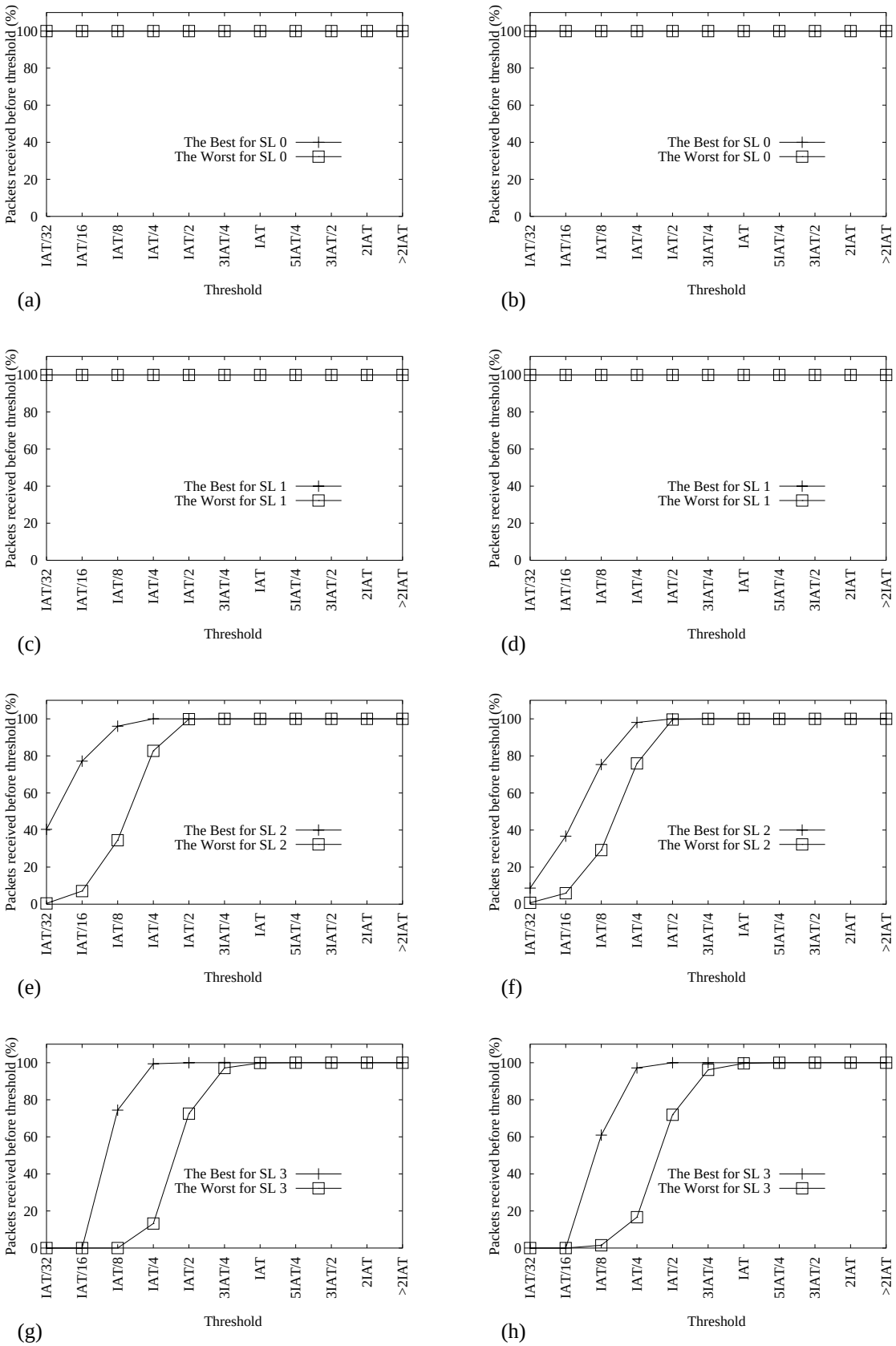

Figure 10: Behavior of the best and the worst connection for irregular network for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.

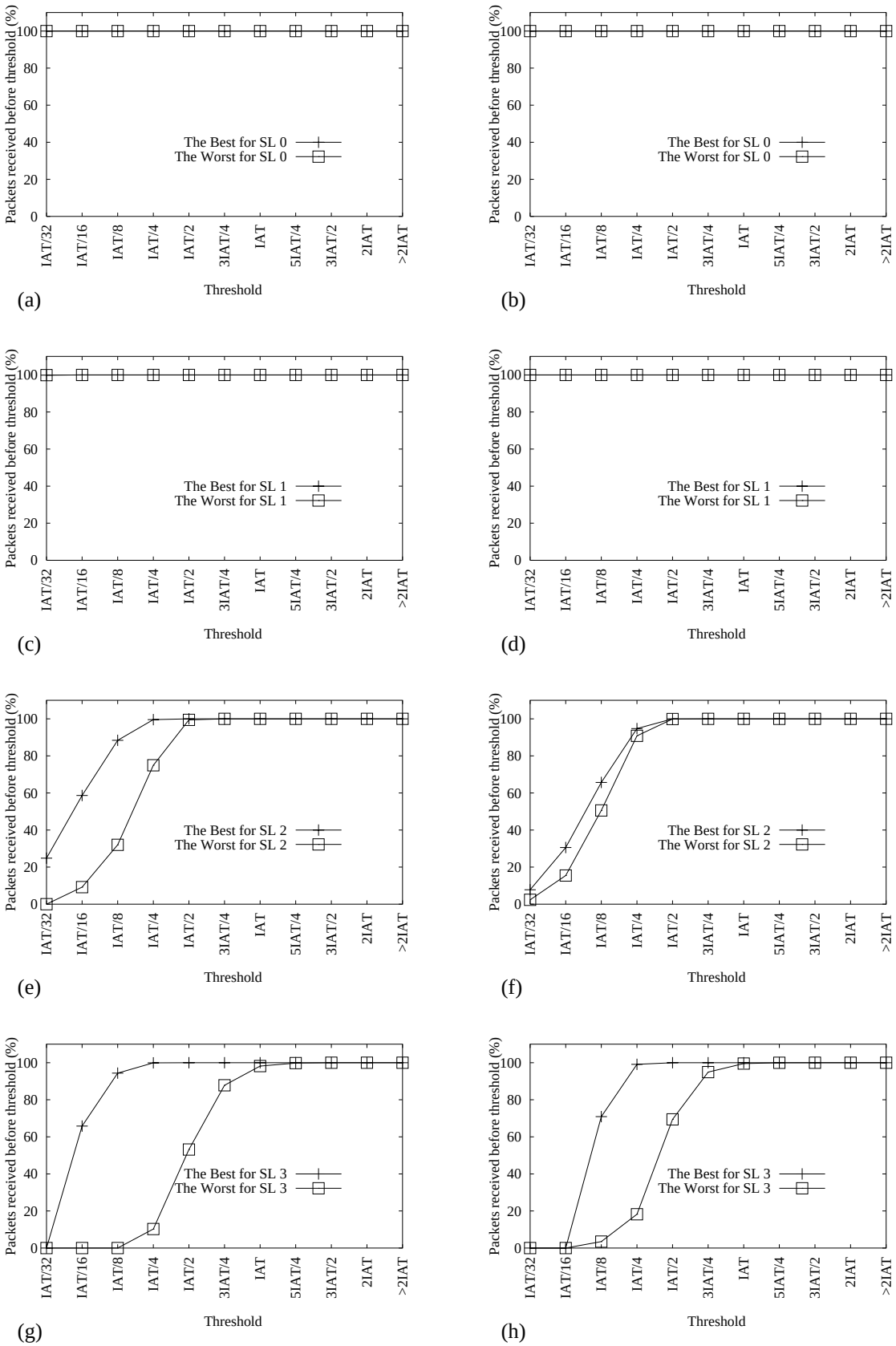

Figure 11: Behavior of the best and the worst connection for hypercube network for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.

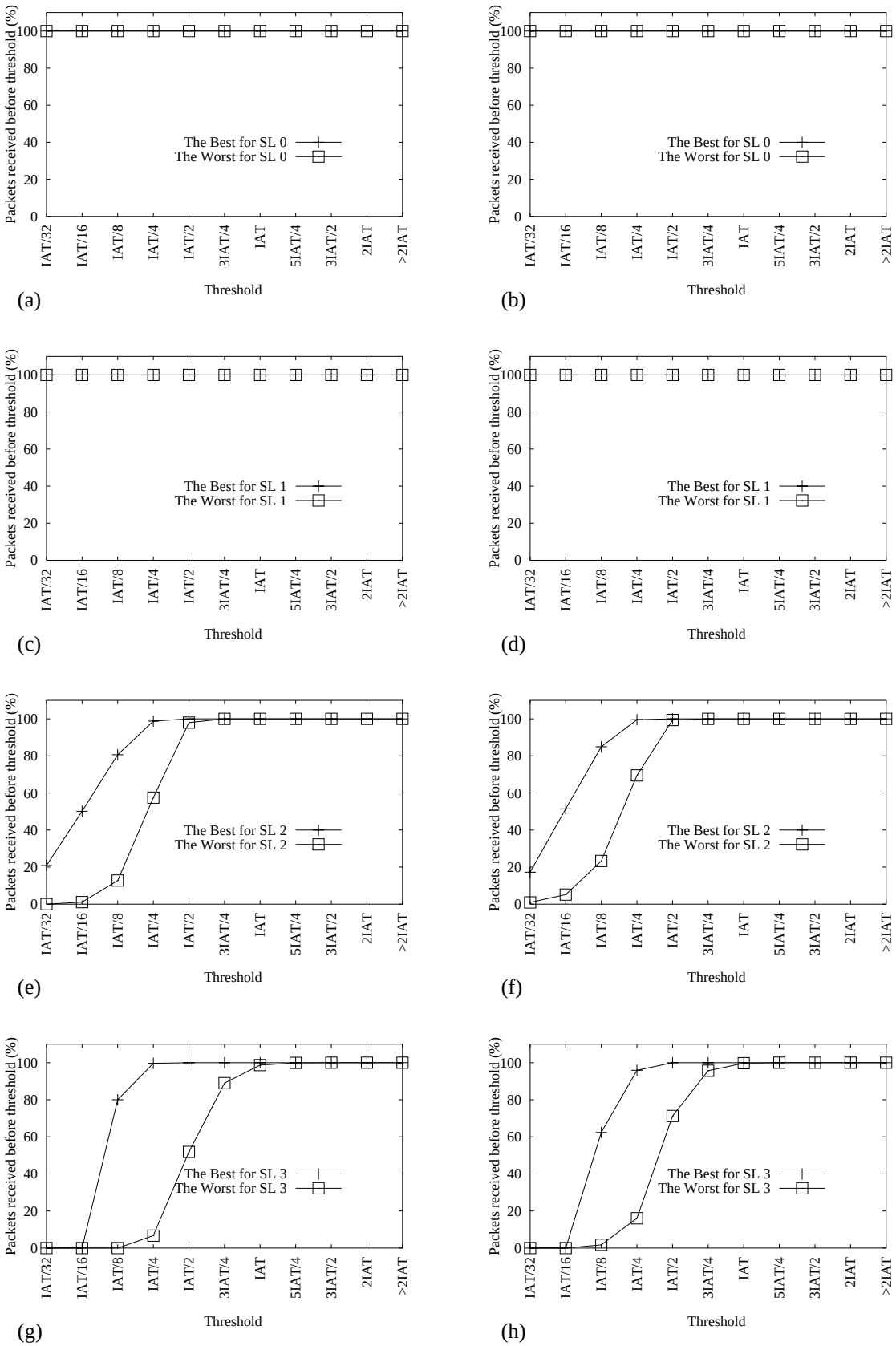

Figure 12: Behavior of the best and the worst connection for mesh network for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.

6 Conclusions

We have proposed a methodology to compute the virtual lane arbitration tables for InfiniBand. Our proposal has the significant advantage of not requiring recomputing the table entries for previously established connections when a new connection is established. This is because all the entries have been computed with respect to the same reference.

We have tested the behavior of the proposed methodology by simulation using CBR traffic with different mean bandwidth requirements. We have seen that, the maximum utilization that can be reached is almost the maximum available for both small and large packet sizes, regardless of the network topology. We have also shown that all types of traffic can satisfy their QoS requirements. In particular, we have evaluated the percentage of packets that arrive before a certain deadline. Even for service levels (SLs) with the lowest interarrival time (IAT), almost all the packets arrive before their $\frac{IAT}{2}$ or soon after. We have also shown that all the connections sharing the same SL and virtual lane (VL) have very similar behavior, despite the fact they have very different mean bandwidth requirements. In particular, we have seen that the best and the worst connection have a similar behavior.

We have also evaluated jitter, showing that all packets arrive with jitter between $\frac{-IAT}{8}$ and $\frac{IAT}{8}$ for connections with low bandwidth requirements. For connections with high bandwidth requirements the jitter has a Gaussian distribution with all the packets within the interval $[-IAT, IAT]$.

We are currently examining a number of extensions to this work. First, we are testing our proposal for traffic that has a latency deadline. Second, we are studying different classifications for the case when we have fewer VLs than SLs. Third, we are exploring other alternative designs with the goal to improving the QoS support. Finally, we would like to test our proposal in an InfiniBand commercial product as soon as there is one available, in order to verify the practicality of our model.

References

[1] S. Blake, D. Back, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. *RFC 2475*, Dec. 1998.

[2] P. by ISSG Technology Communications. Infiniband architectural technology. Technical Report TC000702TB, Compaq Computer Corporation, July 2000.

[3] A. Forum. *ATM Forum traffic management specification. Version 4.0*, May 1995.

[4] N. Giroux and S. Ganti. *Quality of Service in ATM Networks*. Prentice Hall, 1999.

[5] InfiniBand Trade Association. *InfiniBand Architecture Specification Volume 1. Release 1.0*, Oct. 2000.

[6] M. J. Karol, M. G. Hluchyj, and S. P. Morgan. Input versus output queueing on a space-division packet switch. *IEEE trans. on commun.*, COM-35:1347–1356, 1987.

[7] $InfiniBand^{TM}\ Trade\ Association$. *http://infinibandta.com*.

[8] J. Pelissier. Providing quality of service over Infiniband architecture fabrics. In *Proceedings of the 8th Symposium on Hot Interconnects*, Aug. 2000.

[9] G. Pfister. *High Performance Mass Storage and Parallel I/O*, chapter 42: An Introduction to the InfiniBand Architecture, pages 617–632. IEEE Press and Wiley Press, 2001.

[10] M. Schwartz and D. Beaumount. Quality of service requirements for audio-visual multimedia services. *ATM Forum*, ATM94-0640, July 1994.

[11] F. Silla and J. Duato. Improving the efficiency of adaptive routing in networks with irregular topology. In *Proceedings of the 1997 Int. Conference on High Performance Computing*, Dec. 1997.

[12] A. Systems. Avici Terabit Switch Router. http://www.avici.com/.

Appendix A

In Table 3 and in Figures from 13 to 21, we have included the results corresponding to 4x (10 Gbps) IBA link rate. As it can be observed, the results are similar to the obtained for 1x IBA link rate. Besides, Table 4 and Figures from 22 to 30 show the results corresponding to 12x (30 Gbps) IBA link rate. The same tendency can be observed.

	Topology					
	Irregular		Hypercube		Mesh	
Tried	Packet 256	Packet 4096	Packet 256	Packet 4096	Packet 256	Packet 4096
1500	1500	1500	1500	1500	1500	1500
3000	3000	3000	3000	3000	3000	3000
4500	4500	4500	4500	4500	4500	4500
6000	6000	6000	6000	6000	6000	6000
7500	7500	7500	7500	7500	7500	7500
9000	9000	9000	9000	9000	9000	9000
10500	10500	10500	10500	10500	10500	10500
12000	11005	11103	11024	11134	10968	11064

Table 3: Number of established connections for each network and packet size (in bytes) for different number of tried connections for 4x (10 Gbps) IBA link rate.

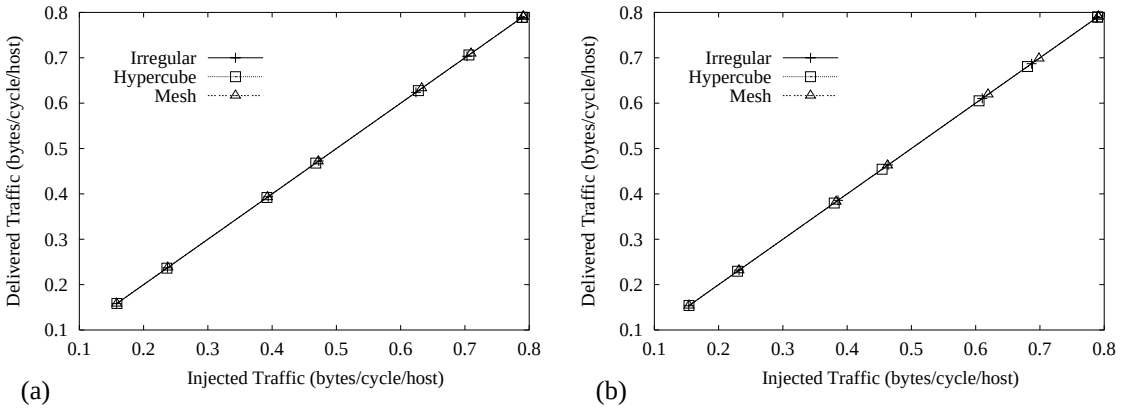

Figure 13: Delivered traffic for irregular and regular networks with (a) small packet size (256 bytes) and (b) large packet size (4096 bytes) for 4x (10 Gbps) IBA link rate

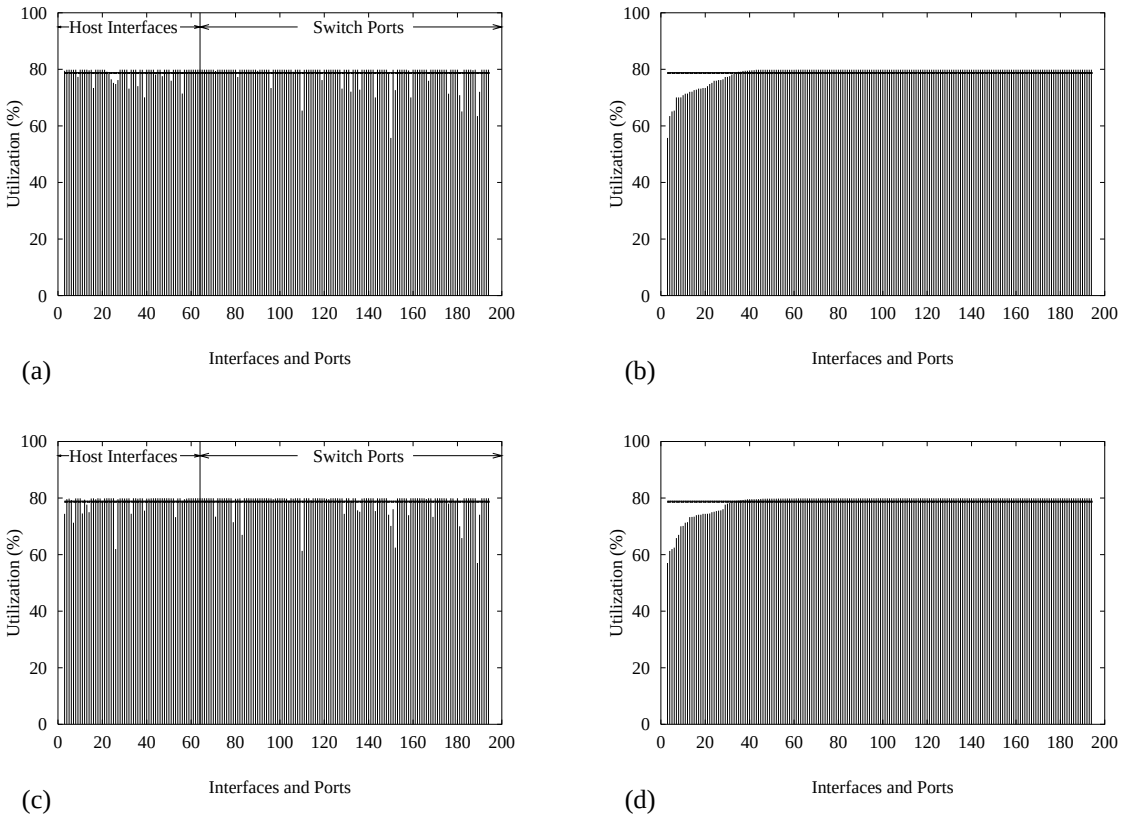

Figure 14: Utilization of switch ports and host interfaces for irregular networks with the maximum number of supported connections for 4x (10 Gbps) IBA link rate for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

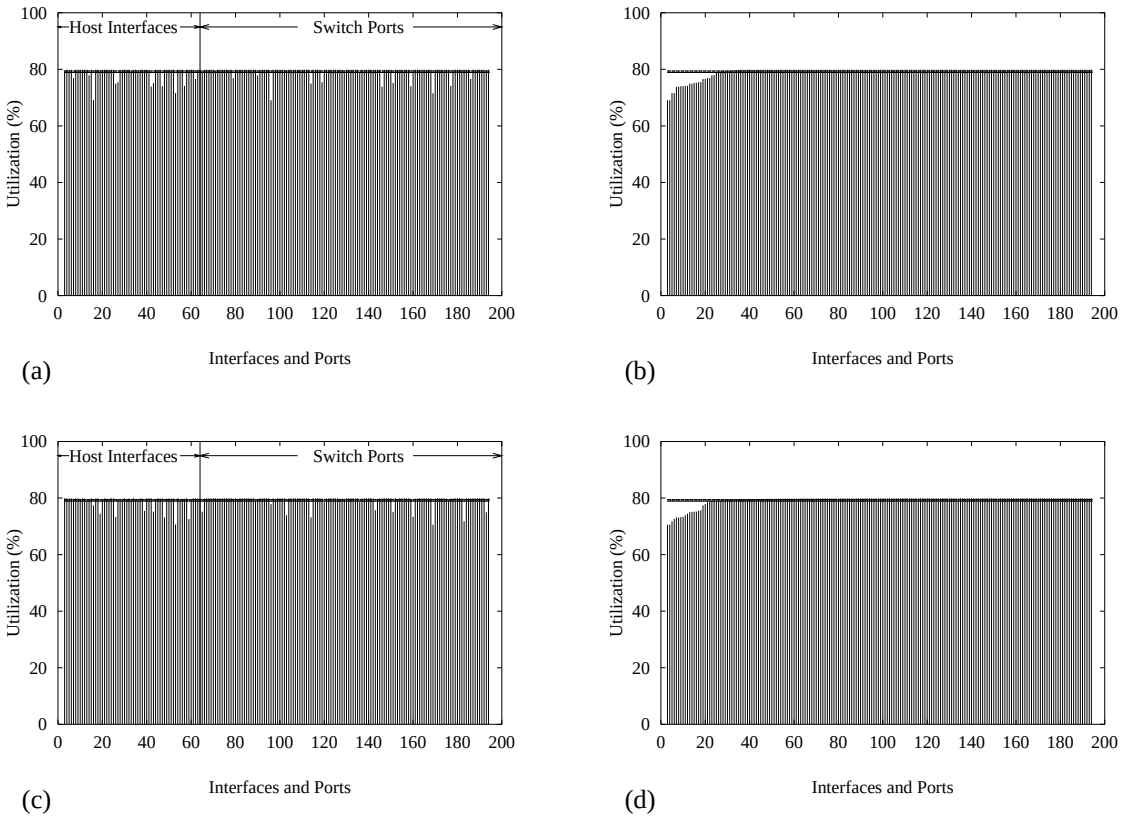

Figure 15: Utilization of switch ports and host interfaces for hypercube network with the maximum number of supported connections for 4x (10 Gbps) IBA link rate for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

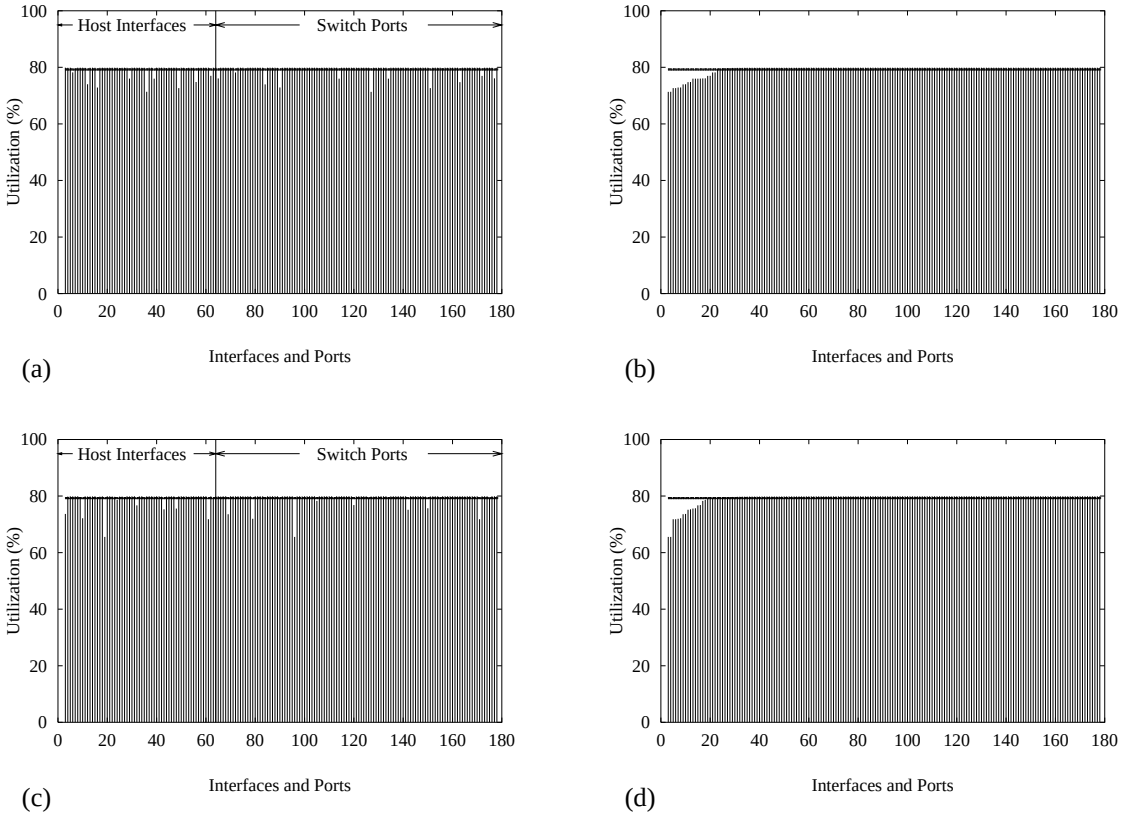

Figure 16: Utilization of switch ports and host interfaces for mesh network with the maximum number of supported connections for 4x (10 Gbps) IBA link rate for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

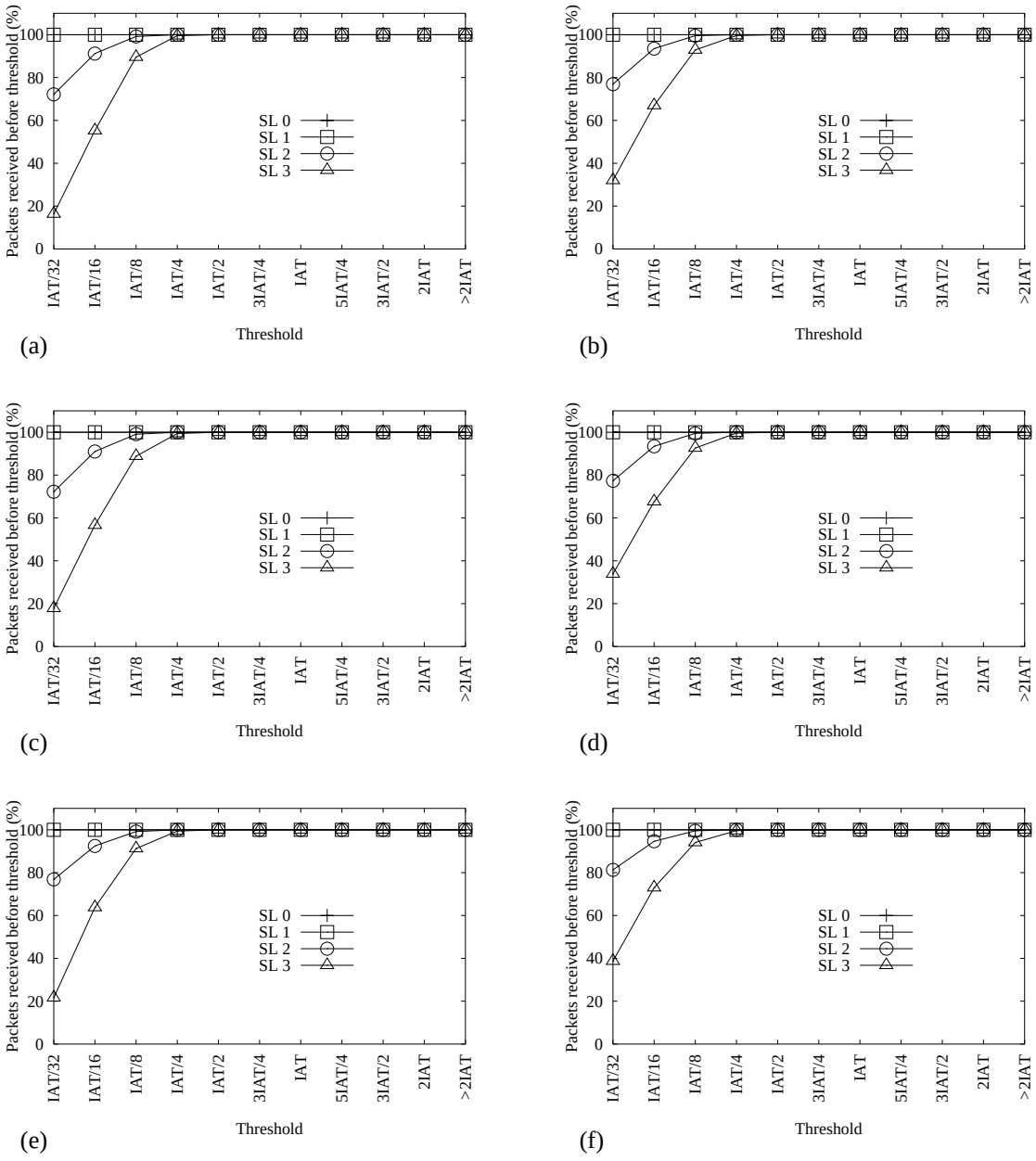

Figure 17: Distribution of packet delay for the maximum number of accepted connections for 4x (10 Gbps) IBA link rate, considering (a), (c) and (e) small packet size and (b), (d), (f) large packet size. (a) and (b) show the results for an irregular network, (c) and (d) for a hypercube network and (e) and (f) for a mesh.

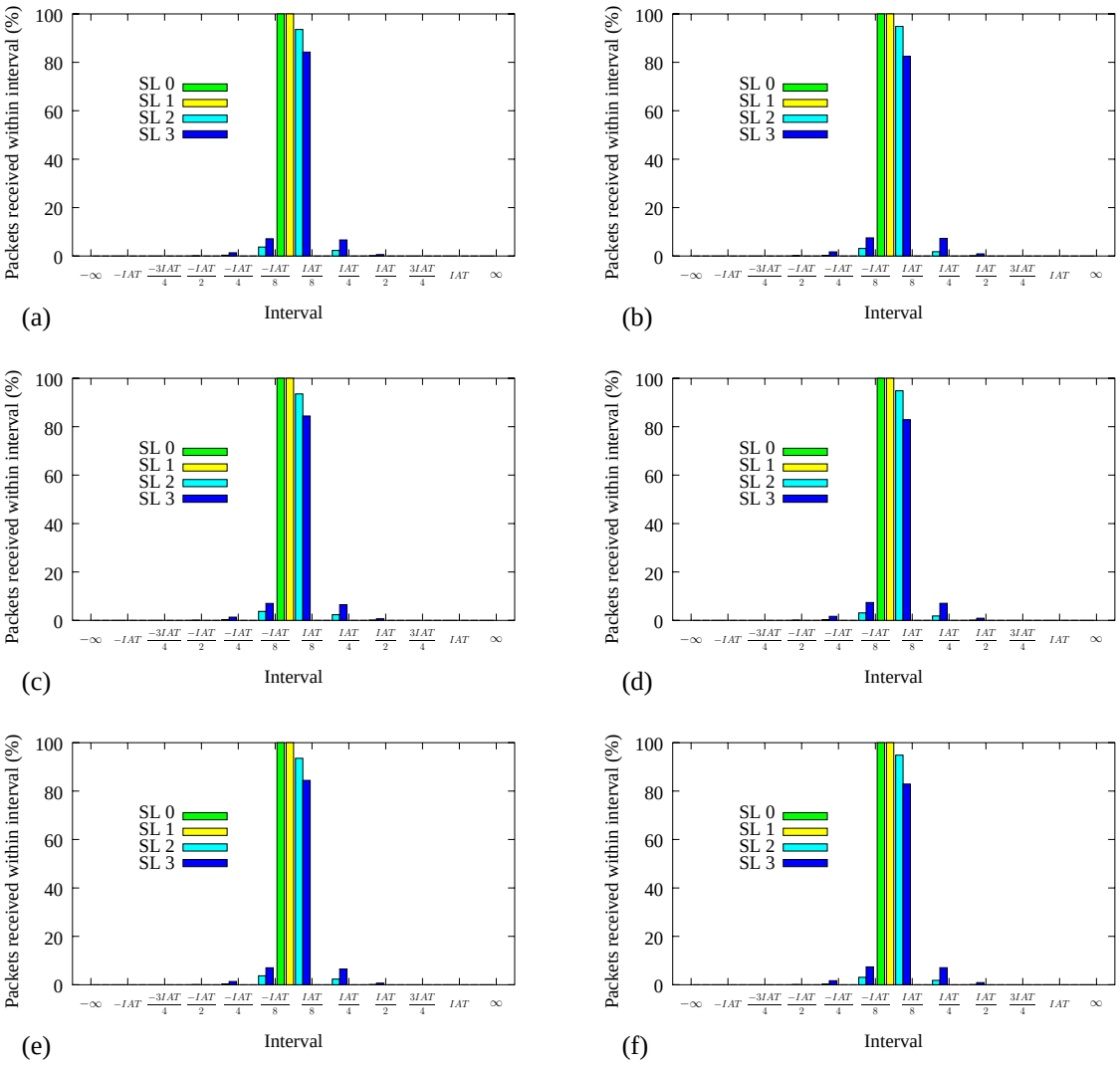

Figure 18: Average packet jitter for 4x (10 Gbps) IBA link rate, (a), (c) and (e) small packet size and (b), (d), (f) large packet size. (a) and (b) are for irregular network, (c) and (d) for hypercube and (e) and (f) for mesh.

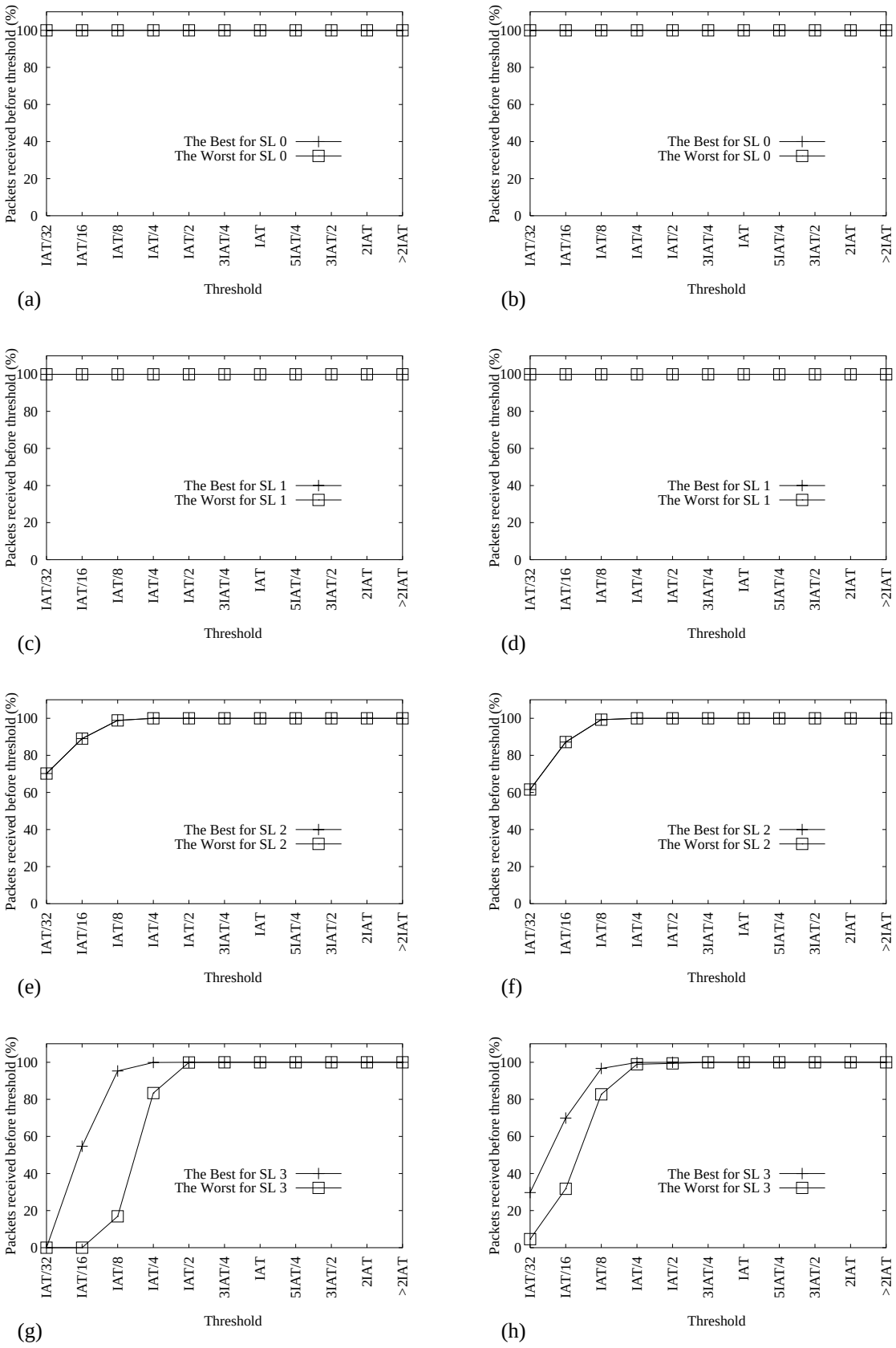

Figure 19: Behavior of the best and the worst connection for irregular network for 4x (10 Gbps) IBA link rate for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.

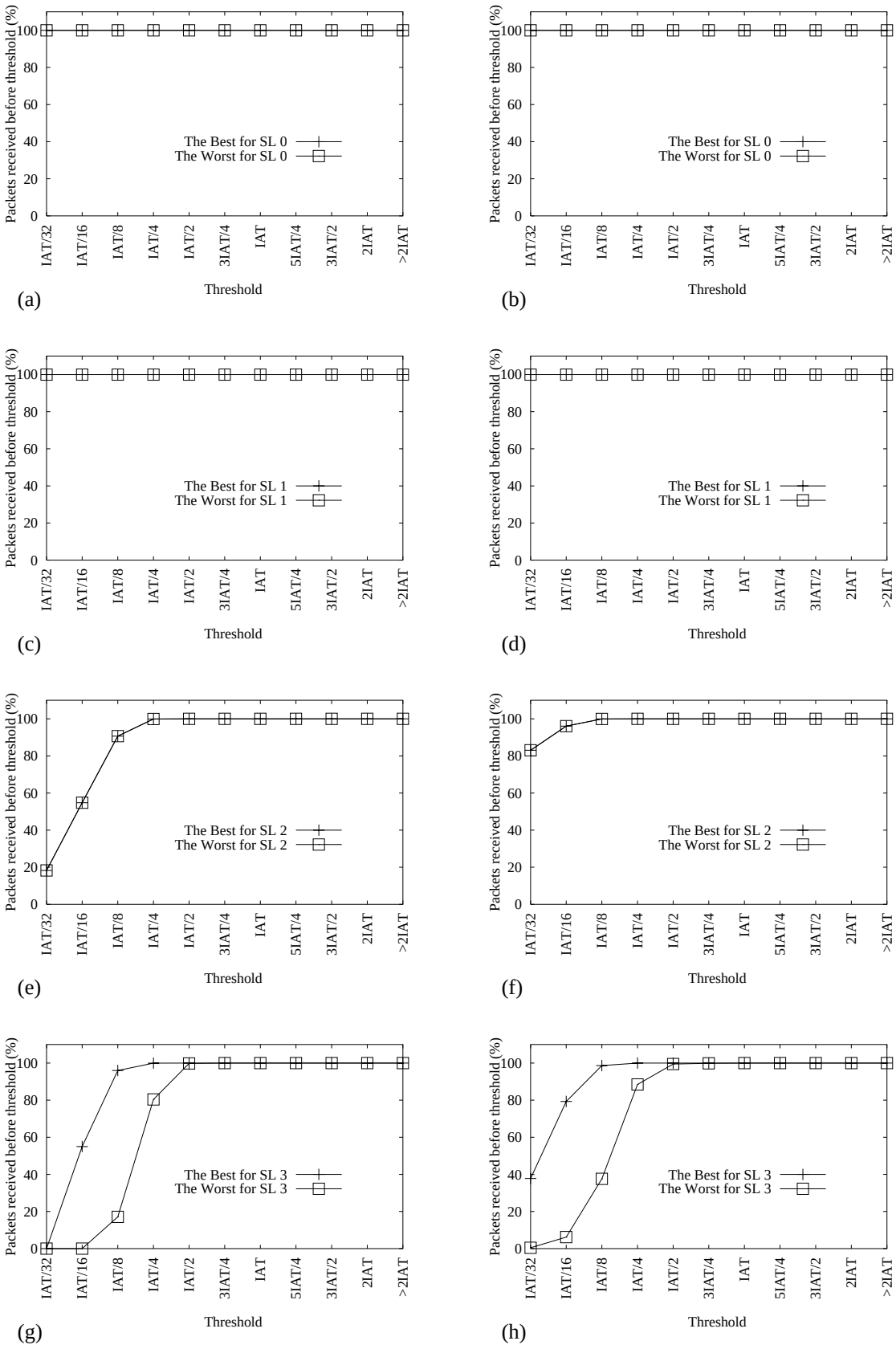

Figure 20: Behavior of the best and the worst connection for hypercube network for 4x (10 Gbps) IBA link rate for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.

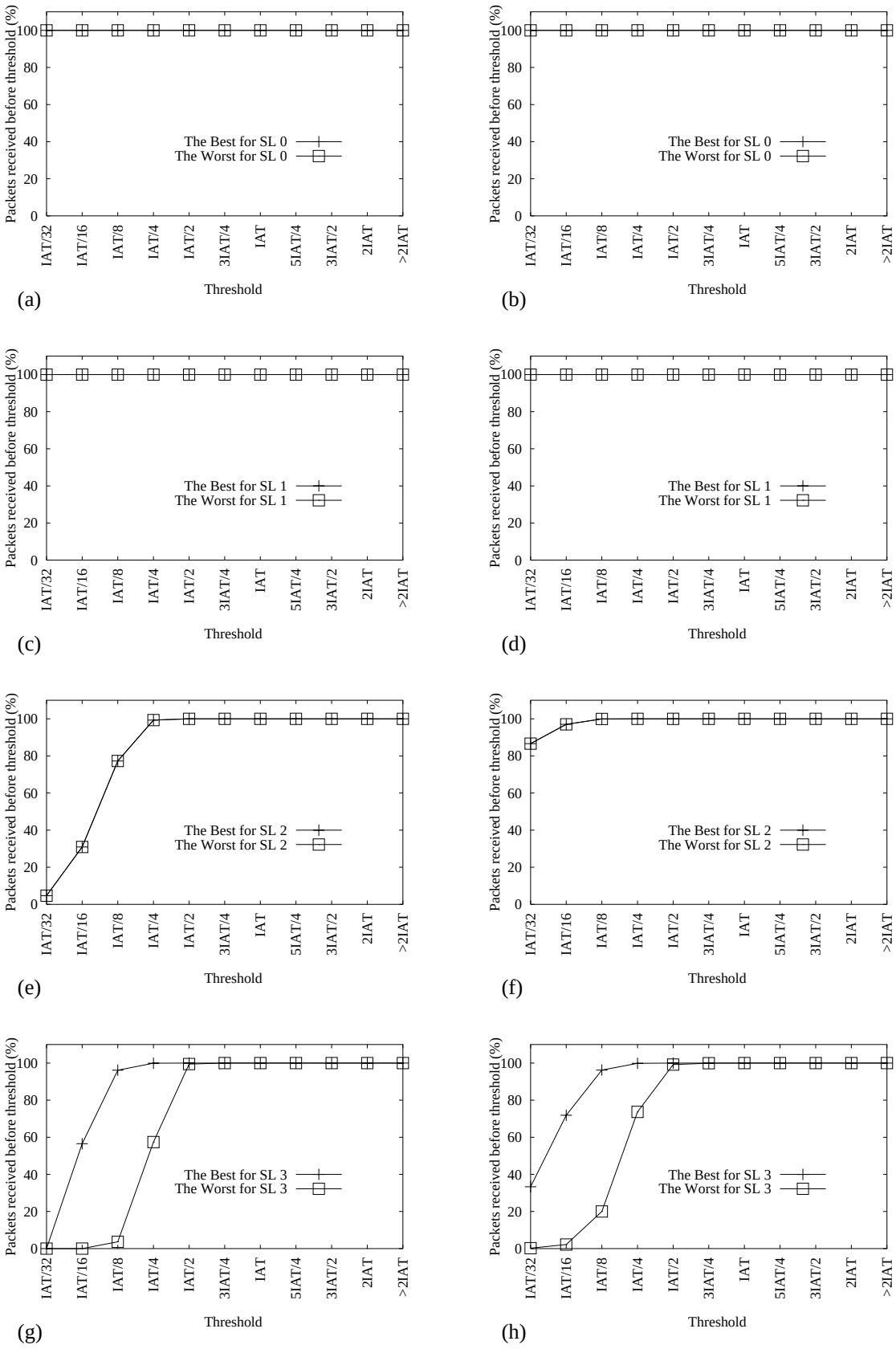

Figure 21: Behavior of the best and the worst connection for mesh network for 4x (10 Gbps) IBA link rate for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.

	Topology					
	Irregular		Hypercube		Mesh	
Tried	Packet 256	Packet 4096	Packet 256	Packet 4096	Packet 256	Packet 4096
5000	5000	5000	5000	5000	5000	5000
10000	10000	10000	10000	10000	10000	10000
15000	15000	15000	15000	15000	15000	15000
20000	20000	20000	20000	20000	20000	20000
25000	25000	25000	25000	25000	25000	25000
30000	30000	30000	30000	30000	30000	30000
31567	31664	32371	32476	35000	31347	31694

Table 4: Number of established connections for each network and packet size (in bytes) for different number of tried connections for 12x (30 Gbps) IBA link rate.

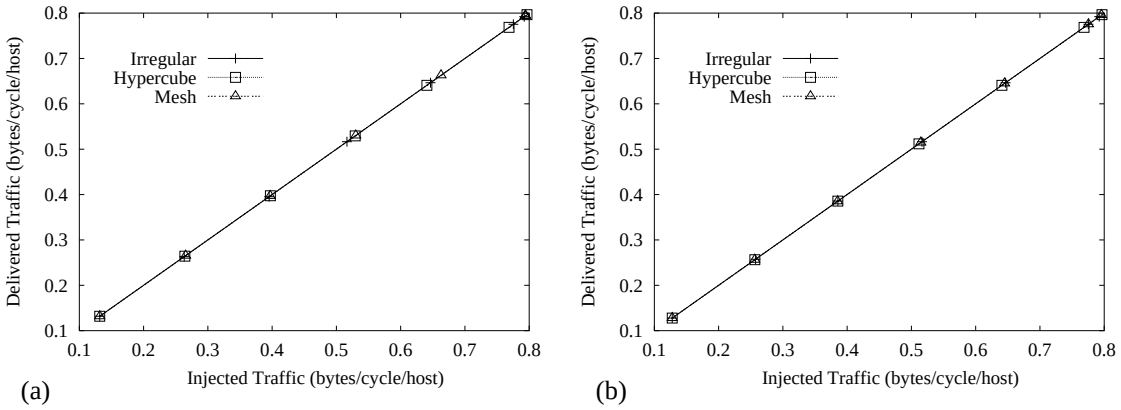

Figure 22: Delivered traffic for irregular and regular networks with (a) small packet size (256 bytes) and (b) large packet size (4096 bytes) for 12x (30 Gbps) IBA link rate

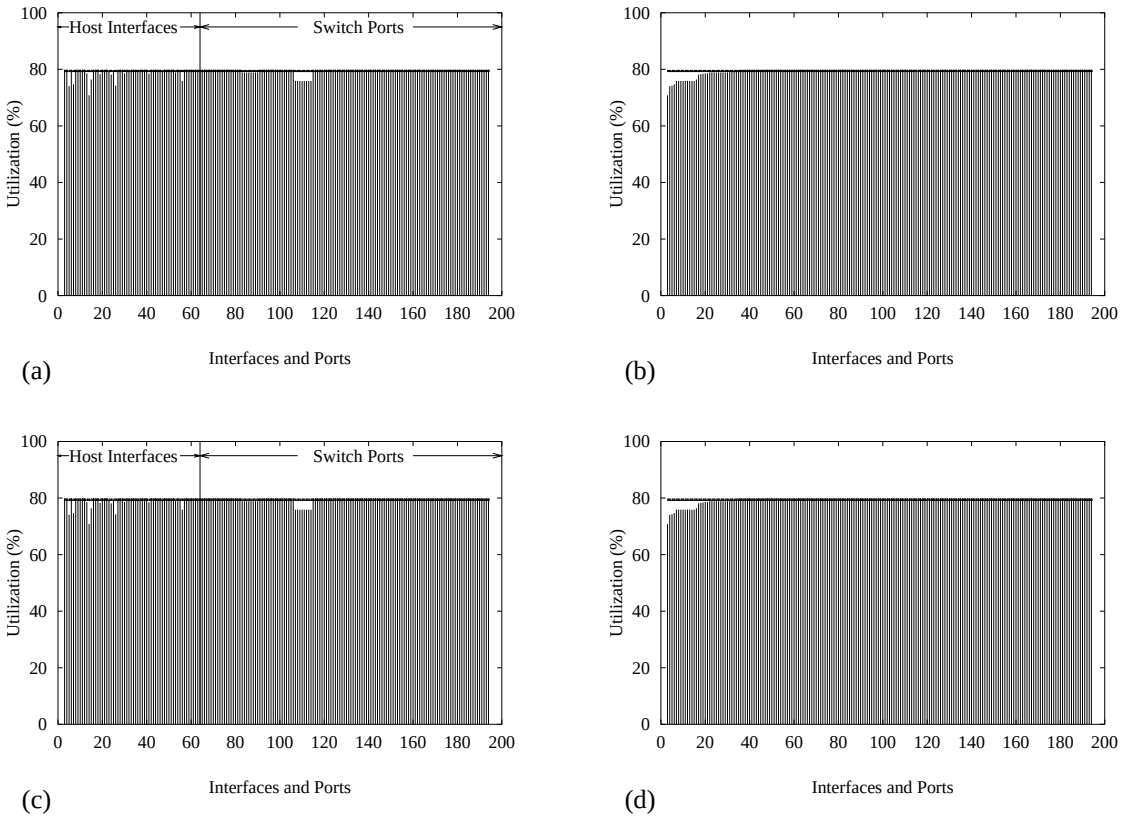

Figure 23: Utilization of switch ports and host interfaces for irregular networks with the maximum number of supported connections for 12x (30 Gbps) IBA link rate for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

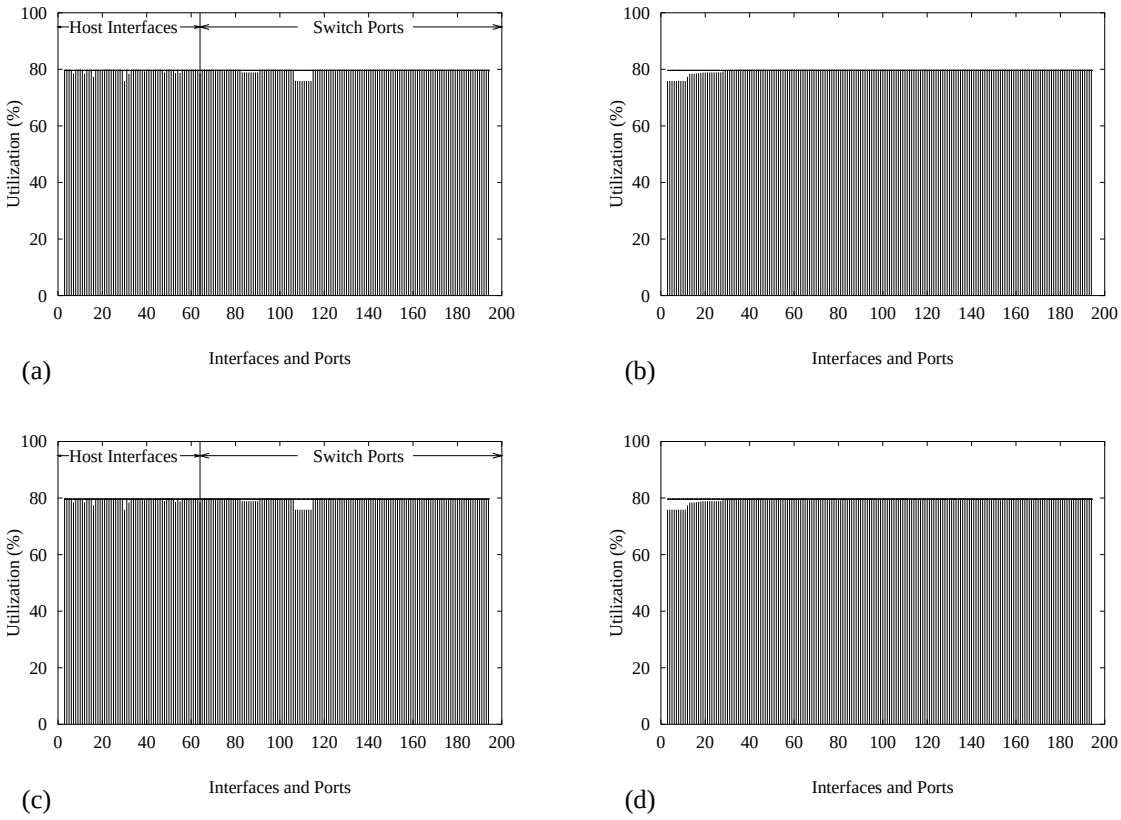

Figure 24: Utilization of switch ports and host interfaces for hypercube network with the maximum number of supported connections for 12x (30 Gbps) IBA link rate for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

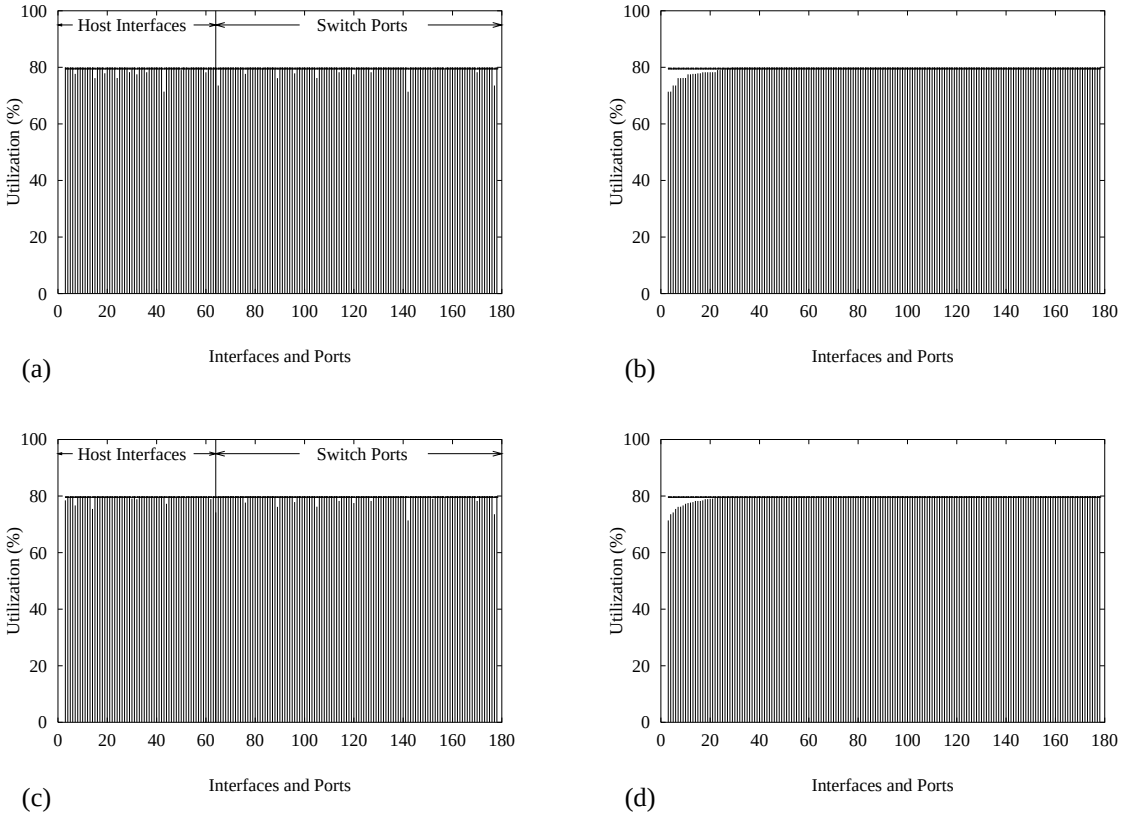

Figure 25: Utilization of switch ports and host interfaces for mesh network with the maximum number of supported connections for 12x (30 Gbps) IBA link rate for (a) and (b) small packet size and (c) and (d) large packet size. In (a) and (c), host interfaces are put before switch ports, and in increasing order of host/switch and interface/port identifier. In (b) and (d), utilizations are plot in increasing value order.

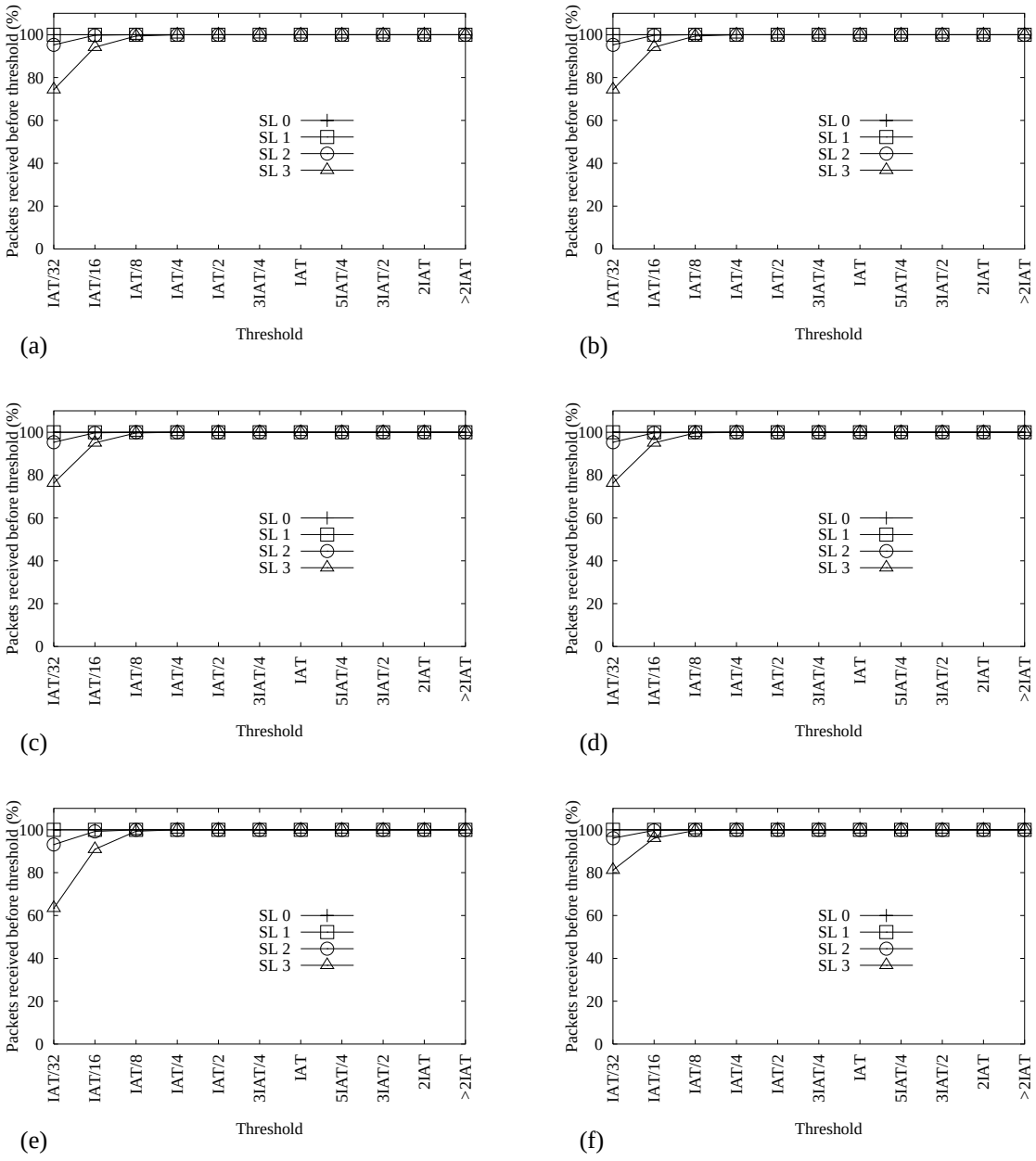

Figure 26: Distribution of packet delay for the maximum number of accepted connections for 12x (30 Gbps) IBA link rate, considering (a), (c) and (e) small packet size and (b), (d), (f) large packet size. (a) and (b) show the results for an irregular network, (c) and (d) for a hypercube network and (e) and (f) for a mesh.

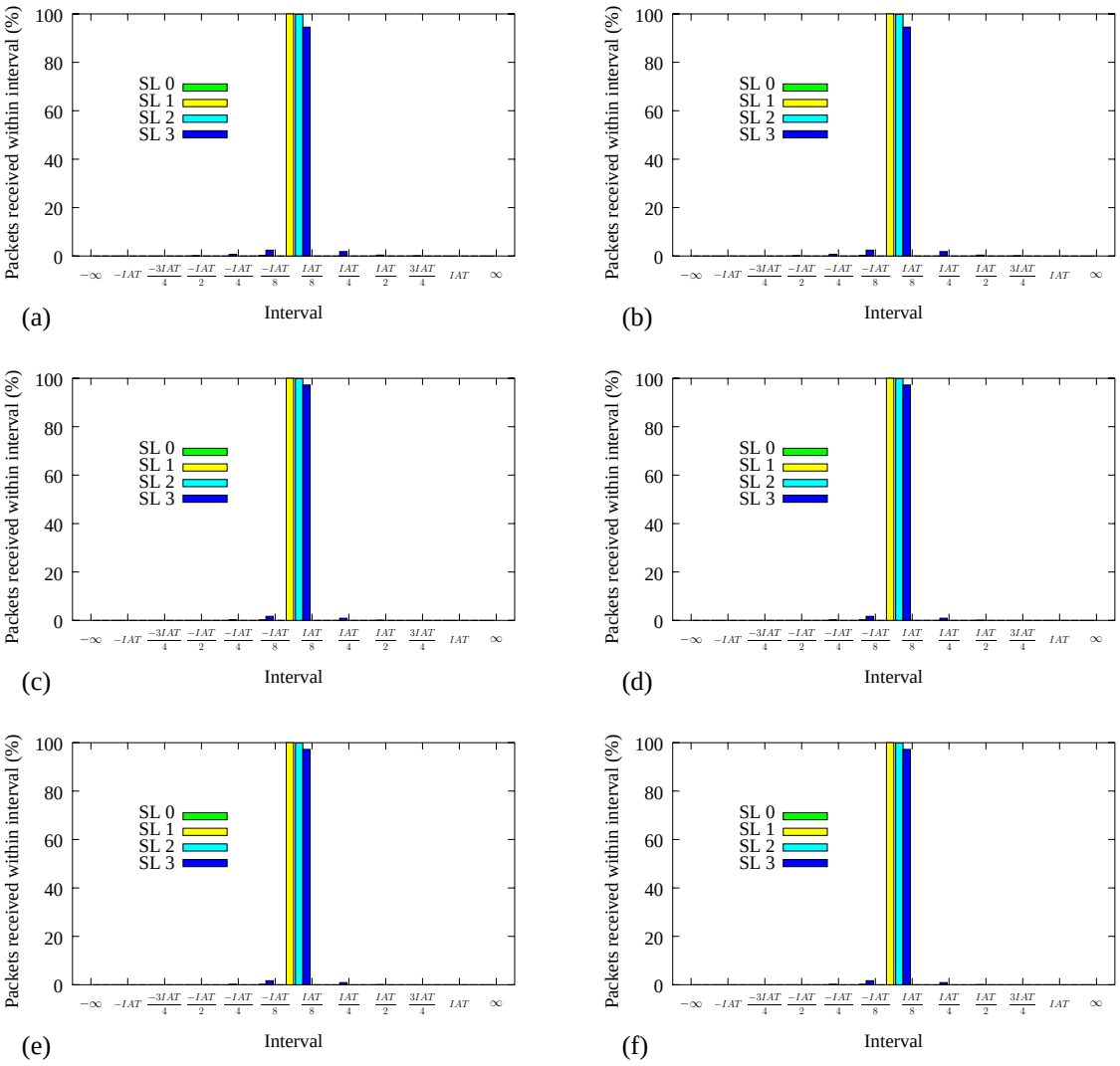

Figure 27: Average packet jitter for 12x (30 Gbps) IBA link rate, (a), (c) and (e) small packet size and (b), (d), (f) large packet size. (a) and (b) are for irregular network, (c) and (d) for hypercube and (e) and (f) for mesh.

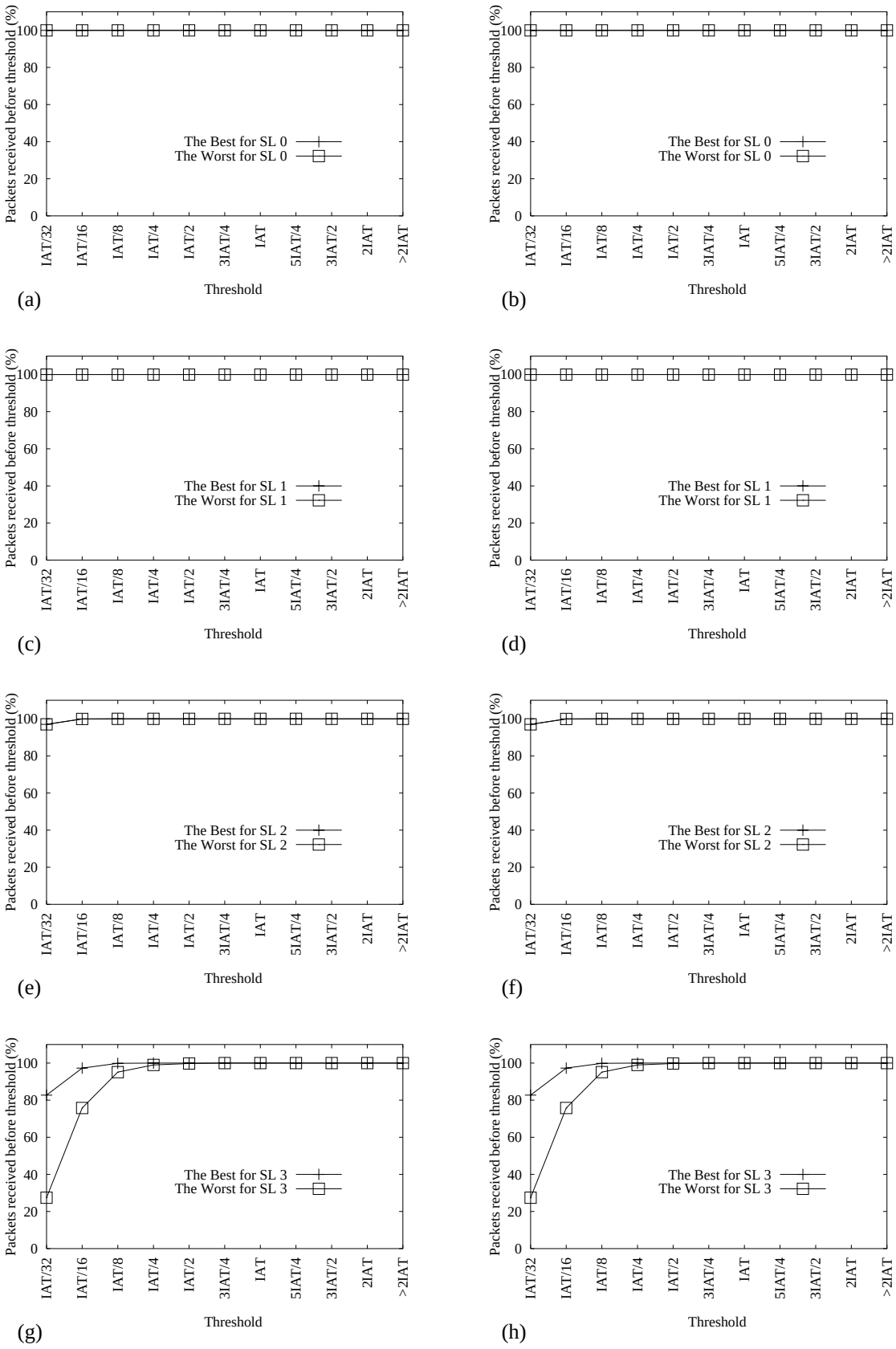

Figure 28: Behavior of the best and the worst connection for irregular network for 12x (30 Gbps) IBA link rate for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.

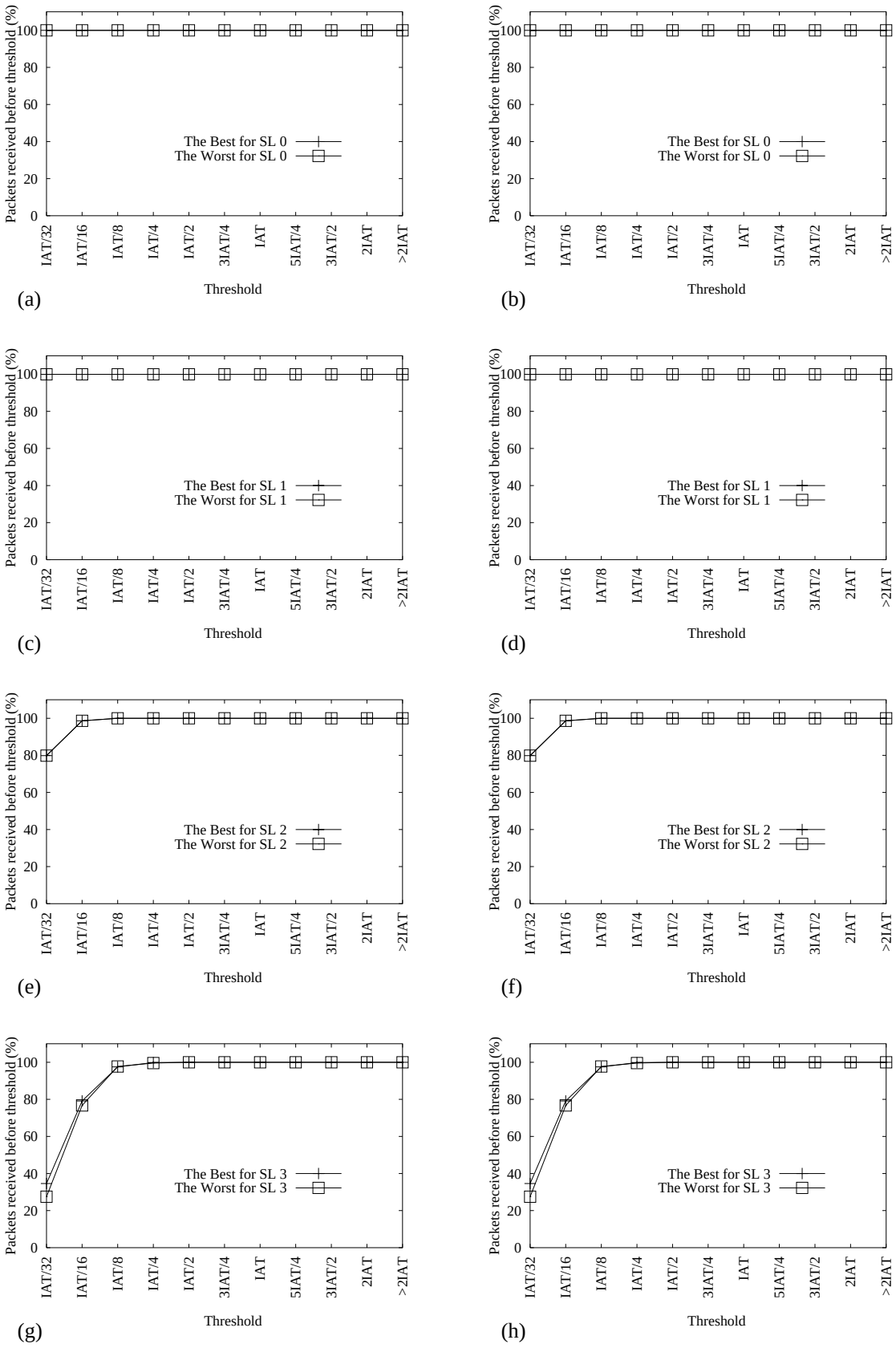

Figure 29: Behavior of the best and the worst connection for hypercube network for 12x (30 Gbps) IBA link rate for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.

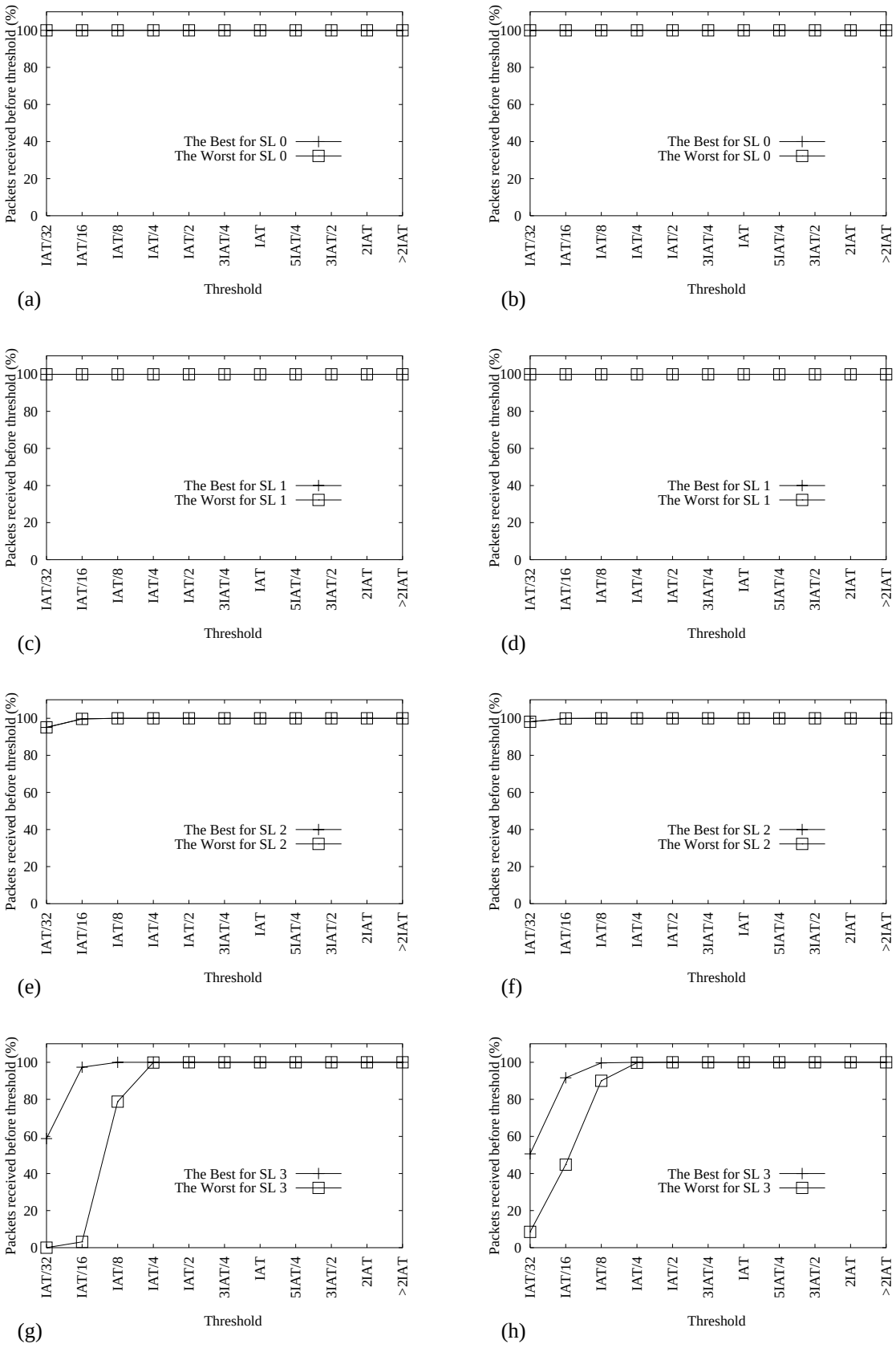

Figure 30: Behavior of the best and the worst connection for mesh network for 12x (30 Gbps) IBA link rate for (a), (c), (e) and (g) small packet size and (b), (d), (f) and (h) large packet size. (a) and (b) are for service level 0, (c) and (d) for service level 1, (e) and (f) for service level 2 and (g) and (h) for service level 3.