2011

An Analysis of flow-based routing

Jennifer Casella

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Rochester Institute of Technology

B. Thomas Golisano College of Computing and Information Sciences

Master of Science in Networking and System Administration

Thesis Approval Form

Student Name: Jennifer Casella

Thesis Title: An Analysis of Flow-Based Routing

Thesis Committee

Name                     Signature                     Date

Sumita Mishra
Chair

Sharon Mason
Committee Member

Luther Troell
Committee Member
An Analysis of Flow-Based Routing

By

Jennifer Casella

Committee Members:
Sumita Mishra
Sharon Mason
Luther Troell

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Networking and System Administration

Rochester Institute of Technology

B. Thomas Golisano College of Computing and Information Sciences

2/10/2011
Thesis Reproduction Permission Form

Rochester Institute of Technology

B. Thomas Golisano College of Computing and Information Sciences

Master of Science in Networking and System Administration

An Analysis of Flow-Based Routing

I, Jennifer Casella, hereby grant permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction must not be for commercial use or profit.

Date: ___________ Signature of Author: __________________________
Abstract

Since their development in the early 1970’s, the underlying function of IP routers has not changed – they still support a best effort delivery method in order to pass frames from source to destination. With the advent of newer, bandwidth intensive Internet-based services and applications, such as video conferencing and telemedicine, many individuals wonder if the current approach to routing is the most practical. “The Internet needs to provide quality of service (“QoS”) as predictably as conventional circuit switching networks. Although some QoS capabilities in an isolated environment have been demonstrated, providing end-to-end QoS at a large scale across the Internet remains an unsolved problem [1].” The alternative to the traditional method of IP routing is a concept known as flow-based routing, whereas traffic is sent across the network as part of a common flow, rather than individually inspecting each packet.

As part of this thesis, the differences between flow-based routing and the current standard of IP routing will be investigated. There are many benefits to be had from routing based on flows, for both routers and applications. Some research has already been done on specific aspects of flow-based routing, but because the concept is so cutting-edge, resources are scarce. This study delves into the benefits and obstacles of flow-based routing, and analyzes characteristics such as practicality and security, along with the benefits of this model.
Table of Contents

Abstract ................................................................................................................................. 4
Table of Contents .................................................................................................................. 5
Introduction .......................................................................................................................... 6
Overview and Weaknesses of Traditional IP Routing ......................................................... 8
Overview of QoS .................................................................................................................. 11
History and Downfalls of Packet-Based Service: ATM ....................................................... 13
What is Flow-Based Routing? ............................................................................................. 15
   Flow-Based Routing and Resource Reservation .............................................................. 17
Characteristics of a Flow ...................................................................................................... 19
   Storing Flow State Information ..................................................................................... 20
Hardware Requirements for Flow-Based Router ............................................................... 21
   Software Requirements for Flow-Based Router ............................................................. 23
      Distributed Routing Algorithms ................................................................................. 24
      Flow Routing Table .................................................................................................. 24
      Flow-based Routing in IPv6 ....................................................................................... 24
Benefits of Flow-Based Routing ......................................................................................... 26
   Cisco Express Forwarding vs. Flow Based Switching .................................................... 27
Security Considerations for Flow-Based Routing ............................................................. 29
  OPNET Modeler 16.0 Simulation ..................................................................................... 30
     Configuration of Testing Environment ......................................................................... 33
     Project 1: Flow-based Routing ................................................................................... 33
     Scenario 1: IP Routing with OSPF .............................................................................. 36
     Scenario 2: IP Routing with OSPF and QoS ............................................................... 42
     Scenario 3: Flow-based Routing ................................................................................ 45
     Roadblocks .................................................................................................................... 55
  OPNET Modeler 16.0 Simulation Analysis/Results .......................................................... 57
     CPU Utilization ............................................................................................................ 58
     Traffic Dropped (packets/sec) .................................................................................... 63
     Background Traffic Delay ............................................................................................ 64
     Processing Delay ......................................................................................................... 65
     IP Processor: Forwarding Memory Free Size .............................................................. 67
     Survivability Analysis .................................................................................................. 68
Conclusion ........................................................................................................................... 71
Future Direction .................................................................................................................. 72
References ........................................................................................................................... 74
Introduction

Current services such as voice and video require high amounts of bandwidth and Quality of Service for reliable delivery of time-sensitive traffic. It is primitive that the delivery method, bandwidth consumption, and control of latency/delay in packet transport are improved upon from routers' current capabilities. “All these problems can be solved with no change to TCP/IP by routing flows rather than packets” [2]. A new method to avoid the redundant inspections that routers perform on packets travelling to the same destination is the first step in changing the way packets are routed across networks. A flow-based solution is capable of recognizing flows – state information is stored about the initial packet, which is then used to switch the remaining packets in the flow. “The unique level of data obtained in flow-based routing, such as flow length, rate, delay variation and other parameters, enable a number of new network benefits” [4].

State information is created on-the-fly and is capable of being deleted without any additional signaling intervention [3]. The ability to route incoming packets based on pre-determined information provides a certain level of predictability. For this reason, networking devices are able to route several different flows simultaneously. “Leveraging flow state information allows the design of novel congestion control schemes that are more efficient at improving network-level behavior” [3]. The benefits of this design from a QoS standpoint are huge. Ensuring that packets are not dropped randomly would improve the quality of applications and services that cannot tolerate loss. It would also enable the routing protocol to reach its maximum transmission rate more quickly. Traffic is controlled in TCP with
the concept of windowing, which requires that a sending device slow down if it is transmitting too much data or if a packet was lost; however, if packets were never lost, the speed of transmission of a flow would increase. Many different approaches to a flow-based technique have been experimented with in the field, each introducing their own methods and benefits.

It would be a great advancement if routers could inspect packets and store routing information from only the first packet in a flow; then route all packets that follow based on that criteria. Router resources would be more efficiently used; time would be saved in link failure recoveries by storing a primary and alternate route in the router’s memory; we would save on power bills, because routers’ energy consumption would decrease; and most importantly network devices would be able to support the service needs of increasing application demands.
Overview and Weaknesses of Traditional IP Routing

The primary function of the Internet Protocol (IP) is to route packets from source to destination. Routers are responsible for the processing and forwarding of traffic through various paths, pre-determined by routing protocols (i.e. EIGRP, OSPF, BGP). In order to maintain information about directly connected and remote networks and make forwarding decisions, a router maintains a routing table, which is populated with network information learned by neighboring devices. The routing table will typically contain routes to different destinations, making note of their associated metrics. The routing table entries are highly aggregated with today’s BGP backbone routers have upwards of 350,000 routing table entries. Although this does provide some advantages such as reduced router memory and less updates to neighbors, it does have a cost – “to look up a packet’s next hop, we need to find the longest prefix matching the header, which is a more complicated operation than a simple index into a table” [5]. When a router needs to forward a packet, it will compare information found in the IP header with entries in its routing table. “This simple model allows IP routers to be stateless: a router does not need to know anything about the potentially large number of individual connections passing through it; it simply forwards each IP packet based on the destination address contained in the packet header. [5]” Once the appropriate route is found, the packet will be transmitted out the associated interface to reach the destination network (a default route or default gateway will be used if no route exists for a particular network).
One of the major disadvantages of traditional IP routing is per-packet inspection – each packet needs to have its IP header information compared with the routing table before the router decides which interface to forward the packet out of. This process is costly in terms of routers resources and power utilization (as compared to routing based on flows). The best effort forwarding capabilities are only advantageous when the network is not congested and the application transmitting traffic is not sensitive to delay. Also, if a router’s buffer starts to overflow, the default operation is to drop the excess packets – this could be very disruptive for drop-sensitive applications.

Traditional IP routing is designed around the assumption that there is no performance monitoring to maintain the integrity of the network – the intelligence of the network is built into the end stations. Although this design reduces complexity at the routers’ level, it also creates an environment that is predicated on unreliable, best effort delivery of packets. “The spectacular success of the Internet and the best effort paradigm is one of the defining events of the present period. Most service providers offer excellent quality of service simply by keeping link utilization low (less than 50%, say) [6].” The absence of reliability forces an environment that is vulnerable to packet loss, data corruption, and duplicate or out of order arrival of packets. Furthermore, “current IP networks do not meet the requirements of the future integrated-service networks that carry multimedia data traffic with a high Quality of Service (QoS) [8].” IP routing has no concept of resource reservation, a necessity for guaranteed end-to-end performance, to ensure the transmission of application data without delay, jitter, or loss. This results in packets not reaching their destination in
a timely manner; which could have negative impact on the application’s performance.
Overview of QoS

Although QoS is not the main focus point for this thesis, a general understanding of its concepts and purpose is necessary. “Quality of Service (QoS) refers to the capability of a network to provide better service to selected network traffic over various technologies, including Frame Relay, Asynchronous Transfer Mode (ATM), Ethernet and 802.1 networks, SONET, and IP-routed networks that may use any or all of these underlying technologies [7].” QoS aims to provide several different features to improve end-to-end network service including, but not limited to: dedicated bandwidth, improvement of loss characteristics, and traffic shaping. “QoS is mainly a question of dealing effectively with overloads arising either from planning oversight or capacity reduction due to equipment failure [6].”

There are three different types of QoS service available for deployment within a network: best effort, Integrated Services (IntServ) and Differentiated Services (DiffServ). Best effort, as discussed in the previous IP routing section, is an unreliable, non-guaranteed delivery method for network traffic. One example of this is FIFO queuing – First In, First Out. The IntServ model requires that the application initially requests a certain level of service from the network, and confirms that it is available before data is transmitted. The Resource Reservation Protocol (RSVP) is one IntServ implementation that is widely deployed. Applications that are capable of supporting RSVP will send PATH messages, which are propagated throughout the network. If a network device supports RSVP and has the capacity to accommodate the application’s request, it will respond with a RESV packet containing the end-to-end flow specifications. One downside of the IntServ QoS
model is its ability to scale – it gets increasing difficult to keep track of reservations as more routers are exist between source and destination. Lastly, the DiffServ model supports differing QoS requirements and does not require the application to signal its requests to the network routers before transmitted data. Routers will inspect each packet sent by an application to determine what level of service it is requesting – this is done by analyzing the IP Precedence, Differentiated Services Code Point (DSCP), or Explicit Congestion Notification (ECN) values, all part of the Type of Service (ToS) field. “The network uses the QoS specification to classify, mark, shape, and police traffic, and to perform intelligent queuing [7].”

No network is too small to take advantages of the benefits that QoS has to offer. Configuring QoS on a network enables administrators to maintain “controlled and predictable service for a variety of networked applications and traffic types” [7]. The ability to control how much and which network resources an application can use ensures that mission critical and delay-sensitive traffic gets priority over non-critical traffic. Having this type of control over the network also enables administrators to support a wide variety of applications with the least amount of service interruption. As more and more applications are developed that depend upon network resources to be available to them, alternate ways in deploy QoS much be researched. “All current communications protocols have either major QoS failings (IP and MPLS limitations) or cannot support fast data calls (ATM). No conventional switches or routers have been able to provide one converged network that can support high QoS and short data calls with fast error recovery and high availability [2].”
History and Downfalls of Packet-Based Service: ATM

Asynchronous Transfer Mode (ATM) Switching was a technology introduced in the mid-1980’s that aimed to provide a high-performance, cell-oriented method of switching packets. ATM encodes data into fixed-length cells, making it ideal for running regular data traffic and real-time, delay-sensitive traffic on the same network.

“ATM attempted to solve the QoS problem and much was learned during the design of the technology, but due to the long, complex software flow state setup process, ATM could not be used for short data cells [2].” The fixed packet size associated with ATM was selected because engineers were not sure if hardware ASIC’s would be able to support packets of varying lengths. “Fixing the size lost about 20% efficiency for IP data that had packet sizes on average of 350 bytes (or greater) and short burst of packets (10-14 packets per transfer) [2]. Furthermore, the processing power required in PCs to handle the large number of interrupts introduced a steep cost, compared to cheap, dumb Ethernet NICs. “The introduction of cheap 100 Mbps Ethernet boards was the final deathblow for ATM at the desktop [2].”

A second mistake made in the development of ATM was the intense round trip call setup required in software. The time required to setup calls limited the call setup rate to “about 3% of the rate needed for TCP/IP data, thus relegating all data traffic to be trunked (not switched) [2].” This resulted in the required presence of both a packet router and ATM switch at nodes – a large duplication of effort, and
high cost. Although not as huge a success as some may have thought, ATM surfaced a new, advanced development— a packet-switching technology, capable of switching traffic and satisfying QoS requirements of voice/video applications simultaneously.
What is Flow-Based Routing?

Flow-based routing, or QoS routing, “is the routing technique where packets are routed from source to destination, selecting the path that satisfies all QoS (i.e., Bandwidth and Delay) requirements [10].” Unlike the stateless IP model, a stateful routing device is able to “implement more sophisticated routing and packet scheduling by using additional packet header fields and by maintaining information about flows and applications” [5]. Several attempts have been made to incorporate QoS into the current Internet service model. “Although there have been advances in the solution proposals for network-level QoS provisioning, the models such as Intserv and DiffServ are far from being deployed in the Internet. The reluctance to the adoption of these models is tied to the changes that are required in the networking infrastructure [9].” Flow-based routing extends the current routing model in several ways, including, but not limited to: multiple paths to the same destination can be calculated for different flows; and traffic can be re-routed on-the-fly from its current path to one of its alternate paths.

Routing based on flows enables a router to manage traffic data streams as separate flows. Flow-based routers have the capability to improve the QoS, performance, and utilization of existing IP networks by keeping track of flow-state information. By maintaining a flow database, the IP router would be able to support “rate guarantees, delay guarantees, and eliminate packet loss for flows of any size, while still maintaining high line and trunk utilization” [2]. After a router has collected and stored all state information, any packets belonging to the same flow will be routed on the same route. “The route need not be the same route for all flows
An Analysis of Flow-Based Routing

Jennifer Casella

16

An Analysis of Flow-Based Routing

An Analysis of Flow-Based Routing

Jennifer Casella

16

to a given destination since each flow can now be kept in order by being kept on its own route [2].” Furthermore, when routing decisions are made for a particular flow an alternate, backup route is also decided upon so the flow can be rerouted in the event of a hardware failure on the device.

Flow-based routing is also capable of offering a variety of enhancements including guaranteed bandwidth, fast error recovery, and dynamic load balancing. “The method by which a flow router achieves a guarantee is to compare the rate of the packet arrival in the state block to the guaranteed rate and discard if the user exceeds the agreed rate by some burst tolerance [2].” Applications are capable of transmitting traffic up to the burst limit, but on the average, the transmission rate is controlled and the need to do this is minimized.

As previously mentioned, fast error recovery is achieved by storing multiple backup, alternate routes in the event of link failure. If a failure in the path occurs, the first packet in the flow to reach the failure point will be sent back to the source – this indicates a failure, and that this packet and all succeeding packets should be transmitted over an alternate route. The flows that were being transmitted across the original path will be mutually distributed across various alternate paths. TCP traffic will be reduced by selective discard if the load on the newly directed output port is too high. “The guaranteed traffic and UDP traffic will not be affected so long as there is sufficient TCP traffic to absorb the overload [2].” In this manner, link utilization can be maximized until a failure happens.

“With near-equal cost route information, a router can distribute the load over all near-equal paths based on the current load information” – this is how flow-based
routing satisfies dynamic load balancing [2]. Since each flow will be associated with its own route, different flows can reach the same destination via different paths. Continuously measuring the load on each port will ensure that dynamic load balancing selects the best path for each flow – it enables flow-based routing to achieve over 80% of link utilization on all paths/ports. Furthermore, flow-based routers are able to route flows across different paths without causing issues with packets being out of order. “This gain can be very significant and any such gain correspondingly reduces the total network cost [2].” The following figure illustrates the forwarding differences between traditional and flow-based routing:

![Flow-Based Routing and Resource Reservation](image-url)
There is a distinct difference between flow-based routing and resource reservation operation performed by protocols such as RSVP. “While resource reservation protocols provide a method for requesting and reserving network resources, they do not provide a mechanism for determining a network path that had adequate resources to accommodate the requested QoS [13].” Flow-based routing provides the ability to select a path that can guarantee QoS requests, as well as reserving the resources required along the path. Implementing flow-based routing with the added capabilities of resource reservation protocols provides for the most control over both the path and resources needed to satisfy QoS requirements.
Characteristics of a Flow

“A flow is defined as the stream of IP packets between a particular source IP address and port going to a unique destination IP address and port, all of which packets are using the same protocol [11].” Examples of individual flows include voice calls, streaming video, file transfers or web access. It is important for a device to have an understanding of what a flow consists of so it can cache only the information necessary to identify it.

Within the IPv4 packet header, there are five fields that identify a flow: protocol, source address, destination address, source port, and destination port. Likewise in IPv6, there are three fields: flow label, source address and destination address. Creating a hash of these values for either version of IP will yield the Flow ID that is stored in the flow routing table. The following diagram illustrates these fields:
Enabling a router to manage flows at input will result in the absence of an output queue, reducing delay and loss in the network.

**Storing Flow State Information**

The flow routing table will store information related to each flow so the router can eliminate lookups and making faster decisions to route packets. Where a flow is defined generally as the source and destination, each entry in the flow routing table will contain the source and destination information, in addition to an action associated with the flow entry. “The routing action typically is the address of the next hop router to which the packet should be sent; in some applications, the action could also take the form of “do not forward the packet” which is useful for access control (i.e.: an ISP may not permit certain flows to pass through its network) [5].”

One example of the flow table structure may be the following:

<table>
<thead>
<tr>
<th>Label</th>
<th>Source Address</th>
<th>Destination Address</th>
<th>In Port</th>
<th>Out Port</th>
<th>Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>192.168.0.1</td>
<td>192.168.0.10</td>
<td>Ge1/8</td>
<td>Ge1/10</td>
<td>1870</td>
</tr>
<tr>
<td>Y</td>
<td>172.16.8.10</td>
<td>172.16.8.19</td>
<td>Ge2/9</td>
<td>Ge1/8</td>
<td>3500</td>
</tr>
<tr>
<td>Z</td>
<td>10.10.4.4</td>
<td>10.10.4.5</td>
<td>Ge2/18</td>
<td>Ge3/18</td>
<td>800</td>
</tr>
</tbody>
</table>

Each entry in the flow table is identified with a flow label – a unique value that is the result of a hash of the source IP address, destination IP address, source port number, destination port number, and protocol. When a flow enters the router, its flow ID will be calculated. In order to maintain the size of the flow table, each entry will be removed from the table if the flow is not actively sending/receiving traffic and its timeout has expired. As previously mentioned, a new entry will only be created in the flow table if a matching one does not already exist.
Hardware Requirements for Flow-Based Router

Flow-based routers will need to be designed with the ability to intelligently process traffic flows. Flow-based routing can be implemented on current IP routers, with the addition of memory to store the flow state table information in software. Storing information in hardware would result in faster switching of packets, but would require new Application-Specific Integrated Circuits (ASICs) to be developed and installed on to the motherboard of routers. In either case, it would require a significant change in the software architecture of the device to be able to process and route information based on flows. Some known, readily available flow-based routers include the Caspian Apeiro, and the FR-1000 developed and manufactured by Anagran.

Transitioning to a flow-based router environment does not have to require the upgrade of all routers in a network infrastructure. It is possible to integrate flow-based routers with traditional IP routers; however, there must be a designated path of flow-based routers across the network to satisfy QoS requirements of specific applications. “The high QoS part of the network need not however, have more capacity than is required by the premium real time streaming traffic, like video and voice. Thus there are many mixed network possibilities where flow routers only need to be added, as premium traffic demands their use.” The following diagram illustrates an existing IP network with a path of flow routers that can provide flow-based capabilities between the end nodes:
Software Requirements for Flow-Based Router

“Due to the worldwide acceptance of IP, it would be unreasonable not to build a new network around IP.” IP, itself, does not put any limitations on QoS, error recovery, or router availability/scalability – the problem has always been the actual router’s design [2]. A flow-based router possesses some unique characteristics that give it the ability to route traffic more efficiently, satisfying an application’s QoS requirements.

“Depending on the scope of the path selection process, an algorithm could either return the best next hop or the entire path to the destination. The first case is more similar to the traditional hop-by-hop routing and is referred to as Distributed QoS routing. The latter is termed as Source QoS routing [15].” As its name suggests, Source QoS routing requires that all path computation be done at the source. This method of QoS routing requires that all routers in the network maintain a real-time state table of the entire network’s resource availability. OSPF can actually be extended in order to support the requirements of resource availability updates, but as expected, this added functionality will amount to more overhead for the routers. Furthermore, keeping each router in the network up-to-date can result in inconsistencies across routers’ state tables. These two characteristics, in addition to path computation, limit the scalability of Source QoS routing in networks. “In distributed routing algorithms, the path computation is shared by various routers in the network. Hence, there is no computation burden on any single router in the network [15].”
Distributed Routing Algorithms

Distributed routing algorithms can be of two types – those in which routers maintain a global state, and those that do not. Distributed routing algorithms which require routers to store state information are faced with the same issues as Source QoS routing, which could lead to degraded performance – high device overhead, the need for computational resources, and the potential for inconsistencies. A flooding technique is used in algorithms that do not require storing state information.

Flow Routing Table

Flow-based routers store state information and perform bandwidth reservation/admission control accordingly. “When a new flow arrives, an entry in the state table is created; this entry will be removed after a period of time [1].” New flows in TCP traffic can be distinguished easily with the packets of the 3-way TCP handshake (SYN=1, ACK=1). In UDP flows, this process is a bit more complex – a router would need to check four characteristics of a UDP packet: source IP, source port, destination IP, and destination port. All of these values need to be evaluated against the current flow table, creating a new entry if no match is found. In order to support a flow table, routers will need to be equipped with a large amount of memory to store and look up state information. [1]

Flow-based Routing in IPv6

The IPv6 specification, RFC 2460, defines a flow as “a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers” [16]. The IPv6
protocol has a 20-bit bit header field known as the “Flow Label” – this field is used by the packet to signify that a sequence of packets is requesting special QoS by IPv6 routers. “Hosts or routers that do not support the functions of the Flow Label field are required to set the field to zero when originating a packet, pass the field on unchanged when forwarding a packet, and ignore the field when receiving a packet [16].”

The flow’s source is responsible for assigning a random value to the Flow Label field (1 to FFFFF). All packets that belong to the same flow will be sent with the same source address, destination address, and Flow Label value. “A source must not re-use a flow label for a new flow within the maximum lifetime of any flow-handling state that might have been established for the prior use of that flow label [16].” For this reason, it is required that a device store Flow Label designated in NVRAM.
Benefits of Flow-Based Routing

There are several benefits of deploying flow-based routing in modern network environments, most of them specifically related to QoS. The following chart summarizes some of the requirements of today’s traffic flows, and how flow-based routers satisfy these:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Flow-Based Router Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protect certain traffic type from other (voice vs. video, www vs. P2P)</td>
<td>• Alternative congestion control schemes for high rate vs. low rate, responsive vs. unresponsive, long-lived vs. short-lived flows</td>
</tr>
<tr>
<td>Enable ON/OFF service model</td>
<td>• Call Admission Control for UDP flows&lt;br&gt;• Per-flow pre-emption capabilities</td>
</tr>
<tr>
<td>Optimize network goodput and user response time</td>
<td>• Call Admission Control for TCP flows&lt;br&gt;• Alternative congestion control schemes</td>
</tr>
<tr>
<td>Tight jitter and guaranteed QoS for real-time applications</td>
<td>• Call Admission Control for flows with guarantees&lt;br&gt;• Guaranteed QoS capabilities for low, constant jitter even at high utilization</td>
</tr>
<tr>
<td>Network simplification (cost reduction)</td>
<td>• Increased node intelligence: dynamic resource management and granular statistics collection</td>
</tr>
</tbody>
</table>

Additionally, flow-based routing “can be incorporated into existing IP routers, without modifying the IP protocol, providing all known QoS features, without increasing the cost, while permitting the utilization to be increased to over 80%” [2].

Although quite a bit of research has gone into flow-based routing implementation and design, several questions are still raised with the introduction of the technology:

- How can routers determine the QoS capabilities of each outgoing link in order to reserve the appropriate resource on the link?
- Although we have defined what information will be stored in the flow table, how will routing decisions be made? (source-based, destination-based, flow-based)
- Will existing routing metrics be used, or will new ones be developed?
- How will the transition to flow-based routing affect router overhead?
- Will flow-based routing be able to achieve the required level of scalability?

**Cisco Express Forwarding vs. Flow Based Switching**

Cisco Systems has implemented Cisco Express Forwarding (CEF) in their Supervisor line cards for Catalyst 6500 switches. In some traditional methods of flow-based switching, the control plane is relied upon in order to forward the first packet of each flow entering the switch. This could put a large load on the switch due to the short lives of some flows. “In many customer environments, flow-based switching can impose a bottleneck on overall throughput [17].”

Cisco’s development of CEF aims at eliminating the control plane from the forwarding path of flows. The following diagram illustrates this operation:
CEF makes use of different components on the Supervisor card in order to build a topology of the network around it, which gets programmed into hardware in high performance lookup memory known TCAM. At all times, the linecard will have a complete picture of the topology and is capable of making traffic forwarding decisions quickly. The TCAM entries are updated and synced between different components at all times [17].
Security Considerations for Flow-Based Routing

“QoS-based routing requires additional security measures both to validate QoS requests for flows and to prevent resource-depletion type of threats that can arise when flows are allowed to make arbitrary resource requests along various paths in the network [13].” By analyzing the source IP, source port, destination IP, and destination port of UDP flows at the transport layer, a denial-of-service (DoS) attack could result. “Hackers can just send in as many UDP datagrams as possible. Because each flow contains one packet, the router will be swamped with the task of setting up and tearing down the UDP single-packet flows [1].” In this manner, a network can easily be brought down.

“Excessive resource consumption by an errant flow results in denial of resources to legitimate flows. While these situations may be prevented by setting up proper policy constraints, charging models and policing at various points in the network, the formalization of such protection requires work [13].”
OPNET Modeler 16.0 Simulation

“Performance management has to be viewed from an application perspective, and application transactions need to be tracked end-to-end through the entire infrastructure including client devices, routing protocols, network configurations, server architectures and other components. Integrated solutions that address this end-to-end requirement will have a big impact” [18]. Current networking technologies have become too intricate for traditional analytical methods to yield a precise understanding of system behavior. OPNET Technologies, Inc products provide networking professionals with software that gives them the ability to optimize and maximize the performance of their networks and communications. OPNET solutions have been operationally proven in many customer environments including corporate and government enterprises, defense agencies, network service providers and network equipment manufacturers. OPNET Modeler provides for a virtualized network environment that has the ability to model network behavior and can include aspects such as devices, protocols, servers and applications; “both behavior and performance of modeled systems can be analyzed by performing discrete event simulations. The OPNET environment incorporates tools for all phases of a study, including model design, simulation, data collection, and data analysis” [19]. Virtually analyzing a network allows individuals to avert problems, troubleshoot current problems, validate changes before they are implemented on the physical network, and plan for future implementations or extensions. Early virtual
implementations of a perceived network could also cut costs, optimize network functionality and reduce security risks.

“OPNET Modeler is the industry’s leading environment for network modeling and simulation, allowing for the design and study of communication networks, devices, protocols and applications with unmatched flexibility and scalability” [20]. Developed at MIT and introduced as the first network simulator in 1987, OPNET Modeler’s object-oriented approach through modeling and graphical editors emulates actual networks and their components – a system can be intuitively mapped to a model. All network types and technologies are supported in Modeler. The software has a number of key features, which makes it the industry’s best for network simulation. These include but are not limited to: hierarchical network models, comprehensive support, total openness, integrated debugger, animation, financial cost attribute for devices and geographical and mobility modeling. All OPNET products are equipped with a Model Library, which contains the majority of networks, devices and applications utilized in current networking environments. The objects in this library include Data Link layer technologies (VLANs, Token Ring, Frame Relay, Ethernet, etc), protocols (IP, OSPF, TCP, etc), support for wireless LANs, VoIP, IPv6, and several vendor device models (3Com, AMD, Cisco, IBM, Juniper, Sun, etc).

OPNET Modeler is based upon hierarchical editors that are analogous to real networks, protocols and equipment. Network models are typically centered on the project editor, which is used to create a network model comprised of items from the model library, collect performance statistics, run a traffic simulation and analyze the
results. Node and process models can be used in conjunction with this editor in order to build traffic packets, create parameters, and apply filters. The project editor also has support for subnets, as well as point-to-point, bus and radio links. The node editor is useful for modeling nodes. These node instances can be implemented into networks within the project editor. In OPNET, a node is defined by connecting modules with packet streams; an inter-module connection allows for the exchange of packets and status information. Each node’s module can serve a different purpose: generating packets, queuing packets, processing packets, etc. The process model editor is used for the creation of models that control the underlying functionality of node models. Process models are represented by finite state machines (FSMs) and are created with: 1) icons representing states and 2) lines, which represent state transitions. The link model editor gives a user the ability to create new link type objects. Each of these new link types can have different interfaces and representation. It is also possible to include comments or keywords so the link could be easily identified. In order to create new path objects, the path editor is used; path objects define traffic routes. Path objects can be used to route traffic in any protocol model that makes use of logical connections or virtual circuits. The packet format editor is used to define the internal structure of a packet as a set of fields. A packet contains multiple fields, which are represented by multi-colored rectangles, varying in size based upon the number of bits in each. The probe editor allows a user to specify the statistics that need to be collected. Many different types of statistics – global, link, node, attribute – can be collected. Lastly, in the simulation sequence editor
simulation features can be controlled. Simulation icons represent simulation sequences, which control the simulation's run-time characteristics.

Configuration of Testing Environment

Project 1: Flow-based Routing

The topology for the test scenario needed to be built out before any testing/analysis could be made on flow-based routing. The simulation was started with the creation of a new Project mode/empty scenario so a custom topology could be formed. Most companies utilize a hierarchical network design model (core, distribution, access); however, in this simulation since we are only concerns with the performance of the layer 3 devices, only routers will be used. A new project was created called ‘flow_based_routing_sim’ and several different scenarios were built off of it, to simulate different types of network environments.
An Analysis of Flow-Based Routing

Jennifer Casella
I built the scenario by selecting all available routers and adv_routers objects available in OPNET Modeler (specific vendor devices can also be selected when building out the topology).
At this point, I was presented with an empty grid, onto which I could drag routing devices on to build out my simulation topology. I elected to use a group of Cisco 7505 series routers for my scenarios. The routers were interconnected with 1 Gbps links, and three end stations were attached to the core in order to pass traffic throughout the network.

**Scenario 1: IP Routing with OSPF**

The first scenario I built from the previous topology was basic IP routing with OSPF. Each device in the network was assigned an IPv4 address. I verified that each end node had an address on a different subnet.
An Analysis of Flow-Based Routing

[Image of network analysis software interface]

Jennifer Casella
An Analysis of Flow-Based Routing

![Image of network attributes](image)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>node_9</td>
</tr>
<tr>
<td>CPU</td>
<td></td>
</tr>
<tr>
<td>VPN</td>
<td></td>
</tr>
<tr>
<td>Ethernet</td>
<td></td>
</tr>
<tr>
<td>IP Multicasting</td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>IP Host Parameters</td>
<td></td>
</tr>
<tr>
<td>Interface Information</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>IF0</td>
</tr>
<tr>
<td>Address</td>
<td>192.0.8.2</td>
</tr>
<tr>
<td>Subnet Mask</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>MTU (bytes)</td>
<td>IP</td>
</tr>
<tr>
<td>Compression Information</td>
<td>None</td>
</tr>
<tr>
<td>IPv6 Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Description</td>
<td>N/A</td>
</tr>
<tr>
<td>Layer 2 Mappings</td>
<td>None</td>
</tr>
<tr>
<td>Passive RIP Routing</td>
<td>Disabled</td>
</tr>
<tr>
<td>Default Route</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>Static Routing Table</td>
<td>None</td>
</tr>
<tr>
<td>IPv6 Default Route</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>Multicast Mode</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

[Image of network attributes](image)
The network environment was now ready to have the OSPF routing protocol deployed across the routers.
An Analysis of Flow-Based Routing

Jennifer Casella
This completed all configuration requirements necessary to build out an environment configured to route packets via OSPF.

**Scenario 2: IP Routing with OSPF and QoS**

The previous scenario was duplicated, and configured to support QoS.
There are numerous different options available for configuring QoS in OPNET environments. I elected to set up class-based WFQ, with a profile based on Type of Service (ToS). This QoS configuration was applied to all router interfaces in the topology.
An Analysis of Flow-Based Routing

![QoS Configuration]

This operation will overwrite existing QoS configurations on IP interfaces.

- **QoS Scheme:** WFQ (Class Based)
- **QoS Profile:** ToS Based
- **Apply selection to subinterfaces**

Apply the above selection to:
- All connected interfaces
- Interfaces across selected link(s)
- Interfaces on selected router(s)

**Visualize QoS configuration**

[OK] [Cancel]
This completed all configuration requirements necessary to build out an environment configured to route packets via OSPF with WFQ QoS.

**Scenario 3: Flow-based Routing**

The initial routing environment was duplicated once more to build out a flow-based routing environment.
OPNET does not have many built-in routers that supports flow-based routing, so I experimentally selected a Juniper ERX router. This edge routing platform, which runs JUNOS, has a built-in capability for both packet-based forwarding and flow-based forwarding. The stateful, flow-based forwarding of the device is performed in the following manner:

Similar to the flow-based concepts covered earlier in this document, the Juniper ERX 1410 performs the following functions to achieve flow-based performance: “The
packet treatment in flow-based forwarding depends on characteristics that were established for the first packet of the packet stream, which is referred to as a flow. To determine if a flow exists for a packet, the system attempts to match the packet’s information to that of an existing session based on the following match criteria—source address, destination address, source port, destination port, protocol, and unique session token number for a given zone and virtual router [24].” The topology used for the flow-based routing scenario was the following:

After setting up all of the topologies for testing, I also needed to configure some traffic flows to be sent through the network. I deployed an Application object,
which enabled me to send multiple types of traffic through the network simultaneously. The Application object was selected from the Object Palette, and its attributes were edited for each scenario. The application definition provides a configurable number of traffic flows. For each scenario, I configured 4 different typed of traffic – voice, video, email and HTTP.
After configuring the traffic type in the application object, an Application Profile object needed to be deployed in order to configure how frequently each traffic type would be sent during the simulation. Traffic was uniformly distributed for the entire length of the simulation.
Before running the simulation, a series of Discrete Event Statistics (DES) were configured to be collected. There are several different statistics available for collection, but I chose most of the options related to what I was looking at.
comparing (i.e. – Ethernet, IP, CPU utilization, bandwidth usage, delay, etc). The following screenshots show the Global and Node statistics configured for collection:
At this point, the simulations were configured and ready to be run in each scenario.

When running the Discrete Event Simulation, the user must select how long they would like to run the flows for, and also if there are any specific reports to be collected. In this case, I ran my application data for 24 hours, and collected some reports I thought would be relevant – video conferencing, voice, delay, link usage, network health, etc. The simulation was then run for all three scenarios.
An Analysis of Flow-Based Routing

[Image: A screenshot of a software configuration interface for flow-based routing simulation.]

Common:
- Duration: 24 hours
- Seed: 123
- Value per statistic: 100
- Update interval: 500000 events
- Simulation Kernel: Based on 'KernelType' preference
- Simulation set name: scenario
- Comments: 

Jennifer Casella
An Analysis of Flow-Based Routing

[Image of software interface showing configuration settings for flow-based routing simulation, including duration and values per statistic.]
Roadblocks

As with any project, I certainly ran into my fair share of issues in setting up the simulation for flow-based routing. At first, I didn’t have a compiler installed on my Windows VM – all simulation attempts were failing with compilation errors. Luckily, OPNET has a great support and documentation structure on their website, so I was able to download and install a supported compiler (Microsoft Visual Studio
2005) in order to move forward with testing. Unfortunately, even after the installation of the compiler, my simulation runs were still failing to compile.

I needed to go through and ensure that the appropriate PATH variables were added and set for OPNET to include the Microsoft Visual Studio compiler. This took several hours to get right as different support posts referenced different versions of Windows/different compilers. After setting the path variables appropriate, OPNET was successfully able to compile the model files.

OPNET has an endless number of configurable options for devices/traffic flows, and for the scope of this project, I was limited to selecting and analyzing only those I thought would show the greatest benefits of flow-based routing.
OPNET Modeler 16.0 Simulation Analysis/Results

The simulation results can be viewed by selecting the ‘View Results’ icon in the Modeler toolbar. The results window allows you to view results for one particular scenario, or it allows you to compare all scenarios that are part of the same project – in this case, I selected to view the results data for all scenarios. Comparing Scenario 1, 2 and 3, there are a couple of different characteristics common to the device that can be compared to analyze the performance of the three network environments.
CPU Utilization

One of the statistics that can be compared between devices is CPU utilization. As you can see from the graph below, the traditional IP routing and the traditional IP routing with QoS environments show greater CPU utilization than that of the flow-based routing environment for the nodes; granted the CPU utilization only goes up to 0.000014% at its peak – the utilization for the flow-based router only peaks at approximately 0.000001%. This is an approximate decrease in CPU utilization of 0.000013%. Although this is not a drastically significant increase, it is still an improvement, nonetheless. The decrease in CPU utilization in the flow-based scenario can be attributed to the fact that flow-based routers perform less per-packet inspections than traditional IP routers. Looking up only the first packet of each flow enables the router to switch subsequent packets, and alleviates excessive router overhead.
An Analysis of Flow-Based Routing

We can see the same behavior in different router nodes in the topology – the CPU utilization of the flow-based router is less than that of the traditional IP routers:
An Analysis of Flow-Based Routing

Jennifer Casella
In a few instances, we can see some cases where the CPU utilization of the flow-based router is higher than that of the traditional IP routers.
You can see that the flow-based router CPU peaks at about 0.037%, while the traditional IP routers’ CPU is at a constant rate of 0%. In these cases, traffic may have been taking a different path through the network to reach the destination, as it doesn’t appear these specific IP routers were processing any packets; at least not enough to reflect performance data.

**Delay**

Delay is the average end-to-end delay of data packets in all flows. One can see in the following graph, that the overall delay for the flow-based routing scenario was not an expected result, as it is a bit higher than that of the traditional IP routing scenarios.
The average result for flow-based routing was 0.000022 seconds, as compared with about 0.000018 seconds in traditional IP routing. Certain applications do not tolerant any drops or delay very well, so although this value of delay is not significant, it is still present and could potentially cause major problems for a particular service. The delay observed in the scenarios above could have been attributed to packet processing inside the router. Although it may not be possible to eliminate all traces of delay in flow-based routing, I would have expected that there be less delay in the flow-based routing scenario as compared to the others.

**Traffic Dropped (packets/sec)**

Traffic dropped represents the number of packets that are dropped by all routing devices in the scenario, per second. The following graph represents a view of the overlaid statistics of all three scenarios, and as one can see there is only a slight
difference between any of them. The peak traffic dropped is approximately 0.12 packets per second, but this only occurs for the first couple of seconds.

Other results data collected from the scenarios indicates that the network topology is not fully converged until about 4 seconds into the simulation – the initial traffic drop is likely caused by routing not being fully set up in the environment. After all devices in the network converge no packet loss is seen in any of the scenarios, which is the preferred behavior when deploying bandwidth/delay-intensive services, especially in a flow-based routing environment.

**Background Traffic Delay**

The background traffic delay is a measure of the end-to-end delay experienced by information about a background traffic flow. In our environment, we can see that the delay for traditional IP routing traffic peaks at approximately 0.51 seconds. For
the flow-based routing scenario, the delay is only about 0.12 seconds. This delay is only seen for the first couple of seconds during the simulation. The flow-based routing scenario experienced less background traffic delay. The traffic dropped during the initial network convergence can be one factor that explains why background traffic delay was observed. High CPU utilization or buffer overflow on network devices could also introduce dropped packets, indirectly introducing delay.

Processing Delay

The processing delay metric measures the amount of delay that is experienced by a packet as it travels through the IP layer of the routing device. For most all of the devices across all three scenarios, this delay appears to be the same, and pretty insignificant at 0.000000075 seconds.
Other nodes in the topology reflect that the processing delay for the devices across scenarios is consistent with each other:
Processing delay is likely a true representation of the delay that occurs on each packet that passes through a router. Although it would be phenomenal to simulate (or create in physical devices) an environment that does not experience any delay at all, router overhead still needs to be taken into consideration to account for the packet being passed between the different internal components of the devices.

**IP Processor: Forwarding Memory Free Size**

The forwarding memory free size of the device represents the amount of free bytes available in the router’s memory. Across the traditional IP routing scenarios, the forwarding memory free size is utilized significantly at first, but then it steadies out. As with the previous statistics collected, this forwarding memory utilization could be attributed to routing convergence.
Unfortunately, the forwarding memory free size parameters were not able to be collected for the routers in the flow-based routing scenario, so this statistic cannot be used to compare the traditional IP and flow-based routing environments.

**Survivability Analysis**

OPNET’s Flow Analysis module also has a Survivability Analysis feature built in. In this test, OPNET initially gathers data for a baseline flow run, and then fails links/objects throughout the network, while running the same flow analysis for each failure introduced. The results of the Survivability Analysis can be seen below for each scenario.

Both traditional IP routing environments achieved a score of 20%.
An Analysis of Flow-Based Routing

Jennifer Casella

The flow-based routing environment achieved a survivability score of 50%.

The results of this test indicate that the flow-based routing environment is more resilient, and that flows have a higher survivability rate in the flow-based environment as compared to that of the traditional IP routing environment. Obviously both environments are going to experience irrecoverable failures, especially with the single links between the end nodes, but there were a higher percentage of flows affected in the traditional IP routing environment as compared to that of the flow-based environment.

The higher level of survivability for the flow-based routing environment is attributed to its ability keep utilization of links and interfaces high, without affecting
the path of flows. Furthermore, the ability of flow-based routing to calculate alternate paths for each flow improves its resiliency in the event of a link/hardware failure of routers in the path. In this manner, when a failure is simulated in the network, the devices are able to re-route the traffic and recover more quickly.
Conclusion

Based on the current and undergoing research available today on flow-based routing, it appears to offer a level of service beyond that of the current paradigm to support the increasing needs of network applications and traffic. Storing state information about flows that pass through routers will ensure that the devices can offer the appropriate level of QoS, and faster routing with the elimination of per-packet inspection.

A simulation was constructed in OPNET Modeler, which consisted of three different scenarios – traditional IP routing, traditional IP routing with QoS, and flow-based routing using Juniper ERX routers. Based on the simulations results, flow-based routing proved to have an edge over traditional IP routing in some metrics that were compared, notably CPU utilization. It was expected that the flow-based routing scenario would have proved to be more robust in the simulation, but in this particular environment that was not the case. More intensive traffic flows, different topologies, and additional device configurations could all be tested in a virtual environment similar to the one constructed to further signify the benefits of flow-based routing.
Future Direction

Because OPNET’s library of routers does not support a wide array of flow-based routers, I would like to make use of Modeler’s node model and build my own flow-based router. This would require further research on the algorithmic backend for the routing process. OPNET is a great tool for simulating different network topologies/environment characteristics, but it does have some limitations around fully evaluating flow-based routing. OPNET Modeler has no way measure the power consumption of routing devices. A physical prototype would need to be obtained in order to measure this, and other physical characteristics of flow-based routers.

In the future, I would like to pursue further simulation testing with other tools, namely The Network Simulator (ns02). Like OPNET, ns-2 also provides a simulation environment for testing various routing protocols and traffic flows in wired and wireless networks. ns2 works on the basis of C++ and/or OTcl scripts and commands being uploaded to the application for the purpose of building out network testing scenarios. Without a web interface ns2 may present challenges for users that are not comfortable with using the command line [22].

Another tool that could be used to simulate a flow-based routing environment is OMNeT++. OMNeT++, like n2, is based on C++ code. The tool provides both a graphical and a command line tool for creating simulation environments, as well as logs, graphs, and charts for analysis of the network. In this manner, OMNeT++ is
very similar to OPNET in terms of the extensive web interface. The following screenshot shows an OMNeT++ simulation environment:

Both ns2 and OMNeT++ are available for free download as they are open source applications.

Anagran has developed and produces their own flow-based router: the FR-1000. Although it would probably be near impossible to get hold of some of these devices, it would be a great addition to my research to test out a flow-based router path in a physical environment. There are several packet generator tools available that could simulate voice or video calls going through the network, and comparisons can be drawn between the FR-1000 and other popular vendor routers.
References


Multimedia Data. [Online] Available: 


