Infrastructure Optimizer
Modeling of Interconnect Components

HyPerformix, Inc.
4301 Westbank Drive
Building A, Suite 300
Austin, TX 78746
www.hyperformix.com

Abstract
This whitepaper presents an overview of the HyPerformix Infrastructure Optimizer™ interconnect models.

Introduction

A distributed information system is constructed with various data networking techniques that include local area networks (LAN) and wide area networks (WAN). Routers and LAN switches are generally used to connect together a number of LANs or to connect a LAN with a WAN. The features which distinguish a router from a switch are significantly blurred in contemporary products. A traditional LAN switch (henceforth to be called a switch) uses a layer 2 address (in the OSI model) to forward a layer 2 frame from one LAN to another. A traditional router uses a layer 3 address to decide how to forward a layer 3 packet from one hop to the next during the transfer of a packet from the source to the destination. Recently introduced routers and switches, however, can use information pertaining to OSI layers 2 – 7. Hence, both switches and routers are used in a distributed information system.

A bridge is used to interconnect LAN segments. There are two types of bridges: transparent and source routing. A transparent bridge is functionally identical to a switch. Source routing bridges are used primarily for the interconnection of token ring LANs. With source routing, a LAN station ascertains the route to be followed by a frame to each destination before any frames are transmitted. This information is inserted in the header of the frame and is used by each bridge.

The objective of this paper is to present information on how Optimizer models network interconnects, and to discuss the performance issues of routers and switches in the context of an Optimizer model of a distributed information system.

Routers, Switches and Bridge Components

The router, switch and bridge components have the following parameters: name, configuration, cost and performance parameters. The name of each component must be unique within a model. The cost is used to represent a monetary value and is only used in model documentation. The configuration provides a drop-down list of pre-defined performance characteristics of commonly used products or the user can define new configurations.

The performance parameters, which include simultaneous streams, throughput, and background load, may be set individually by the user. The *simultaneous streams* value is an integer and indicates the number of packet streams that can be concurrently forwarded by the router. For a router or switch to have simultaneous streams value greater than 1, it must be capable of processing more than 1 packet simultaneously. The *throughput rate* is the rate at which packets are forwarded from one network interface to another, stated in units of packets per second. The *throughput rate* value is the maximum throughput value per stream. Often, we see maximum throughput rates being published by router vendors without any mention of the frame size and network type for the given data. In such cases, it is advisable to assume that the network type is Ethernet (of bandwidth 100 Mbits/sec) and the frame size is 64 bytes since this scenario, in general, permits measurement of the inherent maximum packet forwarding ability of the router. Vendors typically specify throughput as a value corresponding to the aggregate throughput; i.e., number of simultaneous streams multiplied by maximum throughput per stream. It is important to read vendor specs carefully. Finally, the background load is specified as a percentage in the Optimizer GUI and is assumed by the model to affect each active stream equally.

Simulation of Packet Processing

A packet transferred across a LAN is encapsulated in a LAN frame. During simulation, Optimizer models several different types of queuing delay, which are described in the following section. In addition, the packet forwarding delay is defined by the following formula:

```
Packet Forwarding Delay = 1 / (t * (1 - b)),
```

Where 't' is the per-stream throughput and 'b' is the background load, expressed as a fraction.

The user may adjust this forwarding delay on a per packet, or per message basis. The forwarding delay of all packets through an interconnect may be adjusted by changing the throughput rate of the device. However, the modeler may also increase or decrease forwarding delays of selected packets or messages to represent various types of processing that may occur in a router or switch. For example, if the user wanted to represent the overhead in performing data compression on some packets, he may increase the forwarding delay for all packets in that message.

Simulating Queuing in Interconnects

In Optimizer, interconnects have infinitely long input port queues. A two-tier queuing hierarchy arbitrates service among competing input ports (NICs) and among competing packets within a NIC. An input NIC is selected by a forwarding stream in round-robin fashion. When a NIC is selected, it can transfer up to 10 packets before relinquishing control to the next NIC. Packets arriving to a particular input port (NIC) are processed First-Come-First-Serve whenever that particular NIC is selected to transmit.

Interconnects also have infinitely long output port queues - although, these output ports are implemented as input ports to the connected datalink. Therefore, congestion building at the interconnect's output ports is actually reflected in the population of the connected datalink. These output port queues may be configured to process packets First-Come-First-Serve (the default) or strict Priority basis. Packet priority is initially set at the source. The priority is inherited from the thread initiating the transfer. The user may override this default setting through network policy processing at any interconnect along the path.

Simulation of Flow Control Features

In simulating the flow of a packet along its route, Optimizer takes into account standard transport layer features that impact packet flow and network performance (i.e. TCP Windowing). However, it ignores features that have insignificant impact on network performance. For example, packet loss detection and retransmission of lost packets in TCP are not simulated. Thus, interconnects simulated in Optimizer will *not* drop packets due to network congestion or bit errors. While packet loss does happen in practice, it's very rare during normal operation of a well-designed network. Thus, it has insignificant impact on performance. It is appropriate to state here that available Optimizer model outputs include packet population statistics of network devices which can be used to detect the build up of congestion in a network. Steps taken to avoid congestion in the network will result in avoidance of packet loss.

Simulation of Packet Delivery

The path a particular packet takes through the network may be determined at the source or at each *hop* or interconnect unit while en route. The information used to determine the path is maintained in routing or forwarding tables. Network control packet traffic which is normally exchanged between interconnect devices to build and maintain these tables is not simulated in Optimizer. At the beginning of simulation, Optimizer processes the topology diagram of the model and constructs a routing table that contains the least time route between any two datalinks.[1]

In Optimizer, while processing an ADN 'Send' statement or remote IO operation, the path from source to destination is configured at the source. All packets pertaining to the same message will travel along this same path. The modeler, however, may override this path determination at any interconnect in the path. Such customization includes changing the routing path for all packets in the message or just selected packets, traffic balancing among several alternate paths or datalinks, and even changing the destination of the packets.

Packet size is determined at the source as the smallest of the maximum payload size permitted in the datalinks in the default route from the source to the destination. This is in accordance with the path MTU protocol used in the Internet. Routers encountered along the path will not fragment large packets into smaller ones. IP fragmentation is unlikely to occur on interconnects when TCP is used because TCP tries to avoid it by fragmenting packets at the source. In the case of UDP, most packets are expected to be small, the exception being NFS. Packet fragmentation is rarely a significant contributor to end-to-end performance. If file servers and clients are located on the same LAN or only traverse LANs of the same type (e.g., Ethernet), NFS traffic would not normally pass through routers or need to be fragmented if it did.

Some switches provide two alternative forwarding techniques: cut-through and store-and-forward. The store-and-forward technique requires that the entire frame is completely received before its destination interface is determined. In the cut-through forwarding technique, the destination network interface is determined after receiving a portion of the frame (often 64 bytes). Then the frame is immediately forwarded to the destination network interface. Thus, frame reception and transmission is conducted in parallel after receipt of the initial portion. One disadvantage with the cut-through technique is that frames with errors are transmitted by the switch since an error in a frame cannot be detected until it is completely received. On the other hand, the store-and-forward technique detects frames with errors and discards them prior to being forwarded to the destination interface. Switches are configurable to use the cut-through technique or the store-and-forward technique or dynamically use one or the other depending on the error rate. Cut-through forwarding may significantly improve forwarding delay performance of the switch, depending on the nature of frame traffic. However, cut-through forwarding can not be used when forwarding packets from a slower source datalink to a faster destination datalink. Optimizer models only the store-and-forward feature of switches.

Simulation of Load Balancing Switches

Many interconnect vendors offer specialized load balancing switches and routers. These load balancing products can balance among servers within a site; examples include: Cisco's LD (Local Director), and CSS (Content Services Switch), and F5's Big IP.

[1] For information on how this is done, please refer to the "Message Routing" section of the "Infrastructure Optimizer Modeling of Networks" whitepaper.

The concept of virtual server groups is critical to load balancing with interconnects in Optimizer. In the "real" world, the user's packet is directed to a virtual IP address. This IP address is provided by DNS based on the lookup information stored for a particular machine name – a name that is virtualized. The IP address supplied is actually that of a load balancing switch or router. In Optimizer, this virtualization is possible for processes attached to dimensioned server components. Modelers may direct a message to the process group (that is, without regard to the particular server process instance), and allow Optimizer to perform the load balancing automatically. Optimizer uses a round-robin algorithm by default to distribute the load. Modelers may also choose to control the load balancing by defining a network policy for load balancing. The Optimizer GUI offers the user several algorithms, and allows the user to define a custom algorithm. When the load balancing policy is assigned to a particular switch or router, this component, like the IP address provided in the real world by a DNS server, becomes the temporary packet destination. Once the packet reaches this switch, the load balancing algorithm is applied, the final destination is set, and the packet is forwarded on to that destination.

Another type of load balancing applies to network components. Balancing network traffic among multiple datalinks connecting two interconnect devices, or across multiple alternate paths is also possible in Optimizer. Exercising these features is done through the use of network policies.

Network Policies

Packet selection for customization in interconnects is performed through the definition and processing of network policies. These policies are analogous to policies found in many Policy Based Management Tools (PBMT) like Checkpoint Floodgate or HP Policy Expert. The policies state conditions for selecting packets and special network processing actions for these selected packets. Packets may be classified according to a number of selection criteria, including: source, destination, protocol, size; priority, and more. The special processing could include customized routing path specification, load balancing, prioritized routing and more. These policies may be applied to individual packets or to entire messages. All network policy definitions are global in the model but are individually assigned to specific interconnects in the network. These interconnects will examine packets passing through them to see if any assigned policies apply. The granularity of this examination and processing is set by the user. To optimize simulation time, the user may elect to perform this processing once per message, or for all packets in a message.

Finding Answers to Interconnect-related Questions

Identifying Bottlenecks

In most systems, interconnects are rarely the leading cause of delay in end-to-end performance. Arriving packets are throttled by the bandwidth of the incoming datalink – that is, the performance characteristics of the datalink govern the offered load to the interconnect. Departing packets are likewise governed by the outbound datalink.

A term commonly used to characterize a router or switch is "wire-speed" performance. This phrase indicates that the router or switch is capable of forwarding as many packets to a network per second as the network bandwidth will permit – as limited by the speed of network interfaces (NICs) supported by the device. Thus a router delivering wire-speed performance for a particular network type will not be a bottleneck in connecting to a network of that type. Most routers will be able to deliver wire-speed performance for low bandwidth networks such as 56 Kbits/sec point to point link or 128 Kbits/sec ISDN line. However, many routers are unable to deliver wire-speed performance for gigabit Ethernets. ASIC-based hardware routers can deliver wire-speed performance for multiple gigabit Ethernet interfaces.

However, it is often observed in client-server architectures that many dispersed clients send traffic along many subnets to converge on a central server. This puts a heavy demand on the server's nearby router(s). As a result, routers monitor themselves for impending congestion and invoke a defensive strategy to mitigate it – they notify the sourcing TCP session. This is intended as a flow control mechanism to get the client TCP to slow down.

This mechanism, however, is not fool-proof. Packets get dropped by routers when their buffer capacity is exhausted. TCP has a recovery mechanism when it detects this (via a timeout / missing acknowledgement) - packet retransmission. Significant delays can be experienced under these circumstances.

Although Optimizer does not simulate these flow control mechanisms, Modelers can check for router congestion occurrences by collecting population statistics on interconnects and outbound datalinks. Packet queues will build in interconnects if the arriving packet rate exceeds the capacity of the interconnect to process them (throughput is too small or insufficient number of data streams). Large datalink queue lengths may occur if there is a traffic pile-up in an attached router or switch.

Studying Future Systems

If you are doing a performance study of a future system, it is advisable that you characterize a router with the maximum throughput value that can deliver wire-speed performance. You can then slowly increase the maximum throughput and repeat your simulation to determine the effect of higher maximum throughput on your system.

Studying the Trade-offs of Data Compression

Almost all routers that support interfaces to wide area networking links have data compression features that can be turned on or off. Data compression is a good technique to increase data throughput through a wide area networking link between two routers from the same vendor (in some cases the same model). Since there are many data compression techniques, the extent of compression varies greatly.

If you are using data compression, you should carefully consider its effect on the model. In general, you can incorporate compression in the model by increasing the bandwidth of the wide area networking link. However, data compression may reduce the maximum throughput of a router if the router CPU is performing both packet forwarding and compression. Some routers have compression hardware that offloads the compression work from the CPU. In such cases data compression may not impact maximum throughput. Thus, incorporating data compression in a model requires getting information from the vendor on how data compression is being done and how much compression is being achieved.

Further information

Two other documents that supplement this paper are available. They provide a higher level view of how networks and data links are modeled in Optimizer:

1. Infrastructure Optimizer Modeling of Networks
   http://www.hyperformix.com/whitepapers/modeling-networks-whitepaper.pdf
2. Infrastructure Optimizer Modeling of Data Link Components
   http://www.hyperformix.com/whitepapers/modeling-datalinks-whitepaper.pdf

