SPACEWIRE HOT MODULES

Session: SpaceWire Networks and Protocols

Long Paper

Asaf Baron, Isask'har Walter, Israel Cidon, Ran Ginosar, and Isaac Keslassy
VLSI Systems Research Center, Elec. Eng. Dept., Technion, Haifa 32000, Israel

Ofer Lapid
Israel Ministry of Defense

E-mail: {ab@tx, zigi@tx, cidon@ee, ran@ee, isaac@ee}.technion.ac.il, ofer.lapid@gmail.com

ABSTRACT

SpaceWire networks may include units, such as memory or processor units, which are bandwidth limited and in high demand by several other units. Such modules are termed Hot Modules. SpaceWire uses wormhole routing to deliver packets comprising multiple flits (N-chars). Wormhole routing helps minimize the number of buffers and the transmission latency. On the other hand, under high load, the network can become congested due to long packets that occupy resources at multiple network nodes, and block the paths of many other packets. This situation may be exacerbated at the presence of Hot Modules. In this paper we demonstrate that a single Hot Module can both dramatically reduce the network efficiency and cause an unfair allocation of system resources. First, the network efficiency is reduced because many packets can wait in different routers for the Hot Module to be available, while blocking other packets that are not destined to the Hot Module. Second, the allocation of system resources is unfair because in order to reach the Hot Module, packets from farther nodes need to win more arbitration cases than packets from nodes that are closer to the Hot Module. We explore several solutions for the Hot Module problem in SpaceWire networks. Possible solutions include provisions for priority-based routing and end-to-end access regulation mechanisms. We employ network simulations to investigate how the solutions address the network-efficiency and access-fairness problems.

1. INTRODUCTION

SpaceWire is a new set of specifications for onboard data-handling networks [1][2][3]. SpaceWire networks rely on wormhole routing [4] to transfer packets. In wormhole routing, each packet is divided into small fixed-size parts called N-chars (or flits), which are transmitted to the next hop without waiting for the entire packet to be received. This causes transmitted packets to be segmented and “spread” along the path between the source and destination in a pipeline fashion. Consequently, wormhole routing has relatively small buffer requirements, and incurs low latencies at light load. However, the main drawback of wormhole routing is its sensitivity to packet blocking, which can quickly fill up buffers along the entire path. In order to alleviate congestion effects, wormhole networks might employ virtual channels [5] to enable high utilization. Nonetheless, SpaceWire networks do not include virtual channels to allow packets to bypass blocked ones. Therefore, SpaceWire networks are especially vulnerable to high congestion.

The congestion problem in SpaceWire networks further intensifies at usage peaks, when the aggregated traffic demand exceeds the bandwidth of the destination module. Similarly, the destination module may also occasionally operate at a slower than average speed (e.g., a variable-speed encoder, decoder or storage device) and become congested, coincidently or not with an incidental usage peak. We term such a bandwidth-limited, high-demand unit a *hot module* [6]. In such situations, the hot module is unable to consume incoming packets fast enough. A possible example for such a module onboard a satellite is a central processing unit or the main mass-storage device.

Congested modules exist in systems with any communication infrastructure, but wormhole-based systems are much more sensitive to hot modules, as the entire network may be affected: The hop-by-hop backpressure, associated with wormhole routing, causes buffers at the router adjacent to the hot module to be filled up and become stalled, blocking new arrivals to this router. This creates a domino effect, by which the delivery of packets to ports of more distant routers is slowed down, forming a *saturation tree* [7] with the hot module as its root.

Figure 1 and Figure 2 illustrate such a hot-module saturation tree. First, Figure 1 shows how the elongated shape of a typical satellite makes the network topology to be typically linear or near-linear, when short cables are used to protect signal integrity. Such a topology is exemplified in Figure 2, which represents a possible network topology with all its modules (using the benchmark traffic introduced in [8]). In this network, the storage unit is a hot module. The links leading from the satellite sensors to the storage unit form a saturation tree, shown in bold. (Note that a second saturation tree exists on the links from the storage unit to the downlinks.) The delivery of packets along the links in this saturation tree will be especially slow. Moreover, note that the domino effect stretches beyond the traffic on the saturation tree, as packets that are destined to other modules find no free buffers at certain routers on their route. For instance, packets sent from unit 29 to the Data Handling System (DHS) might get blocked in the network, blocking in turn packets sent from unit 25 to unit 23, which did not have any common link with the saturation tree. Thus, the network suffers from increased delays in packet delivery, as well as from unfair network utilization. This double threat is particularly troublesome in wormhole-based architectures due to packet "stretching" across multiple hops causing the hot-module effects to extend network-wide instantly. This phenomenon is *independent of links and router bandwidth*. In fact, such a network freeze may build up even in a system with infinite capacity links because of a single heavily-loaded module causing full buffers. Consequently, even largely over-provisioned networks suffer from poor performance if potential hot modules are left unhandled.

RELATED WORK

The negative effects of hot modules were partially explored in off-chip interconnection networks (e.g. [7]-[14]). In that literature there is no clear distinction between the issue of hot-module and the congestion of a network port. Typically, suggested solutions attempt to prevent regular traffic from being affected by the traffic of a saturation tree, either by not allowing one to form or by allocating hot traffic exclusive network resources. Unfortunately, such solutions do not bring a fair allocation of the hot resource. For example, some works modify the network routers in order to throttle packet injection at high loads (e.g., [10], [11]), discard packets (e.g., [12]), deflect packets away from loaded locations (e.g. [13], [14]), use separate buffers for traffic destined to a hot module (e.g. [9]), or simply use a large number of virtual channels. However, such solutions complicate the design of the network and may significantly increase its hardware cost. Note that most of the previous techniques can only slightly postpone the hot-module effect as they increase the number of buffers used by non-hot-module traffic (by adding buffers or routing non-hot-module traffic away), and in some cases they can even exacerbate the hot-module effect by throttling non-hot-module packets.

In [6], similar problems were addressed in an on-chip environment. In this paper, we partly use the solution proposed in [6]. However, since N-Char interleaving is not allowed in SpaceWire networks, we also show how the solution needs to be further improved.

Finally, other solutions to network congestion problems were proposed in SpaceWire networks as well [15][16], but none of these specifically address the hot-module problem.

RESULTS

In this paper, we examine two solutions to alleviate the hot-module effects in SpaceWire networks. First, we explore a simple mechanism that supports Packet-Level Priority (PLP) [8]. In this scheme, when competing for network resources at the routers, packets traveling to the Hot Module receive a low priority, while other packets enjoy a higher level of priority. We show how this PLP-based mechanism manages to isolate the hot-module traffic from non-hot-module traffic, and therefore to mitigate the detrimental effects of hot modules on non-hot-module traffic.

In addition, we discuss a one-to-many end-to-end credit-based access regulation mechanism [6]. An allocation controller is introduced to arbitrate between the requests sent by the traffic sources. The controller allows the system designer to regulate hot-module access according to the quality of service requirements of the specific system application. The allocation algorithm employed by the controller is system-specific, since the hot module is independent of the network. Requests and grants are transmitted as small high-priority *credit* packets. In this paper, we explain how the credit mechanism prevents the accumulation of packets destined to a hot module within the network buffers. Consequently, we show how non-hot-module traffic remains unaffected even as the hot-module load increases significantly. In addition, we demonstrate how this credit mechanism alleviates the unfairness among flows resulting from their distance to the hot module. Finally, we show how combining the two solutions based on PLP and credits yields the best results among the proposed solutions.

The rest of this paper is organized as follows: In Section 2, the effects of hot modules on SpaceWire networks are discussed. Section 3 introduces two mechanisms to mitigate these effects, and Section 4 presents simulation results for the suggested techniques. Finally, Section 5 concludes this paper.

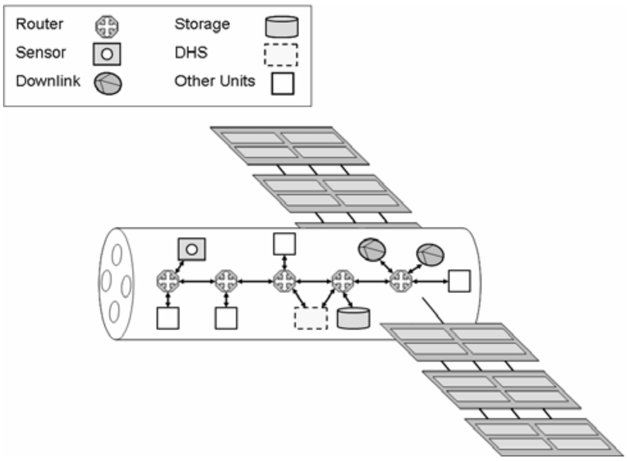

Figure 1: Elongated Shape of a Typical Satellite

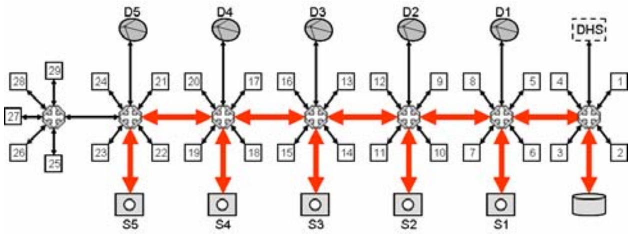

Figure 2: SpaceWire Network Topology

2. HOT-MODULE EFFECTS

The network buffer congestion due to hot modules has several negative effects on system performance. First, the hot-module *access latency* is increased, as packets destined to it contend for the limited hot-module bandwidth. As a result, when the load increases, the saturation tree becomes more and more congested, and packets traveling on the links of the saturation tree experience higher delays.

In addition, significant fairness problems arise, because different source modules are typically located at different distances from the hot module (as illustrated in Figure 2). More precisely, modules close to the hot module enjoy a much larger share of the hot-module bandwidth than distant ones. This is caused by the fact that routers typically employ a locally fair, round-robin arbitration between packets (or N-Chars) of similar priority waiting at different input ports and contending for the same output port. Therefore, when its inputs are saturated, each router that is part of the hot-module saturation tree equally divides the bandwidth available at its upstream port among its input ports. Consequently, hot-module throughput at a source can be modeled as dropping exponentially as a function of the number of hops between the source and the hot module. Further, the presence of other hot modules can also cause significant differences in access latency as a function of the number of hops —

in fact, packets sent by more distant sources are more likely to be blocked by traffic destined to another hot module. As a consequence, units that are located relatively far from the hot module experience extremely long access times when the hot-module load increases.

Furthermore, performance degradation due to hot-module load is not restricted to the hot-module traffic itself. In typical networks, hot-module and non-hot-module traffic compete for the same network buffer space and router ports. Therefore, hot modules that slowly service incoming data hinder the delivery of non-hot-module packets, as blocked hot-module packets wait inside the network occupying expensive buffers. As a result, packets destined to lightly loaded modules are also being stalled by the network, suffering delays and fairness problems similar to hot-module packets.

Note that the above discussion applies to any network in the presence of congested end-points. However, left unhandled, hot-module effects in the SpaceWire wormhole network are more severe than in a store-and-forward network, as packets are blocked across multiple routers and buffering space is limited. Further, hot-module effects are especially severe in SpaceWire networks, because N-Char interleaving is not allowed, and therefore it is harder to bypass blocked packets.

Also, it is important to note that these delay and fairness effects are symptoms of hot modules and not of an inadequately provisioned network. In fact, a wormhole network would suffer from the presence of a hot module even with links and routers of infinite capacity.

3. REDUCING HOT-MODULE EFFECTS

In order to reduce the dramatic effects that hot modules have on SpaceWire networks, we examine two possible solutions: a simple, inexpensive packet priority scheme and a many-to-one end-to-end credit-based arbitration mechanism. We first evaluate each of the suggested mechanisms separately, and will later consider using them jointly.

PACKET-LEVEL PRIORITY (PLP)

In this scheme, packets from different flows are assigned different packet-level priorities [8]. In particular, long packets destined at the hot-module have lower priority than short packets that are destined to the Hot Module and packets destined at other modules. Packet priorities are considered during output-port arbitration in network routers; if packets with different priorities contend for a single router output port, the packet with the highest priority wins the arbitration.

CREDIT-BASED ACCESS REGULATION PROTOCOL

The second scheme implemented in this paper is a credit-based access regulation mechanism [6]: each source can inject a packet into the network only after being granted permission by a *hot-module allocation controller*. The idea underlying this scheme is that packets that cannot be serviced by the hot module are not injected into the network and therefore do not wait inside the network. Consequently, a saturation tree cannot form and traffic not destined to the hot module remains unaffected during congested periods.

Two types of control messages regulate the access to a hot-module: a *credit request* packet, by which a source asks for permission to transmit a packet to the hot module; and *a credit reply* packet, which is sent by the hot-module allocation controller to grant permission. Due to their significance and short length, request and reply messages are given a high priority level. All other traffic in the network has normal priority.

The hot-module allocation controller can implement any arbitration scheme to select the next source module to be served, according to the designer's will. In this work, we assume simple round-robin service.

Finally, we distinguish between two possible implementations of the credit approval mechanism. First, in the original mechanism described in [6], new credit replies are transmitted just before the end of the transmission of packets currently received by the hot module. However, since credit replies may need to wait a long time for the transmission of the current packets to be over, such an implementation might incur higher delays. It makes sense in [6] thanks to flit interleaving, which is unavailable in SpaceWire networks. Therefore, unless mentioned, we use a new credit approval mechanism, in which the credit approval is sent by the hot module as soon as the first N-Char of the packet currently received at the hot module has arrived. In the next section, we further discuss the effects of these two alternative implementations.

4. SIMULATION RESULTS

In this section, the performance of the two schemes is examined by means of simulation. We present four sets of results: (1) a standard SpaceWire network; (2) a SpaceWire network that supports PLP; (3) a SpaceWire network with a credit-based access regulation protocol; and (4) a SpaceWire network with both PLP and a credit-based access regulation protocol.

The term “end-to-end latency” in this paper refers to the time elapsed since the packet is created at the source until its last flit enters the destination. Therefore, the measured latency accounts for source queuing, network blocking, multiplexing, link bandwidth limitations, and overhead of the access regulation protocol. The results are generated using the OPNET based simulator [17], modeling a SpaceWire network at the N-Char level. The model includes all network-layer components, including wormhole flow control, routing, finite router buffers and link capacities.

SIMULATION SETTINGS

The topology of the simulated network can be seen in Figure 2. We use the benchmark from [8], but instead of using only one sensor that sends Payload (PLD) Data in a rate of 300 Mbit/sec, we use 5 sensors that send 60 Mbit/sec of PLD Data each. All the PLD traffic created by the sensors is destined at the storage in packets of 60,000 bytes each. Likewise, the storage also creates PLD traffic at a rate of 300 Mbit/sec that is destined to the downlink units also in packets of 60,000 bytes. The rest of the traffic is composed of telemetry data that is sent to the DHS by all other units and telecommand data that is sent from the DHS to all other units.

RESULTS

Figure 3 (respectively Figure 4) illustrates the average end-to-end delay of payload packets destined to the storage hot module (respectively to the DHS module). Both figures display four different plots, each based on a different scheduling methodology. Further, each hot-module plot displays five different curves and each non-hot-module plot displays seven curves, corresponding to the average ETE (end-to-end) delay in milliseconds over all sources located a given number of hops away from the destination.

Each plot shows the average ETE packet delay as a function of the hot-module link utilization. In order to illustrate the influence of the hot-module link utilization on the ETE delay, on each plot, the traffic demand is kept identical, while the hot-module link capacity is gradually reduced so as to increase the hot-module link utilization (the ratio of the fixed demand by the variable capacity), and accentuate the corresponding hot-module effects. The x-axis of each plot indicates the hot-module link utilization, comprised between 0 and 1.

Figure 3(a) illustrates the average ETE delay of packets from the sensors to the storage hot module, when the network strictly follows the standard SpaceWire specifications. It clearly shows how the average ETE delays are large and increase as the hot-module link utilization increases. It also demonstrates the hot-module unfairness effect, since the average ETE delays increase for packets with more hops on the way. Similarly, Figure 4(a) illustrates the average ETE delay of traffic that is destined to the DHS when the network follows the standard SpaceWire specifications. Again, the average ETE delays highly depend on the hop location, causing extreme unfairness between flows.

Figure 3(b) shows that using PLP does not improve the ETE delay of long packets destined to the hot module, because they all have low priority. Further, it also does not significantly mitigate the unfairness problem. On the contrary, Figure 4(b) shows that using PLP does help in reducing the ETE delay of traffic that is *not* destined to the hot module, because the priority of this traffic is higher. However, the improvement is not dramatic, because the network is still congested with the long packets that are destined to the hot module

Figure 3(c) illustrates the use of the credit-based mechanism. It demonstrates how this mechanism manages to significantly reduce delay unfairness, because the hot module fairly allocates credits among all sources. In addition, Figure 4(c) shows how the credit-based scheme reduces the average delay of packets destined to the DHS unit. This is because the network is less congested by hot-module traffic, and therefore non-hot-module traffic is less affected by the hot-module congested flows. Obviously, since the credit-based mechanism only concerns hot-module traffic, it does not significantly reduce unfairness among flows destined to the DHS unit. Also, note that as the hot-module link utilization goes to 1, non-hot-module traffic is not directly affected, and therefore its average ETE delay will stay bounded, and oscillate around some value depending on the random traffic — hence the simulation artifact in the figure.

Finally, Figure 3(d) and Figure 4(d) show the results when jointly using both mechanisms. These figures illustrate how the combination of both mechanisms significantly mitigates the hot-module effects, by isolating hot-module and non-hot-module traffic, reducing hot-module traffic delay, and reducing the unfairness in the average ETE delay of hot-module traffic.

Figure 5 illustrates how the scheduling scheme strongly affects the hot-module unfairness effect. It plots the average ETE delay in milliseconds as a function of the source location, with a hot-module link utilization of 97%, using three different scheduling schemes. In the first scheme, a standard SpaceWire network is used, as in Figure 3(a) and Figure 4(a). It can clearly be seen that the delay increases with the number of hops that separate the source from the destination, for hot-module traffic as well as for non-hot-module traffic.

The two other schemes are both credit-based schemes. The first scheme uses the original implementation that posts a credit upon the arrival of the last N-Char, and the second one uses the new implementation that posts a credit upon the arrival of the first N-Char. As seen in Figure 5(a), the delay unfairness among hot-module flows nearly disappears, especially with the new implementation, which additionally reduces the average ETE delay by issuing credits upon the arrival of the first N-Char instead of the last one. Further, since credit-based schemes reduce the amount of hot-module traffic in the network, non-hot-module traffic finds less congestion and therefore encounters lower delays, as shown in Figure 5(b). In particular, in the original implementation of the credit-based scheme, credits take more time to arrive than in the new implementations, and therefore non-hot-module traffic finds less hot-module traffic congestion, thus leading to even lower delays.

5. CONCLUSION

In this paper, we showed how SpaceWire networks can contain hot modules that might cause congestion throughout the network and engender high delays and delay unfairness among traffic sources. In order to solve these problems, we have analyzed two mechanisms: first, a packet-level priority (PLP) scheme; and second, a credit-based end-to-end arbitration scheme. We showed how a combination of these schemes can efficiently mitigate the effects of hot modules and protect non-hot-module traffic.

Many of the problems mentioned in this paper can hardly be analyzed, or even detected, without simulation and benchmarking tools. For instance, we showed how non-hot-module traffic can be affected, even when it does not share any link with the saturation tree. We believe that such examples, sometimes quite counter-intuitive, further support the use of benchmarks and simulations, as argued in our companion paper [8].

REFERENCES

[1] Internet reference: spacewire.esa.int

[2] S.M. Parkes, “SpaceWire: Links, Nodes, Routers and Networks”, European Cooperation for Space Standardization, standard number ECSS-E50-12A, January 2003.

[3] S.M. Parkes and C. McClements, “SpaceWire Networks”, DASIA 2002.

[4] W.J. Dally and C. Seitz, “The Torus Routing Chip”, Distributed Computing, 1(3), 1986.

[5] W. Dally, “Virtual Channels Flow Control”, Int. Symp. Computer Architecture (ISCA), 1990.

[6] I. Walter, I. Cidon, R. Ginosar, and A. Kolodny, “Access Regulation to Hot-Modules in Wormhole NoCs”, First IEEE/ACM Int. Symp. on Networks-on-Chip (NOCS07), 2007.

[7] G. F. Pfister and V. A. Norton, “Hot Spot Contention and Combining in Multistage Interconnection Networks”, IEEE Trans. Comp., C-34(10), 1985.

[8] A. Baron *et al.*, “Benchmarking SpaceWire Networks”, Int. SpaceWire Conf., 2007.

[9] J. Duato *et al.*, “A New Scalable and Cost-Effective Congestion Management Strategy for Lossless Multistage Interconnection Networks”, High-Performance Computer Architecture (HPCA), 2005.

[10] E. Baydal, P. Lopez, and J. Duato, “A Congestion Control Mechanism for Wormhole Networks”, Ninth Euromicro Workshop on Parallel and Distributed (PDP '01), 2001.

[11] A. Smai and L. Thorelli, “Global Reactive Congestion Control in Multicomputer Networks”, 5th Int. Conf. on High Performance Computing, 1998.

[12] W. S. Ho and D. L. Eager, “A Novel Strategy for Controlling Hot-spot Congestion”, lnt. Conf. Parallel Processing, 1989.

[13] T. Lang and L. Kurisaki, “Nonuniform Traffic Spots (NUTS) in Multistage Interconnection Networks”, J. Parallel and Distributed Computing, 1990.

[14] P. Gawghan and S. Yalamanchi, “Adaptive Routing Protocols for Hypercube Interconnection Networks”, IEEE Trans. Comp., 1993.

[15] S Mills, S Parkes, “TCP/IP Over SpaceWire”, DASIA 2003.

[16] S. Parkes et al., “CCSDS Time-Critical Onboard Networking Service”, SpaceOps 2006.

[17] OPNET Modeler, www.opnet.com

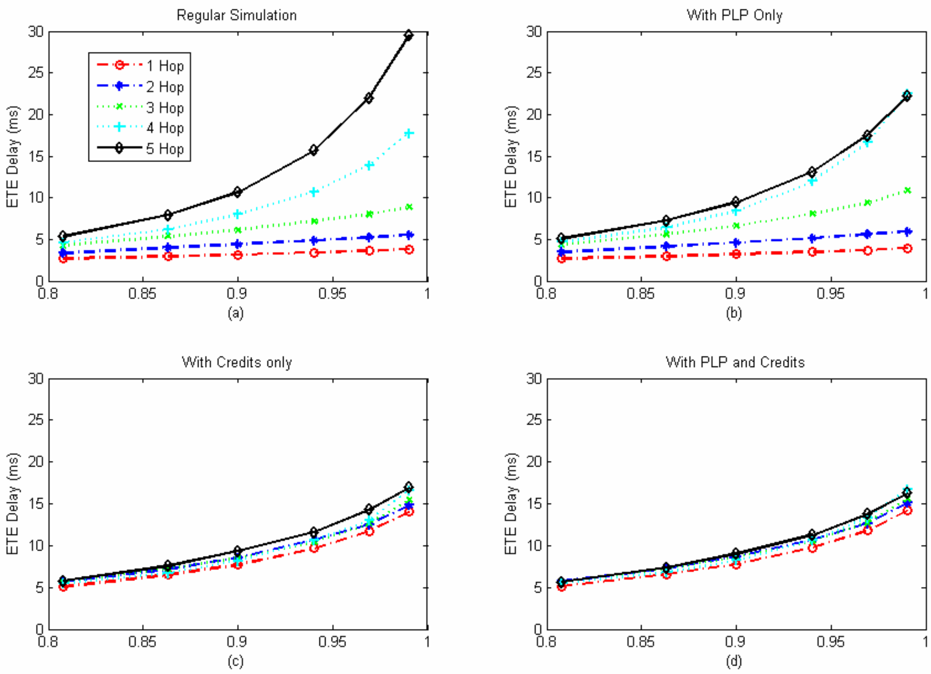

Figure 3: Average End-to-End Delay to the Storage Hot Module

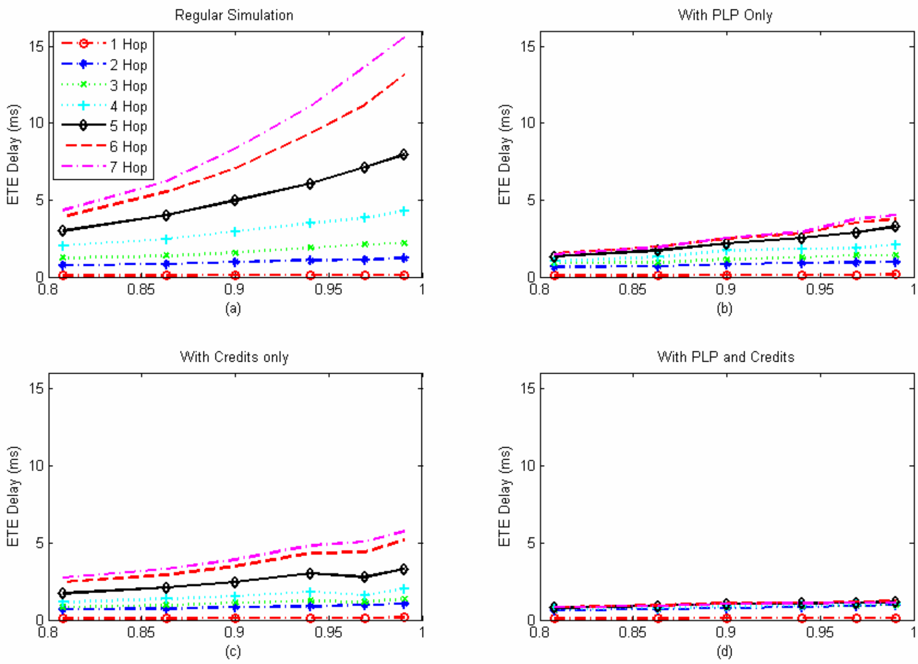

Figure 4: Average End-to-End Delay to the DHS Unit

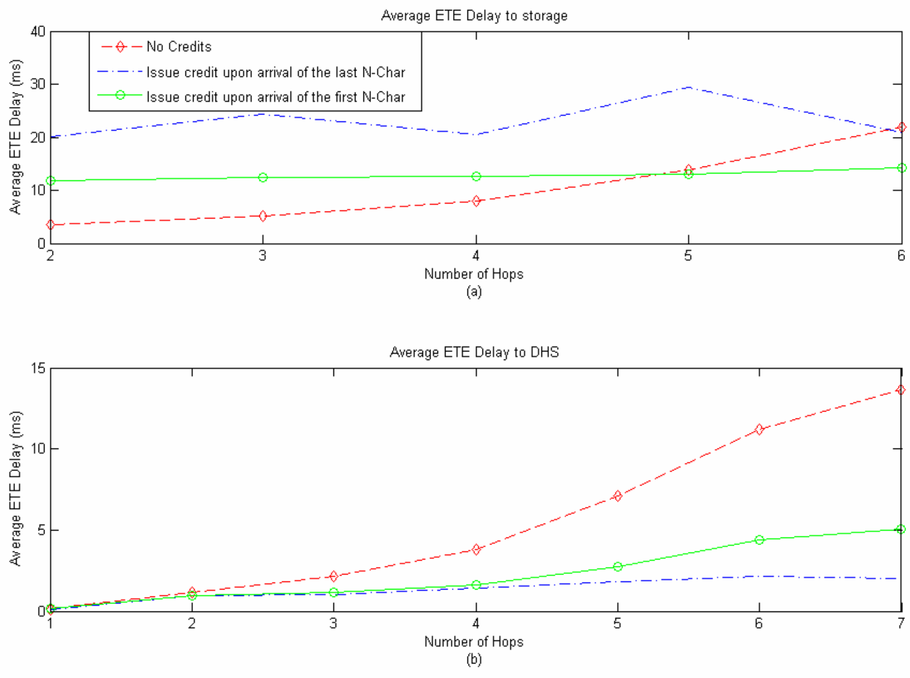

Figure 5: Delay Unfairness