32nd IEEE Conference on Local Computer Networks

An Initial Performance Evaluation of Rapid PHY Selection (RPS) for Energy Efficient Ethernet

Francisco Blanquicet and Ken Christensen
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
{fblanqui, christen}@cse.usf.edu

***Abstract*— The IEEE 802.3 Energy Efficient Ethernet (EEE) study group is considering Rapid PHY Selection (RPS) as a mechanism to quickly switch the data rate of an Ethernet link to match link data rate with link utilization. When switching the data rate, RPS causes a momentary disruption of the link. This disruption may cause packet loss due to buffer overflow in upstream switches. We emulate RPS using PAUSE flow control and experimentally study the possible effects of RPS on TCP and UDP file transfer. We show that RPS has little or no perceivable effect on performance, but has some subtle effects on TCP throughput if PAUSE flow control is enabled in the file server.**

I. INTRODUCTION

Reducing the energy use of IT equipment is becoming increasingly important for economic and environmental reasons. Data centers are constrained by power demands of servers and networking equipment. Beyond data centers are millions of Ethernet links from desktop PC to LAN switch. Many of these links operate at a very low utilization level most of the time. These links are rapidly migrating to a 1 Gb/s default link data rate (and very likely to 10 Gb/s within the next 10 years). Measurements have shown that a 1 Gb/s link consumes approximately 2 to 4 W more than a 100 Mb/s link [3]. If lightly utilized 1 Gb/s links could operate at a lower data rate (to match link utilization), it is estimated that a savings of about $480 million per year in the US would be achieved [6].

The Ethernet community formed the IEEE 802.3 Energy Efficient Ethernet (EEE) study group in November 2006 [5]. The major focus of the EEE study group is to investigate methods of matching Ethernet link data rate to link utilization. The idea of matching link data rate to utilization was first explored as Adaptive Link Rate (ALR) in [3]. To achieve this, a fast mechanism for switching link data rate is needed. A new mechanism called Rapid PHY Selection (RPS) is being considered as a means of quickly changing link data rate. RPS could be based on a MAC frame handshake to initiate a link date rate switch. Following the handshake, the link would resynchronize at the new data rate (e.g., from 1 Gb/s to 100 Mb/s). The current direction in the IEEE 802.3 EEE study group is that RPS would be allowed for all PHY types. In addition to a mechanism to switch the link data rate, a control policy is needed to determine when to switch the data rate. Control policies have been previously investigated in [4]. Figure 1 shows an RPS handshake and link resynchronization.

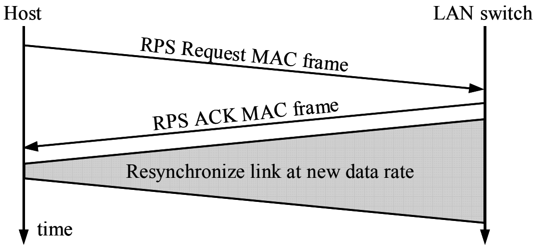

Figure 1. RPS handshake and link resynchronization (RPS switch)

During the resynchronization period the link is disrupted, or blocked, and no packets can be transmitted on it. Thus, packets arriving into a LAN switch and destined for a blocked link could potentially fill up and overflow a switch buffer causing packet loss to occur. The effects of RPS-induced packet loss on higher layer protocols and applications are unknown.

II. ANALYSIS OF RPS-INDUCED PACKET LOSS

A worst case loss situation will occur if a burst of packets is arriving into a switch buffer at the same time when an RPS switch occurs on the downstream (destination) link. In this case, the switch will be unable to deliver the arriving packets to the downstream link during the period of link disruption caused by the RPS switch. If packets arrive at a full data rate, in 1 millisecond at 1 Gb/s, 122 KB of packet data will arrive. At 10 Gb/s, 1.2 MB of packet data will arrive.

The amount of packet loss, L, in bits can be calculated for a given burst rate, R, burst size, B, switch buffer size, S, and RPS switching time, T, as,

$$L = R \cdot \min\left(\frac{B}{R}, T\right) - S \qquad (1)$$

for $R \cdot \min(B/R, T) < S$. If $R \cdot \min(B/R, T) > S$ then no packet loss will occur. Equation (1) assumes packets as a fluid flow, so to determine the number of packets lost we would take the ceiling of L divided by the mean packet length. Using (1) we can analyze loss as a function of switch buffer size or any other variable parameter. Figure 2 shows a plot of percentage of a burst lost as a function of buffer size for $T = 10$ ms, $B = 10$ MB, and $R = 2$, 4, and 6 Gb/s (for a 10 Gb/s link). As could be expected, larger buffer sizes minimize loss.

This material is based on work supported by the National Science Foundation under Grant No. 0520081.

0742-1303/07 $25.00 © 2007 IEEE
DOI 10.1109/LCN.2007.96

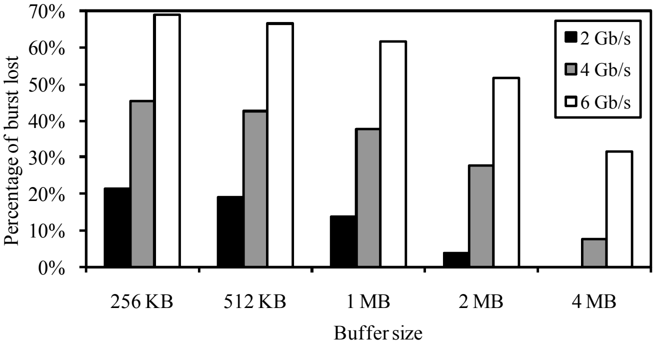

Figure 2. RPS-induced burst loss as a function of buffer size

To understand RPS-induced packet loss, key open questions are:

1. What is the probability of RPS-induced burst loss occurring?
2. What is the impact to higher-layer protocols and applications of RPS-induced packet loss if it occurs?
3. Can RPS-induced packet loss be reduced or prevented?

We address questions (2) and (3) in this paper with an experimental evaluation. Question (1) is future work.

III. Experimental Evaluation of RPS

RPS has not yet been implemented, so it cannot be directly studied. In this paper, the effect of an RPS switching time is emulated using IEEE 802.3x PAUSE flow control. Some of the results have already been presented to the EEE study group [1].

A. *Emulating RPS with PAUSE Flow Control*

PAUSE flow control is part of the IEEE 802.3 Ethernet standard for full-duplex operation. PAUSE flow control allows one end of a link to apply back pressure to the other end of the link to temporarily stop packet flow. PAUSE is implemented with a MAC fame. A PAUSE flow control MAC frame sent from a desktop PC to a LAN switch will block the link for the period of time set in the PAUSE opcode field. The maximum possible pause time is 65535 pause_quanta (where one pause_quanta is 512 bit times), which is 33.6 ms for a 1 Gb/s link. At the end of the pause time the link returns to the same data rate, so PAUSE can only be used to emulate the RPS resynchronization delay and not the actual changing of link data rate. The exact time required for 1 Gb/s link resynchronization is unknown, but is expected to be in the range of 1 to 20 ms (from discussions at the January 2007 IEEE 802.3 EEE study group meetings [5]).

B. *Description of file transfer experiments*

Experiments were conducted to evaluate the effect of RPS switching on file transfer using both TCP and UDP. Figure 3 shows the experiment configuration. The client was a Dell OptiPlex GX620 with a 2 GHz Pentium 4 processor and 1 GB of RAM. The server was a Dell OptiPlex GX270 with a 2 GHz Pentium 4 processor and 1 GB of RAM. The client had a Broadcom NetXtreme 57xx Gigabit Controller NIC and the server an Intel Pro/1000 MT NIC. The operating system in both hosts was Windows XP with service pack 2. The LAN switch was a 5-port Linksys EG005W workgroup switch. The link data rates were 1 Gb/s in all cases. All equipment was co-located within a laboratory bench.

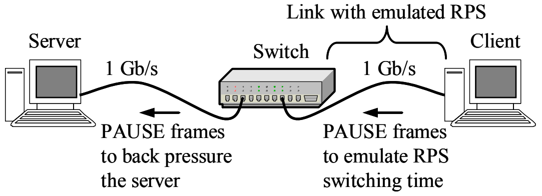

Figure 3. Experiment configuration

The NPG raw send program [8] running in the client was used to generate PAUSE MAC frames to emulate RPS switching events. The NPG program has the ability to specify a repetition time for a packet. For TCP file transfer, a custom web server program (weblite [2]) was used in the server and a custom program to get files from a web server was used in the client. The file get program in the client was instrumented to measure file download time to an accuracy of 10 ms. For UDP file transfer, custom server and client programs were used. The UDP server and client programs did not detect or attempt to recover from packet loss (thus, file transfer could be incomplete). The client program was instrumented to measure file download time. For the UDP file transfer, packet loss was measured by reading the SNMP MIB variable ifInUcastPkts for the Ethernet interface in the client and comparing it to the number of UDP packets sent from the server.

For TCP file transfers, the response variable of interest was the file download time. For UDP file transfers, the response variables of interest were the file download time and number of packets lost. Control variables were:

- File size in server – fixed at 500 MB. This is a reasonably large file.
- Frequency of emulated RPS switching events – fixed at 1 per second. It is unlikely that RPS events would occur this frequently. Thus, this can be considered as an extreme case.
- Time period of emulated RPS switching events – varied as 1, 10, and 20 ms. These values represent a reasonable range for switching time. Exact times are still unknown and are a function of PHY layer characteristics.
- Response to PAUSE MAC frame in the server – varied as enabled and disabled.

The LAN switch automatically generated PAUSE MAC frames when its buffers were full. These PAUSE MAC frames were sent from the switch to the server. At the server it was possible to enable or disable response to the PAUSE MAC frames. With response disabled, PAUSE MAC frames were ignored.

Experiments for each control variable setting were replicated five times and the results averaged. Control experiments were also executed for an RPS switching time of 0 ms (i.e., no RPS).

TABLE I. RESULTS FOR THE TCP FILE TRANSFER EXPERIMENT

	PAUSE disabled in server	PAUSE enabled in server
RPS	***Transfer time***	***Transfer time***
control	15.64 s	15.64 s
1 ms	15.64	16.16
10	15.64	16.41
20	15.93	16.65

TABLE II. RESULTS FOR THE UDP FILE TRANSFER EXPERIMENT

	PAUSE disabled in server		PAUSE enabled in server	
RPS	***Transfer time***	***Packet loss***	***Transfer time***	***Packet loss***
control	28.08 s	0 %	28.08 s	0 %
1 ms	28.18	0.05	28.25	0
10	28.14	1.00	28.35	0
20	28.14	1.98	28.68	0

C. Results and observations for file transfer experiments

Table I shows the results from the TCP file transfer experiment. For TCP file transfers with PAUSE disabled in the server, the increase in transfer time was roughly equal to the total emulated RPS time for the time of transfer. For all cases, there was one emulated RPS switching time per second. So, in 15 s of transfer time there were 15 emulated RPS switches. For a 20 ms switching time this corresponds to 300 ms total emulated switching time. The increase in transfer time from the control case (15.64 s) to the 20 ms case (15.93 s) was about 300 ms. Thus, it appears that TCP was able to recover from any RPS-induced overflow packet loss in no additional time over the RPS switching time. However, for the case where PAUSE was enabled in the server, the increase in transfer time is greater than the total emulated RPS switching time. This appears to be due to some interaction between PAUSE flow control and TCP flow control. Further work is needed to understand this interaction.

Table II shows the results for the UDP file transfer experiment. For UDP data transfer there was no difference in transfer time when PAUSE was disabled in the server and the emulated RPS switching time was increased. This was due to lost packets not being recovered when UDP was used. Packet loss was roughly equal to the percentage of RPS switching time (e.g., for 20 ms RPS switching time per second, roughly 2% of transmitted UDP packets are lost). For the case of PAUSE enabled in the server, transfer time was increased by an amount equal to the total emulated switching time and no packets were lost (showing that PAUSE flow control can prevent overflow packet losses due to RPS switching).

D. Some additional experiments – web surfing and VoIP

Two additional qualitative experiments were performed in order to investigate the effects of RPS on applications. The effect of emulated RPS switching was evaluated for web surfing and VoIP (using Skype [7]). Again, the rate of emulated RPS switches was one per second and 1, 10, and 20 ms switching times were used. Experiments with two people as test subjects were conducted. Neither subject was able to detect any difference in performance between the control case of no RPS switching and the experiment case of RPS switching.

IV. SUMMARY AND FUTURE WORK

Rapid PHY Selection (RPS) is being considered by the IEEE 802.3 Energy Efficient Ethernet (EEE) study group as a mechanism to quickly switch the data rate of an Ethernet link to match link data rate with utilization. Switching the link rate causes a momentary disruption of the link which may cause packet loss due to buffer overflow in upstream switches. We emulated RPS switching time by using IEEE 802.3x PAUSE flow control. We quantitatively studied the possible effects of RPS switching time on TCP and UDP file transfer. We also qualitatively evaluated the effect of RPS switching on web surfing and VoIP. Even in the extreme case of one RPS switch per second, it was shown that the RPS has little or no perceivable effect on TCP file transfer time (a difference of 300 ms in about 15 s is not perceivable). However, further work is needed to fully understand the effect of PAUSE back pressure on a TCP server where PAUSE frames are sent as a result of RPS switching causing buffer filling. With UDP it was shown that PAUSE back pressure can prevent packet loss that result from RPS switching causing upstream buffers to fill.

ACKNOWLEDGEMENT

The authors thank Karthik Sabhanatarajan from the University of Florida for showing us the NPG program used for generating PAUSE MAC frames. The authors also thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] K. Christensen, “Rapid PHY Selection (RPS): Emulation and Experiments using PAUSE,“ presentation to the March 2007 EEE SG meeting, 2007. URL: http://grouper.ieee.org/groups/802/3/eee_study/public/mar07/christensen_02_0307.pdf.

[2] K. Christensen, “A Super Light-Weight Secure HTTP Server,” source code, 2007. URL: http://www.csee.usf.edu/~christen/tools/weblite.c.

[3] C. Gunaratne, K. Christensen, and B. Nordman, “Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying, Split TCP Connections, and Scaling of Link Speed,” *International Journal of Network Management*, Vol. 15, No. 5, pp. 297-310, September/October 2005.

[4] C. Gunaratne, K. Christensen, and S. Suen, “Ethernet Adaptive Link Rate (ALR): Analysis of a Buffer Threshold Policy,” *Proceedings of IEEE GLOBECOM*, November 2006.

[5] “IEEE 802.3 Energy Efficient Ethernet Study Group,” 2007. URL: http://grouper.ieee.org/groups/802/3/eee_study/index.html.

[6] B. Nordman, ”Energy Efficient Ethernet:Outstanding Questions,” presentation to the January 2006 EEE SG meeting, 2007. URL: http://grouper.ieee.org/groups/802/3/eee_study/public/jan07/nordman_01_0107.pdf.

[7] Skype, 2007. URL: http://skype.com.

[8] J. Todd, “Network Packet Generator,” source code, 2007. URL: http://www.wikistc.org/wiki/Network_packet_generator.