

Congestion Control for Switched Ethernet

Gary McAlpine - Intel Corporation
gary.l.mcalpine@intel.com

Abstract

Interconnects for clusters and bladed systems must deliver efficient throughput, low latency, low delay variations and minimal frame drops. The primary technical issues hindering Ethernet adoption for cluster and blade system interconnects are the current methods Ethernet switches use for dealing with congestion, which can happen frequently under cluster and blade system workloads. The common response to congestion is to drop frames and the common method of avoiding the need to drop frames is to utilize very large switch buffers. In this paper, we propose the insertion of a simple self-managing congestion control protocol into existing communication stacks at the edges of layer 2 switched interconnects. We assert that control of the traffic flow into the layer 2 subnet is key to controlling the characteristics of cluster and blade system interconnects. We show simulation results demonstrating how the proposed protocol, coupled with a layer 2 ingress rate control function, can dynamically control traffic flow so as to maximize the throughput efficiency while minimizing the loss, delay, and delay variations.

1. Introduction

Although Ethernet is typically used as a local area network (LAN) technology, there is substantial interest in utilizing Ethernet in cluster and blade system interconnects. Ethernet is well-known, widely available, and broadly compatible. Unfortunately, the IETF and IEEE standards do not currently support the congestion management (CM) mechanisms necessary to enable Ethernet based interconnects to provide the appropriate characteristics for clusters and bladed systems. In our previous paper [1], we explored a 3 level architectural approach to CM that was designed to leverage existing transport and network layer mechanisms at level 3, add layer 2 subnet mechanisms at level 2, and leverage existing and/or new link layer mechanisms at level 1. We took this 3 level approach because the layer 2 technology of interest was full-duplex Ethernet and our primary goal was to get an appropriate set of methods and mechanisms supported by the standards. To achieve this goal, Ethernet CM would have to operate in harmony with existing and future mechanisms utilized in the standard networking stacks. It would also have to be designed so that it could be seamlessly integrated into the existing stacks and, if supported by at least some of the layer 2 components, show an improvement in the performance characteristics (or no difference at a minimum).

The IEEE 802.1 bridging [2] and 802.3 link layer protocols [3] and the IETF network and transport protocols (IP, TCP, & UDP) are at the heart of the most widely deployed and interoperable communication stacks today. Recent studies have sought to reduce TCP/IP processing overheads in datacenter environments [4][5]. Unfortunately, the streamlining and acceleration of the upper layer stacks potentially creates even more severe and more frequent congestion events in the lower layer interconnects (such as short-range 1 & 10 Gbps Ethernet subnets). New upper layer protocols like iSCSI [6] and RDMA [7] (for storage and cluster communications over TCP/IP) rely on low frame loss rates to achieve low processing overheads, high throughput, and low latencies. Unfortunately, TCP uses the rate of packet loss to gauge the level of congestion along each connection and to control transmission rates accordingly. The only standard method for Ethernet switches to signal congestion is to drop packets. High loss rates can cause a large percentage of traffic to be handled by exception processing, which negatively affects processing overheads and delays. However, the most significant impact to TCP performance is long periods of inactivity due to timeouts resulting from packet drops. And long timeouts can easily bring a cluster or blade system's performance to its knees. Since many target applications of switched Ethernet need to support switching of TCP connections where one end or both are terminated outside the local vicinity, we can't just shorten the timeout times to minimize the impact of drops.

In this paper, we narrow the focus of the research outlined in [1] to the subnet level (level 2) of the architecture. We propose a basic protocol for signaling layer 2 congestion information (L2CI) to the subnet ingress and to support self-management of ingress rate control state. We outline a basic set of functions that can be added to the interface between the upper and lower layers of the stacks to support the L2CI protocol and the use of the congestion information to dynamically control the traffic flow into the layer 2 subnet. We use simulation results to demonstrate how effectively these functions can be utilized to maximize throughput efficiency and minimize loss, delay, and delay variations. And, we demonstrate how the subnet level mechanisms can operate in harmony with existing upper layer mechanisms.

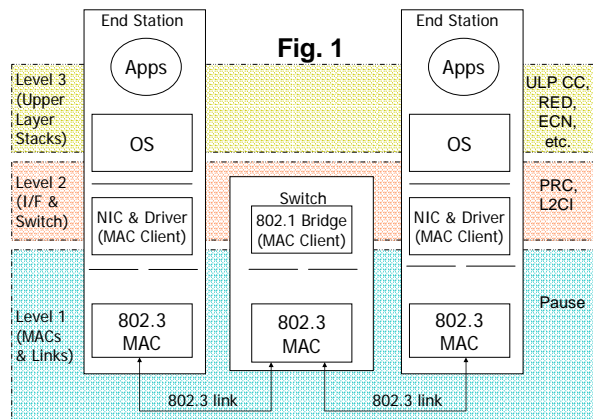
2. Related Prior Work

Prior congestion management research for long-range networks is reflected in such standards as TCP/IP [8][9], ISDN [11], ATM [12], and others. Prior research for short-

range networks is reflected in such standards as Fibre Channel [13], Infiniband™ [14], and Advanced Switching [15]. Recent research on CM for unreliable datagram services are reflected in the IETF draft on DCCP [10]. An example of recent research on CM for TCP is reflected in FAST TCP [19]. Because the list of references to congestion management research over the past 30 years is so long, we do not attempt to provide a comprehensive list for this paper. Our research is uniquely focused on enhancing short range 1 and 10 Gbps IEEE802.1/802.3 subnets with only a few hops. As such, the methods and mechanisms we can consider must be able to seamlessly integrate into the standard networking stacks.

3. Review of Our 3 Level Approach

We broadly classified congestion control mechanisms as link level mechanisms, subnet level mechanisms, and end-to-end mechanisms. Link level mechanisms try to regulate the flow of traffic over each link to avoid frame discards due to transient congestion. Subnet level mechanisms try to optimize traffic flow through a subnet to avoid oversubscription of local subnet resources. End-to-end mechanisms attempt to take action at the “flow” sources or on higher layer “flow-bundles” to avoid oversubscription of network resources end-to-end.



The strategy for our research was to analyze the issues at all 3 levels, and simulate various mechanisms for dealing with congestion at each level. We developed simulation models with various independent mechanisms at each level so that we could test each by itself, as well as in various combinations with mechanisms at other levels.

Figure 1 shows the coarse structure of typical end-stations and switches and their relationship to the 3 levels of congestion control. They include 1) link level: the IEEE802.3 MAC and link layers, plus the MAC Client interface; 2) subnet level: the interface between the upper layers and lower layers, plus the IEEE802.1 switching layer, and 3) end-to-end: the upper layer stacks such as those in the operating systems of servers, workstations, and routers. The primary methods we’ve tested to-date are

some of those currently supported by the IEEE and IETF standards (Pause, RED, ECN, & TCP congestion control [2][16][17]).

We tested various new link level mechanisms, but ultimately found they were ineffective when used stand-alone in multi-stage subnets. We did, however, show that the link level mechanisms (including the existing Pause method supported by IEEE802.3) could be used effectively as a fail-safe against packet drops when utilized with the higher level mechanisms¹. We ultimately decided the new level 1 mechanisms provided too little additional value to justify expending the efforts to standardize them.

We determined the key set of mechanisms needed for effectively controlling the Ethernet subnet characteristics were those at level 2. Support for such mechanisms is also conspicuously missing from the IETF and IEEE standards. In [1] we showed promising results utilizing an initial implementation of a Path Rate Control (PRC) method combined with reverse messaging of layer 2 congestion information (L2-CI) directly from switches. With our initial PRC and L2-CI implementation we were able to achieve ~85% throughput efficiency, but had to utilize ECN at level 3 and link level rate control at level 1 to prevent the overflowing of NIC and switch queues. Since our goal was to support layer 2 congestion management with both TCP and UDP, as well as with other transports and upper layer protocols, we couldn’t very well require the use of ECN with PRC. So, the research that followed the previous paper focused on refining the PRC and L2-CI messaging to improve their combined performance and enable them to operate independent of, but harmoniously with, existing and future level 1 and 3 mechanisms.

4. Path Rate Control (PRC) Interface

Path Rate Control adds 3 basic functions to the interface between the higher layers and lower layers (figure 2): 1) A L2CI Protocol Function for generating and receiving path discovery and congestion feedback messages and for maintaining path congestion and state tables; 2) Path Congestion and State Tables for interfacing path specific information to a PRC function; and 3) A PRC function that supports dynamic scheduling of higher layer flows or flow bundles into the lower layer transmit queue(s), based on path specific congestion levels. The PRC interface is structured to support implementations where the layer 2 side may be implemented primarily in hardware and the higher layer side may be implemented in hardware, firmware, or driver level software. It assumes the higher layer side can utilize existing address translation tables to associate flows with paths. (In our simulations, a path is

¹ Note that RED and Pause are not very compatible in that RED drops packets to signal congestion and Pause tries to prevent packet drops.

