# Ethernet Adaptive Link Rate (ALR): Analysis of a MAC Handshake Protocol

Himanshu Anand, Casey Reardon, Rajagopal Subramaniyan, Alan D. George
*High-performance Computing and Simulation (HCS) Research Laboratory*
*Department of Electrical and Computer Engineering, University of Florida*
*Gainesville, Florida 32611-6200*
*{anand, reardon, subraman, george}@hcs.ufl.edu*

## Abstract

*In this paper, a handshake protocol at the Medium Access Control (MAC) layer is proposed and analyzed for dynamically changing the link rate in the Network Interface Card (NIC), adapting to network utilization, and thus decreasing average power consumption. Simulation results show that this protocol can be used to change link rate in Ethernet network devices without causing user-perceivable delays.*

## 1. Introduction

Edge devices, which are primarily comprised of desktop computers, are the largest Internet-related energy consumers. On an Ethernet link connecting a PC to a first-level LAN switch, power consumption of the link (NIC + switch) differs by approximately 4 W between 1 Gbps and 100 Mbps link rates [1]. For 10 Gbps links, the power consumption is even higher. This observation suggests that switching to lower link rates during low utilization periods will result in reducing the energy consumption in NICs. However, packet delays can be a potential trade-off when NICs operate at lower link rates. In this paper, we propose and analyze a MAC handshake protocol for dynamically switching link rates without causing any user-perceivable delays. The protocol uses a dual-threshold policy proposed in [2] for dictating when to change link rates. Whenever the buffer occupancy drops below the low threshold or rises above the high threshold, the handshake mechanism is activated.

## 2. MAC Handshake Protocol

The protocol includes an exchange of two frames to initiate a link rate change, viz. MAC control frame and ACK/NACK frame. The frame format was adapted from the existing PAUSE frame. The modifications proposed are in the opcode field (to identify the frame as a MAC control or ACK/NACK frame) and in the parameter field (to indicate the link rate). While ALR has been proposed for switching between more than two link rates, we consider only two link rates here.

The procedure followed by the protocol to switch to lower link rate can be described as follows. During the link initialization, ALR capability is advertised through Auto-Negotiation. If node A decides to change the link rate, following the ALR policy, node A sends a MAC control frame with the desired (low) rate in the "parameter" field at the high link rate. Node A starts a timer that expires if an ACK/NACK frame from node B is not received within a specified time. If the timer expires, node A sends the MAC control frame again. Once node B receives the MAC control frame, it stops sending further data frames, and checks whether it can switch to the (low) link rate requested by node A according to the ALR policy. If node B decides in the affirmative, it sends an ACK frame to node A at the old (high) rate. After sending the ACK, node B changes its link rate to the desired value and resynchronizes its clocks. After receiving the ACK, node A changes its speed to the new (low) link rate, resynchronizes its clocks, and resumes the data transmission at the new link rate. If node B decides that it cannot switch the link rate, it sends a NACK frame to node A at the old link rate. If node A receives a NACK frame, it resumes packet transmission at the old link rate again.

If node A decides to switch back to a higher rate, it sends a MAC control frame and node B always replies with an ACK in this case and resynchronizes its link. The time required to resynchronize the clocks is implementation-specific to the PHY transceiver chip. For our simulation model, we fix the resynchronization time as 512 PCI clock cycles (7.75 μs for a clock with 66 MHz frequency), which is the time required to synchronize the clock generators when switching to active state (D0) from cold state (D3) in the case of Intel Gigabit controllers.

* This material is based upon work supported by the National Science Foundation under Grant No. 0519951.

## 3. Performance Evaluation of Protocol

Simulation models were developed using the MLDesigner simulation tool to analyze the performance of the MAC handshake protocol. We modeled a system with a NIC and a switch linecard both with ALR capability. Dummy packets are fed as inputs into the models, whose sizes and inter-arrival times are derived from traffic traces.

Traces were collected at the University of Florida on a 1 Gbps link connecting a desktop PC and a switch using the Ethereal packet capturing tool. Different usage scenarios, such as typical Internet surfing (UF_SRF: sparse and bursty traffic), typical Internet surfing followed by a file transfer (UF_VAR, variable traffic), large file transfer (UF_HDT: high data transmission), and video streaming (UF_HDR: high data reception to study the transmit FIFO state at the switch line card from where the data is being transmitted) were used for trace collection.

A synthetic traffic trace (UF_SYN) was also generated by studying the characteristics of the collected 1 Gbps traces. The aim of generating this trace was to analyze the state of the output buffer and formulate a technique to reduce the high number of switches in cases of traffic that could induce frequent link rate oscillations.

The size of the transmit FIFO queue is fixed at 64 KB as in most commercially available 1000Base-T NICs. For each traffic trace, the mean packet delay is calculated as the overall average of the difference between the time at which a packet arrives at the NIC and gets stored into the queue and the time at which the packet reaches the switch. This delay value takes the queuing delay, processing delay, and propagation delay into account. The NIC includes a mechanism that checks the buffer occupancy every 0.5 milliseconds.

## 4. Simulation Results

The mean packet delay values for all the traces are shown in Table 1. The values of low threshold and high threshold were fixed as 50% and 80% of the buffer size, respectively. It is observed that the mean packet delay is not user-perceivable and does not cause any buffer overflow for any of the traces. For the typical user trace UF_SRF, no link rate switches from low to high rate were observed. Thus, the NIC could operate in the low rate for almost 100% of the time of its operation.

For the synthetic trace, we analyzed the effect of increasing the gap between the low threshold value and the high threshold value on the number of rate switches and mean packet delay. The resulting values are shown in Table 2. The idea is to make the NIC operate in a single (high or low) link rate for a considerable period of time once it has switched to that particular rate. We observed that increasing the gap between the high and low threshold values reduces the number of switches without affecting the mean packet delay by a considerable amount, while still leading to significant power savings.

**Table 1. Mean Packet Delay**

| Trace | Mean Packet Delay (ms) | | | No. of low-to-high rate switches |
|---|---|---|---|---|
| | 1 Gbps | 100 Mbps | ALR | |
| UF_SRF | 0.017 | 0.311 | 0.313 | 0 |
| UF_VAR | 0.01 | 0.251 | 0.253 | 0 |
| UF_HDT | 0.0325 | **Overflow** | 0.446 | 3 |
| UF_HDR | 0.025 | **Overflow** | 0.901 | 10 |
| UF_SYN | 0.0223 | **Overflow** | 0.832 | 46 |

**Table 2. Threshold Analysis for UF_SYN**

| High threshold (% of output buffer size) | Low threshold (% of output buffer size) | Mean Packet Delay (ms) | No. of rate switches |
|---|---|---|---|
| 80 | 50 | 0.832 | 93 |
| 85 | 50 | 0.843 | 62 |
| 86 | 50 | 0.841 | 56 |
| 86 | 15 | 0.835 | 47 |
| 86 | 10 | 0.832 | 45 |

## 5. Conclusions

In this paper, we proposed and analyzed a MAC-layer handshake mechanism to dynamically change the link rate. The method is an effective mechanism for changing link rates without causing any user-perceivable packet delays. We also analyzed the MAC handshake mechanism for traffic with periodically changing link utilization with the help of a synthetically generated trace. We observed a problem of frequent link rate oscillations and proposed a solution to this problem by increasing the gap between high and low threshold values for the output buffer.

## References


[1] 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.

[2] C. Gunaratne, K. Christensen, and S. Suen, "Ethernet Adaptive Link Rate (ALR): Analysis of a buffer threshold policy," to appear in *IEEE GLOBECOM* 2006. http://www.csee.usf.edu/~pgunarat/papers/globe06.pdf