QUICK GUIDE TO THE IBM 8371 MULTILAYER ETHERNET SWITCH

Cedell Alexander

Multiprotocol Switching Development
Networking Division
International Business Machines Corporation
Research Triangle Park, N.C.

May 18, 1999

ABSTRACT

The benefits of LAN and ATM switching technology have been well–chronicled: low–cost hardware that supports scalable networks composed of high–bandwidth links with Quality of Service (QoS) guarantees. These characteristics make the 8371 Multilayer Ethernet Switch a key addition to IBM's networking product set. The 8371 combines the advantages of Fast Ethernet LANs with those of ATM backbones in a manner that protects both hardware and software investments. Together, LAN and ATM switching provide the foundation for a high–performance networking infrastructure that supports both current and future needs.

This paper provides a *quick guide* to the functions and associated benefits provided by Release 1.0 of IBM's 8371 Multilayer Ethernet Switch, which is a high–performance layer–2/layer–3 switch that supports both 10/100 Ethernet ports and ATM ports. The 8371 hardware is available in two forms: (1) a standalone switch product, and (2) a product that is installed as a blade module in the IBM 8265 ATM Switch. Both products have 16 fixed 10/100 Ethernet ports. The standalone product also has two slots for either 8–port Ethernet feature cards or a 2–port OC–3 ATM feature card, while the blade module has an integrated OC–12 interface to the 8265 backplane along with one slot for an Ethernet feature card.

As a foundation, the 8371 offers media–speed layer–2 switching between both Ethernet and Emulated LAN interfaces with an aggregate throughput that exceeds 4.7 Mpps. Value is added to this base via support for Virtual LANs (VLANs) along with two forms of high–performance layer–3 switching. To maximize flexibility, the broadcast control and management simplification benefits of VLANs may be realized through three types of policies: (1) Protocol VLANs, (2) IP Multicast VLANs, and (3) User–Defined Sliding Window VLANs. Layer–3 switching is offered both through support for the ATM Forum's MultiProtocol Over ATM (MPOA) Standard, and through a new router acceleration feature called Self–Learning IP. The Self–Learning IP feature automatically offloads IP forwarding duties from adjacent routers in a manner that can produce dramatic performance improvements (such as a 200–fold increase). Other highlights of 8371 Release 1.0 are support for configurationless plug–and–play operation, QoS support for LAN Emulation, hot–swap capability, and extensive support for SNMP manageability via 13 MIBs that include the RMON function.

With this set of functions, the 8371 switch completes the most comprehensive campus–oriented ATM product set in the industry; especially, with regard to the MPOA standard for building scaleable, vendor–independent virtual routers. The combination of IBM's MPOA support for multiple protocols (IP and IPX) and multiple LAN media types (Ethernet and Token–Ring) is unmatched. The 8371's Ethernet MPOA Client (MPC) functions complement the Token–Ring MPC support already provided by IBM's Multiprotocol Switched Services (MSS) Client feature card for the 8270 LAN switch family. When coupled with the MPOA Server capabilities of IBM's MSS Server, the MPOA Client functions of these two IBM LAN switch products provide a complete system solution for IP and IPX routing needs.

ITIRC KEYWORDS

- LAN Switch
- Multilayer Switch
- Fast Ethernet
- Asynchronous Transfer Mode (ATM)
- LAN Emulation (LANE)
- Multiprotocol Over ATM (MPOA)
- Multiprotocol Switched Services (MSS)
- Virtual LAN (VLAN)

ABOUT THE AUTHOR

Cedell Alexander is a Senior Technical Staff Member who lead the design of the 8371 LAN Switch. Previously, he worked on MPOA development for IBM's Multiprotocol Switched Services (MSS) products, was the chief architect for Release 1.1 of the MSS Server, and led the LAN Emulation Service development team for Release 1.0 of the MSS Server. Cedell has 18+ years of industry experience, has authored 50 technical publications along with 17 patent applications, and has made numerous contributions to the ATM Forum's LAN Emulation/MPOA SubWorking Group. He received BS and MS degrees in Electrical Engineering from the University of Houston, and a PhD in Computer Engineering from Mississippi State University. Cedell may be reached by telephone at 919–254–2106, or via the Internet at acedell@us.ibm.com.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

ACKNOWLEDGMENTS

Acknowledgments are due to the entire 8371 development team. It was their hard work and dedication that made the 8371 product a reality. More specifically, thanks to the following members of the 8371 development team:

Charlie Abry
Bob Bass
Brian Bingham
Dawn Blalock
Tom Bonds
Leonard Bush
Rana Dayal
Chris DeBruin
Bill Dickinson
Jim Ervin
Anthony Gallo
Russell Gardo
Ken Haywood
Matt Henry
Jeff Hines
Neil Horman
Alex Jankowsky
Richard Johnson
Florian Kandefer
Arush Kumar
Barry Kesner
Barry Kyu
Loren Larsen
Allen Lau
John Linville
Mrugresh Majmudar
Scott Matheson
Rick Matlack
Andre Minn
Allen Mitchell
Jian Qu
Sanjeev Rampal
Tom Ryle
Ravi Sabhikhi
Chris Siehl
Rich Stelmach
Olen Stokes
Mark Tippens
Raj Vaidhyanathan
Tim Vann
Dee Vig
Jeff Warren
Steve Woolet
Kevin Zyvoloski

1. INTRODUCTION

The goal of this document is to quickly familiarize the reader with the key capabilities provided by Release 1.0 of the IBM 8371 Multilayer Ethernet Switch, which is a high–performance layer–2/layer–3 switch that supports both 10/100 Ethernet ports and ATM ports. More specifically, the networking functions provided in Release 1.0 of the 8371 LAN Switch include:

- *layer–2 switching* (i.e., transparent bridging) between any pair of ports, which may be either Ethernet LAN ports or ATM Emulated Ethernet LAN (ELAN) ports,
- *Protocol Virtual LANs* (PVLANs) for managing the scope of IP, IPX, and NetBIOS broadcast traffic,
- *IP Multicast VLANs*, based on IGMP snooping, to efficiently handle high–volume multicast traffic (IGMP is the Internet Group Management Protocol that is specified in RFC 1112),
- *Sliding Window VLANs* for user–defined broadcast management of other protocols,
- Extensive support for *SNMP manageability* via 13 MIBs, including *RMON–lite*, which is defined to be Groups 1, 2, 3, and 9 of the Remote Monitoring (RMON) MIB specified in RFC 1757 (i.e., the statistics, history, alarm, and event groups),
- *MPOA Client* support for *layer–3 switching* of both *IP and IPX* protocols, which enables users to realize key scaleability and manageability benefits, such as the capability to incrementally increase routing performance in a vendor–independent manner, and
- *Self–Learning IP* functions for automatically offloading IP forwarding duties from adjacent routers in all–Ethernet environments.

With the above set of functions, the 8371 switch completes the most comprehensive campus–oriented ATM product set in the industry; especially, with regard to the ATM Forum's MultiProtocol Over ATM (MPOA) standard for building scaleable, vendor–independent virtual routers. The combination of IBM's MPOA support for multiple protocols (IP and IPX) and multiple LAN media types (Ethernet and Token–Ring) is unmatched. The 8371's Ethernet MPOA Client (MPC) functions complement the Token–Ring MPC support already provided by IBM's Multiprotocol Switched Services (MSS) Client feature card for the 8270 LAN switch family. When coupled with the MPOA Server capabilities of IBM's MSS Server, the MPOA Client functions of these two IBM LAN switch products provide a complete system solution for IP and IPX routing needs.

1. 1. Hardware Overview

The 8371 switch is available in two forms:

1) a standalone product, and
2) a product that is installed as a blade module in the IBM 8265 ATM Switch.

The blade hardware is similar to the standalone hardware. So, the standalone hardware will be described first; then, the differences associated with the blade will be identified.

The base standalone switch unit is equipped with 16 Ethernet LAN ports. The ports on the base standalone switch unit are 10/100 Mbps TX (i.e., copper) ports. Each of the TX ports can operate at either 10 Mbps or 100 Mbps. The port speed can be explicitly configured or auto–negotiated based on the type of device that is connected to the port. Similarly, the duplex mode, full or half duplex, can also be configured or auto–negotiated.

The 10/100 TX ports use RJ–45 connectors. There are two LEDs per TX port: a Link Status LED, and a Speed Indication LED. The Link Status LED, which is on the left when the product is viewed from the front, is *on* when there is a link connection, and will blink *off* for 80–100 milliseconds when there is either transmit or receive activity. The Speed Indication LED is *on* for a 100 Mbps connection and *off* for a 10 Mbps connection.

The fiber ports use MT–RJ connectors. There is one Link Status LED per fiber port. This LED is *on* when there is a link connection, and will blink *off* for 80–100 milliseconds when there is either transmit or receive activity.

The base unit has two slots for feature card installation. Three types of feature cards are available for installation in these slots in Release 1.0:

1) a feature card with eight 10/100 Mbps TX Ethernet ports,
2) a feature card with eight 100 Mbps FX (i.e., fiber) Fast Ethernet ports, and
3) a feature card with two OC–3 (155 Mbps) ATM ports.

The ATM feature card uses SC connectors and multimode fiber. There is one Link Status LED per OC–3 port. This LED is *on* when there is a link connection and *off* when there is no link connection.

In addition to the port LEDs, each of the above feature cards has two Card Status LEDs: OK and Fail. Similarly, the base unit has three status LEDs: Power, OK, and Fail.

Any of the feature cards can be installed in either slot. Either of the Ethernet feature cards can be installed in one or both slots. Only one ATM feature card is supported. Thus, the 8371 supports up to 32 Fast Ethernet ports without ATM, or up to 24 Fast Ethernet ports with 2 OC–3 ATM ports.

The 8371 switch fabric is capable of full–duplex operation at 5 Gbps (i.e., 5 Gbps in and 5 Gbps out for a total throughput of 10 Gbps), which enables non–blocking operation in all configurations. The switch hardware also provides sophisticated capabilities such as per–flow queueing and scheduling options that include rate, priority, and weighted round robin algorithms.

A standard RS–232 serial port is provided on the base unit for local or remote management of the switch. The RS–232 port is brought out to a 9–pin male D–shell connector located on the front of the box.

The base switch unit is equipped with sufficient flash memory to store two operational code images and two configuration files.

The 8371 blade is available in two versions: one with 10/100 Mbps TX ports, and one with 100 Mbps FX ports. The blade also differs from the standalone unit in that only one slot is available for feature card installation. Only Ethernet feature cards may be installed in this slot. The other slot is used for an OC–12 (622 Mbps) connection to the 8265 backplane. Thus, the blade supports up to 24 Fast Ethernet ports, along with a single OC–12 ATM connection (that is always included).

1. 2. Software Overview

The 8371 software is based on the Nways Common Code, which has been shipped in a number of IBM networking products including the MSS Server, the MSS Client, the 2216, the 2212, and the 2210. The amount of new networking functions developed for the 8371 is relatively small. New functions that were developed for initial deployment on the 8371 include:

- Self–Learning IP,
- MPOA Client support for dual ATM adapters, and
- RMON–lite.

This is advantageous from a number of perspectives, such as high reliability due to reuse of proven code and provision of a user interface that many customers are already familiar with. Functions ported to the 8371platform include:

- ATM signalling and ILMI,
- LAN Emulation Client (LEC) support for Ethernet Emulated LANs (ELANs),
- Transparent bridging and spanning tree protocol,

- Protocol Virtual LANs for IP, IPX, and NetBIOS,
- IP Multicast VLANs,
- User–Defined Sliding Window VLANs,
- MPOA Client support for IP and IPX,
- IP Host support for functions such as telnet and ping,
- Command line and web browser interfaces for configuration and monitoring via local/remote in–band/out–of–band connections,
- SNMP support for an extensive set of MIBs, and
- Box services support for functions such as code image and configuration management.

1. 3. Document Overview

This document attempts to summarize key 8371 features in a manner that promotes an overall understanding of the product and the benefits that it offers. The remainder of the document is organized as follows:

- Section 2. is devoted to the layer–2 functions of bridging and VLANs,
- Section 3. focuses on the MPOA Client capabilities,
- Section 4. describes Self–Learning IP,
- Section 5. deals with configuration and console interfaces,
- Section 6. summarizes the network management and RMON capabilities,
- Section 7. covers miscellaneous functions that fall outside the above set of categories, and
- Section 8. provides performance for the switch.

1. 4. Product Highlights

We'll conclude the introduction with a list of product highlights.

- standalone switch product with16 Fast Ethernet ports and 2 expansion slots for feature cards,
- feature cards include:
  - eight–port 10/100 Mbps TX Ethernet feature card,
  - eight–port 100 Mbps FX Fast Ethernet feature card,
  - two–port OC–3 (155 Mbps) ATM feature card
- expandable up to 32 Fast Ethernet ports, or 24 Fast Ethernet ports with 2 OC–3 ATM uplinks,
- dual ATM uplinks for enhanced reliability and additional backbone throughput,
- integrated LAN switch module for the IBM 8265 ATM Switch that includes:
  - sixteen 10/100 Mbps TX Ethernet ports or sixteen 100 Mbps FX Fast Ethernet ports,
  - OC–12 ATM interface to 8265 backplane, and
  - one expansion slot for an Ethernet feature card
- ports automatically negotiate speed and duplex mode,
- 5 Gbps of full–duplex switch fabric bandwidth (10 Gbps throughput),
- per–flow queuing and rate/priority/weighted round robin scheduling algorithms,
- support for plug–and–play configurationless operation,
- support for ATM Forum LAN Emulation Client,
- support for transparent bridging and IEEE spanning tree protocol,
- media–speed layer–2 switching,
- Protocol Virtual LANs for managing the scope of IP, IPX, and NetBIOS broadcasts,
- IP Multicast VLANs with media–speed forwarding,
- Sliding Window VLANs for user–defined broadcast management of other protocols,
- RMON–lite (Groups 1, 2, 3, and 9 of the RMON MIB),
- support for ATM Forum MPOA Client,
- MPOA support for routing both IP and IPX, including support for network routes,
- MPOA support for shortcuts with other MPOA Clients (Ethernet or Token–Ring), NHRP Clients, Classical IP Clients, and LANE Clients (Ethernet or Token–Ring),

- MPOA load balancing and redundancy support for dual ATM uplinks,
- MPOA support for direct shortcuts between stations residing on local LAN ports, which significantly increases cumulative layer–3 forwarding capacity,
- Completes most comprehensive MPOA solution in industry when coupled with the MSS Server and Token–Ring MSS Client,
- Self–Learning IP function for automatically offloading IP forwarding from adjacent routers,
- media–speed layer–3 switching,
- forwarding capability in excess of 4.7 Mpps,
- SNMP support for 13 MIBs (including MIB II, Ethernet MIB, AToM MIB, Bridge MIB, LAN Emulation Client MIB, and MPOA Client MIB),
- IP host support for telnet and ping,
- command line and web browser configuration/console interfaces, and
- support for multiple code image and configuration file banks, along with non–destructive upgrade procedures.

2. LAYER–2 SWITCHING

This section is devoted to the layer–2 switching functions of bridging and VLANs. We'll start with a summary of the basic bridging techniques. Then, the focus will shift to describing the 8371's VLAN functions.

2. 1. Transparent Bridging

The 8371 supports transparent bridging between ports that are Ethernet LANs or Ethernet ELANs. Media–speed performance is achieved by distributing the bridging database to intelligent port controllers as illustrated in Figure 1 below. The three main components of the internal switch architecture are depicted in the figure; more specifically, these components are: (1) a CPU subsystem, (2) the switch fabric, and (3) intelligent port controllers that perform hardware–assisted frame forwarding.

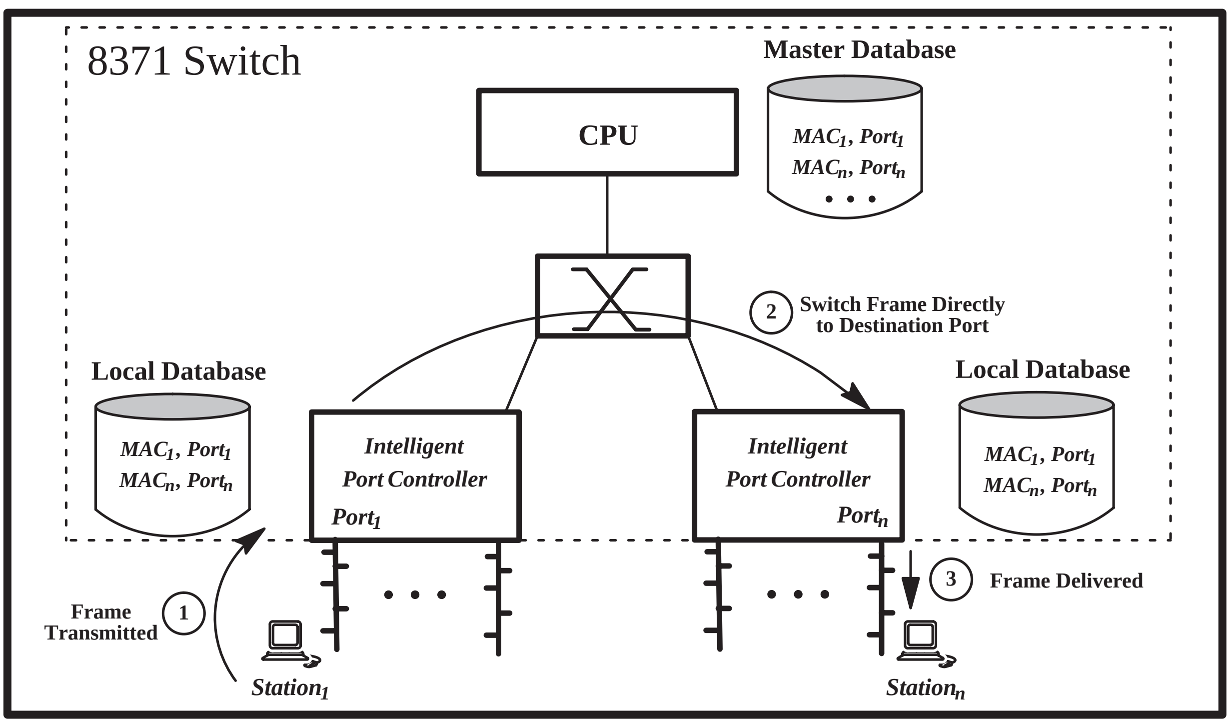

Figure 1. Layer–2 Switching Example

The CPU maintains a *Master Bridging Database* that contains port mappings for all the MAC addresses known to the switch. The Master Database is built in a straightforward manner. When a port controller receives a frame with a previously unknown Source MAC Address (i.e., when the Source MAC Address is not present in port controller's *Local Bridging Database*), the CPU is informed and a new entry for the MAC address is inserted in both the Local and Master Databases.

Each port controller maintains a Local Database that serves as a cache for the Master Database. When a port receives a frame whose Destination MAC Address is in the Local Database, the frame is forwarded directly to the outgoing port via the switch fabric (without involving the CPU). The CPU provides mappings for unknown destinations to individual port controllers on an as–needed basis (i.e., when the port controller needs the mapping to forward frames), which implies that all the Local Databases do not necessarily contain the same set of entries; this approach was taken to make best use of the database memory available on each port controller.

Three types of intelligent port controllers are provided in Release 1.0 of the 8371: (1) Ethernet Port Controllers, (2) OC–3 ATM Port Controllers, and (3) OC–12 ATM Port Controllers (for the 8265 blade module). Each Ethernet port controller is equipped with 512 KBytes of local memory. This memory is used by both layer–2 and layer–3 functions, and is shared by four Ethernet ports. Since each entry in the Local Bridging Database uses 32 bytes of memory, the maximum number of bridging entries supported by one Ethernet port controller is 16K.

Both types of ATM port controllers can support a maximum of 60K entries in the Local Bridging Database.

The size of the Master Bridging Database is configurable, up to a maximum of 256K entries.

As a final comment, note that the transparent bridging support provided by the 8371 is based on a mature component that been enhanced over time to include features often requested by customers, such as the capability to configure permanent entries in the bridging database, and the capability to enable/disable the Spanning Tree Protocol on a port basis (which can significantly decrease convergence time when switching is deployed to the desktop).

2. 2. Virtual LANs (VLANs)

Although *Virtual LANs (VLANs)* and the benefits they render have received considerable attention throughout the networking community, the basic concepts are briefly reviewed below to provide a framework for the description of 8371's VLAN features.

A VLAN is a logical grouping of hosts that form a broadcast domain. The logical grouping is independent of the physical network topology with VLAN membership governed by a set of rules or policies. VLANs have several advantageous characteristics, three of which are listed below.

1) Broadcast domains are controlled. The scope of a broadcast frame originating within a VLAN is limited to LAN (and ELAN) segments containing hosts that are members of the VLAN. Most protocols use broadcasts for address resolution; therefore, if the scope of the broadcast is limited, the scope of host–to–host communication is also limited.

2) IntraVLAN communications is typically very efficient, with VLAN members communicating directly over a switched hardware path.

3) Host moves, adds, and changes are simplified since VLAN membership is independent of physical topology. No reconfiguration is required to retain VLAN memberships when stations move to new physical locations. Similarly, no wiring modifications are needed to move stations from one VLAN to another.

Release 1.0 of the 8371 implements three particular types of VLANs called *Protocol Virtual LANs* (or *PVLANs*), *IP Multicast VLANs*, and *User–Defined Sliding Window VLANs*.

Membership in a PVLAN is determined by network protocol (e.g., NetBIOS) or the combination of protocol and network address (e.g., a particular IP subnet or IPX network). The 8371 creates broadcast domains by dynamically learning the set of PVLANs active on each LAN or ELAN segment. Broadcasts are then limited to the segments containing stations that are members of the PVLAN.

IP Multicast VLANs operate similarly by limiting traffic destined for a multicast group to the segments that contain active members of the group.

Sliding Window VLANs allow VLAN membership to be based on matching a user–defined data pattern/mask at a specified offset in the frame, which is useful in creating VLANs for protocols not explicitly supported by the Protocol Virtual LAN functions, such as AppleTalk.

The net result is that VLANs are important because they enable a switched infrastructure to be partitioned into broadcast and multicast domains in a manner that simplifies network administration with regard to mobility, which is significant with regard to network scalability.

2. 2. 1. Protocol Virtual LANs (PVLANs)

The 8371 utilizes a technique called *Dynamic Protocol Filtering (DPF)* to dynamically partition the bridged network into several Protocol–specific Virtual LANs, or PVLANs. DPF then uses the PVLANs to limit the scope of broadcast/multicast frames that are normally forwarded over all active bridge ports.

DPF monitors the broadcast/multicast traffic received over each bridge port to learn the protocols and subnets being used on that segment. In Release 1.0 of the 8371, the user controls the set of PVLANs to be managed by DPF via configuration (i.e., by simply listing the protocols/subnets to be managed). DPF support is provided for IP, IPX, and NetBIOS. A single NetBIOS PVLAN is supported, while multiple PVLANs may be configured for IP subnets and IPX networks. DPF manages the *forwarding domain* for each configured PVLAN, where the forwarding domain is the subset of bridge ports on which traffic for that PVLAN is being received. Transmission of broadcast/multicast frames for a particular PVLAN is limited to the forwarding domain for that PVLAN. Thus, the scope of broadcast/multicast packets for a particular protocol/subnet is reduced to those segments that are actually utilizing the protocol/subnet. More specifically, IP ARP traffic only travels over LANs/ELANs associated with that IP subnet, IPX broadcast/multicast traffic only travels over LANs/ELANs associated with that IPX network, and NetBIOS broadcast/multicast traffic only travels over LANs/ELANs using NetBIOS.

By controlling traffic on a *per PVLAN* basis, DPF is able to maintain a broad view of the network, which does not require a large amount of switch resources (such as memory). In Release 1.0 of the 8371 switch, DPF is applied to IPX and NetBIOS frames with broadcast or multicast destination MAC addresses, and IP ARPs. DPF runs entirely on the CPU (as opposed to being distributed to the EPIFs); so, if PVLANs are enabled, the above frame types are sent to the CPU for processing (where they would have been flooded to all ports if PVLANs were disabled).

Figure 2 shows an example DPF configuration. In the example environment, multiple protocols/subnets are intermixed across a bridged environment. Assume that *IP Subnet 1*, *IP Subnet 2*, *IPX Network 1*, *IPX Network 2*, and *NetBIOS* are configured for DPF, and that $Port_a$ and $Port_c$ have been identified as members of *IP Subnet 1'*s forwarding domain based on traffic received on those ports. IP ARP traffic for destinations in *IP Subnet 1* is then restricted to LAN_a and $ELAN_c$. Likewise, IPX and NetBIOS traffic is also restricted to a subset of the segments. Assuming that *IP Subnet 3* has not been configured for DPF, any IP ARP traffic for *Subnet 3* would be forwarded over all bridge ports.

The PVLAN capabilities of the 8371 are the same as those provided in IBM's MSS Server and MSS Client products (see for a more–detailed description).

There is essentially no limit on the number of PVLANs that can be supported (i.e., the limiting factor is memory availability on the CPU).

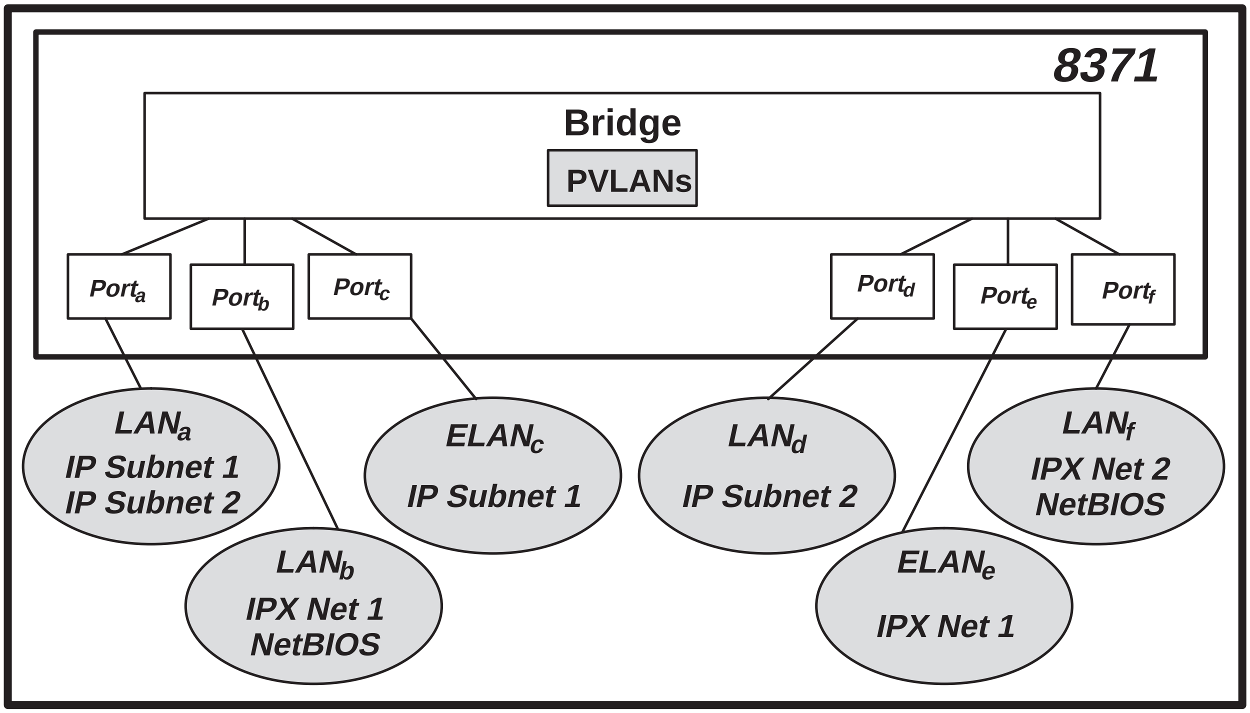

Figure 2. PVLAN Example

2. 2. 2. IP Multicast VLANs

Driven by Internet–related developments, an increasing number of useful applications are emerging that rely on IP multicast protocols to support multiple senders and receivers for purposes such as collaborative computing and multimedia distribution. As these applications grow, so do the demands on the network to provide techniques for efficiently delivering multicast traffic. The traditional approach has been for hosts to advertise their interest in participating in a particular multicast group using the Internet Group Management Protocol (IGMP), and for routers to then form a distribution tree using a multicast routing protocol, such as the Distance Vector Multicast Routing Protocol (DVMRP) or Multicast OSPF (MOSPF).

IP multicast packets are transmitted with layer–2 multicast addresses on LAN media, and if no management techniques are employed, the multicast traffic will be flooded throughout the bridged portion of the network. Our approach to solving this problem is to add IP multicast management intelligence to the 8371switch. The 8371 bridge can participate in IGMP in order to learn which ports are supporting which IP multicast groups, and then only forward the multicast packets for a group to the appropriate subset of ports. IP Multicast VLANs can be automatically created, based on IGMP messages, with no user involvement, and for practical purposes, there is essentially no limit on the number of IP Multicast VLANs that can be supported.

An approach that utilizes the capabilities of the ATM network is illustrated in Figure 3 below. Here, ELANs have been dedicated to particular IP multicast addresses (via configuration), and the 8371 switch directs multicast traffic to the appropriate ELAN (i.e., only multicast traffic for the associated group(s) flows on the dedicated ELANs). Through its IP Multicast VLAN techniques, the 8371 switch can confine multicast traffic received from an ELAN to the LAN segments containing hosts that are currently participating in the multicast group, which makes for an efficient solution.

Note that in the configuration depicted in Figure 3, no special functions are required at the BUS. The BUS simply forwards received traffic on a point–to–multipoint Multicast Forward VCC to all the LECs in the ELAN, which happen to all be supporting members of IP multicast groups; thus, no network bandwidth is wasted and no stations are unnecessarily disturbed. This simplicity of function is an ideal match for the high–performance hardware–assisted forwarding feature of IBM's MSS Server BUS. Furthermore, the Multicast Forward VCCs established by the MSS Server BUSs can be setup with QoS parameters matched to the traffic requirements of the corresponding multicast groups.

Also note that with this LANE–based solution, multicast traffic is logically separated from unicast traffic, and all multicast packet forwarding decisions can be performed at layer–2, even when the destination is in a different subnet.

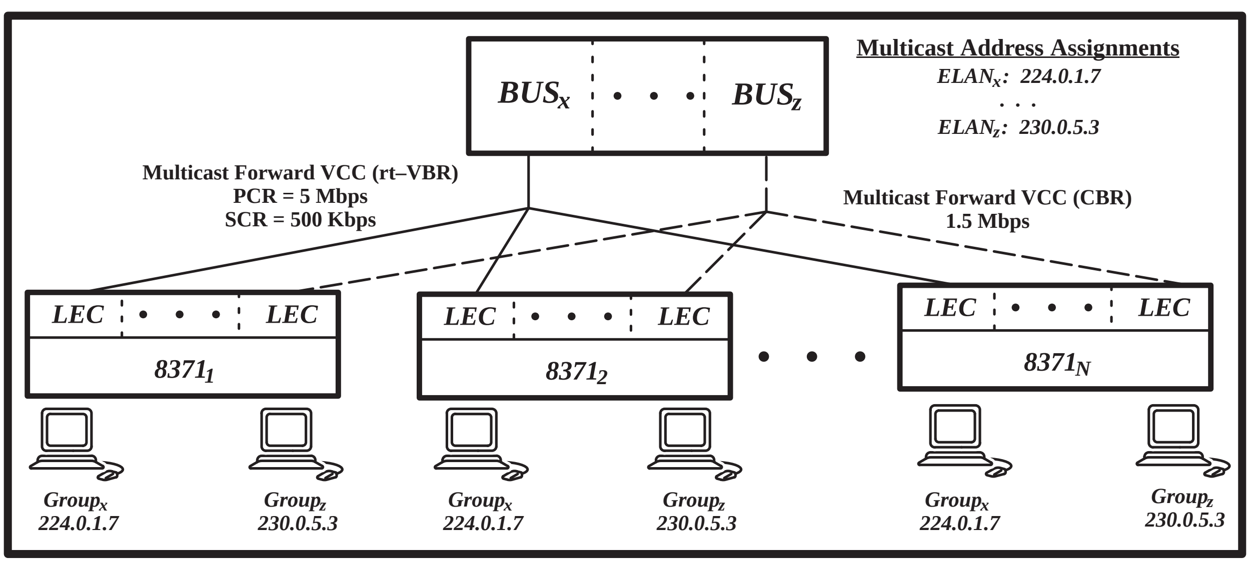

Figure 3. Managing IP Multicasts

2. 2. 3. VLAN Support for DHCP

VLAN support for DHCP has been added in order to provide a means for a network administrator to control which ports DHCP frames are forwarded to. When a DHCP frame is received by the CPU, a check is made to determine whether the receiving port is a member of one or more IP PVLANs. If the receiving port is a member of one or more IP PVLANs, then the DHCP frame is transmitted to the ports that are members of the forwarding domains of these IP PVLANs. If the receiving port is not a member of any IP PVLANs, then the DHCP frame is handled according to the normal bridge forwarding procedures.

2. 2. 4. User–Defined Sliding Window VLANs

The Sliding Window VLAN capabilities of the 8371 are the same as those that were delivered in IBM's MSS Release 2.0.1.

Sliding Window VLANs are the most generic type of VLAN. The user specifies an offset into the frame, a value, and a mask. The CPU then examines received frames at the specified offset looking for a match with the configured {value,mask} pair. Frames that match are included in the VLAN, and only forwarded to the other ports in the forwarding domain of the particular Sliding Window VLAN. One use of this function is to define VLANs for protocols not explicitly supported by the Protocol VLAN functions, with AppleTalk being a prime example. Sliding Window VLANs can also be used as a bridge exclusion filter to drop frames based on specific data patterns.

Sliding Window VLAN filters are only applied to frames received by the CPU. This means that unicast frames may be forwarded without being subjected to any Sliding Window VLAN filters.

3. MPOA CLIENT

The MPOA Client (MPC) function is one of the key features of the 8371 switch. As discussed earlier, the 8371 switch completes the most comprehensive MPOA product set in the industry. The 8371's Ethernet MPOA Client functions complement the Token–Ring MPC support already provided by the MSS Client. When coupled with the MPOA Server capabilities of the MSS Server, IBM's family of MPOA products provide a complete system solution for high–performance IP and IPX routing.

The 8371's MPOA Client functions are a port of the MPC functions originally provided on the MSS Client product. Consequently, much MPOA documentation already exists from work done for prior releases. We won't attempt to replicate all of that information here. Instead, what we'll do in this section is provide an overview of the MPOA model and its benefits, followed by several configuration examples. Next, we'll summarize the basic techniques employed to distribute the MPOA function. Then, the section is concluded with a list highlighting the features of IBM's MPOA Client implementations.

3. 1. MPOA Overview

The *MultiProtocol Over ATM (MPOA) Version 1.0 Specification* standardizes the virtual router model depicted in Figure 4 below. The virtual router is implemented with client–server protocols, where MPOA Clients issue requests to MPOA Servers (MPSs). Internetworking layer route calculation and forwarding functions are partitioned in the model, with MPSs performing the route calculations, MPCs acting as distributed intelligent adapters devoted to high–speed forwarding, and the ATM network providing a scaleable backplane for the virtual router.

The virtual router model provides both scaleability and manageability benefits, with improved manageability being the bottom–line value. The management benefits are based on the scaleability attributes of the model, which allow performance to be increased in an incremental, vendor–independent manner. For example, cumulative forwarding capability can be increased by simply adding more MPCs. Similarly, backplane throughput can be increased with the addition of ATM network capacity.

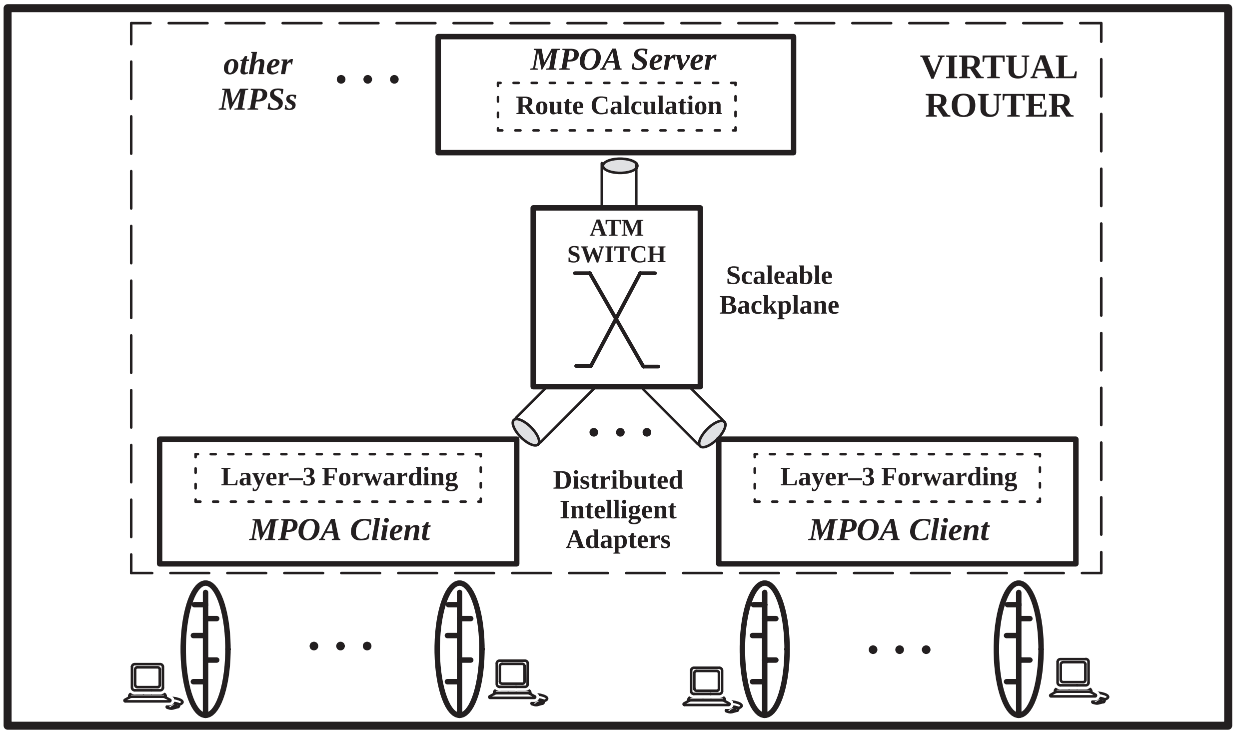

Figure 4. MPOA Virtual Router Model

The virtual router model is contrasted with a more conventional edge router model in Figure 5 below. Given a cost–effective foundation that can scale to meet future needs, it's clearly easier to manage a single virtual router image rather than deal with the complexities of administering multiple distributed edge routers. This is especially true in the case of MPOA, which includes auto–configuration and device discovery protocols that minimize device–specific configuration. Furthermore, the number of devices participating in routing topology protocols is reduced in the virtual router model, which is a scaleability advantage that also reduces edge device complexity.

MPOA is able to provide these benefits in a relatively simple manner by building upon standard networking technologies such as bridging, LAN Emulation, and the Next Hop Resolution Protocol (NHRP). As illustrated in Figure 6 below, there are three basic types of MPOA–enabled devices. MPSs are co–located with router functions and a NHRP Server (NHS), while MPCs reside in *MPOA Hosts* or *MPOA Edge Devices*. All of the devices include a LAN Emulation Client that provides default–path interconnection. MPOA Edge Devices, like the 8371, also include a bridge component for representing stations residing on Ethernet or Token–Ring LAN segments.

The functions performed by a MPC in a MPOA Host are very similar to those performed by a MPC in a MPOA Edge Device, with the main difference being that the MPC in an edge device can represent multiple hosts. The MPC's primary objective is the same in both cases: establishing *shortcut VCCs* and forwarding intersubnet traffic over these VCCs to improve system performance.

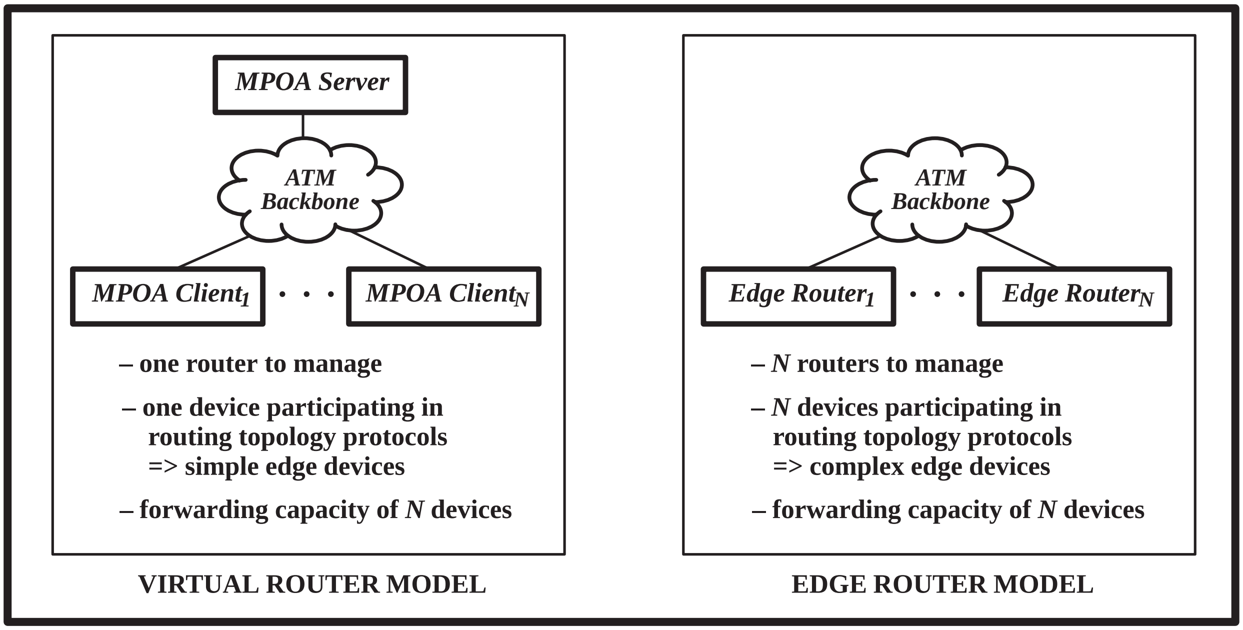

Figure 5. Comparison of Virtual Router and Edge Router Models

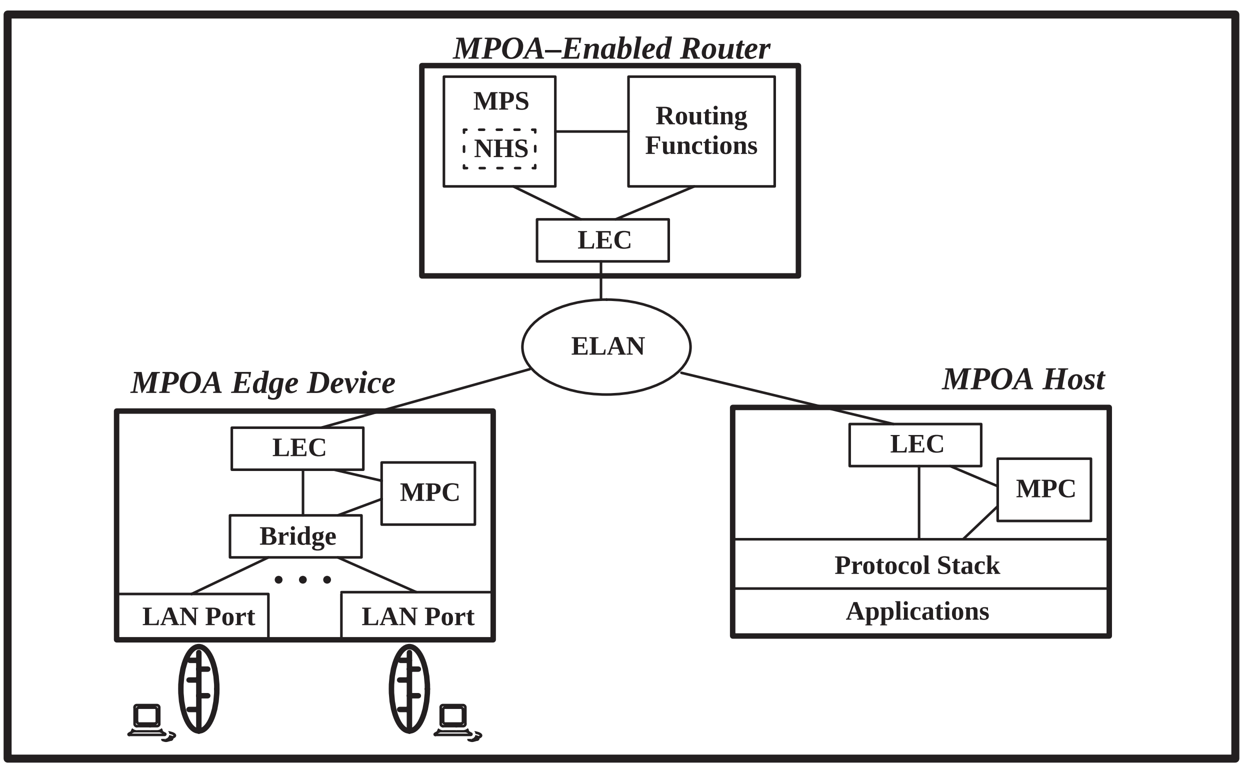

Figure 6. MPOA Components

3. 1. 1. The LANE Connection

MPOA relies on LAN Emulation for three important functions: auto–configuration, dynamic device discovery, and intrasubnet/default–routed–path connectivity. Auto–configuration and dynamic device discovery are features designed to simplify the management of MPOA systems. Auto–configuration allows MPOA configuration parameters to be stored and distributed from a centralized repository, namely the LAN Emulation Configuration Server (LECS). MPOA devices can then obtain their configuration parameters from the LECS while they are being initialized, minimizing configuration at individual devices and helping to create a *plug–and–play* environment.

Dynamic device discovery is the other piece of the plug–and–play story. MPOA devices are not configured with information about neighboring MPOA entities. Instead, MPOA devices dynamically learn about neighboring components through the discovery protocol. The discovery mechanism is quite simple. MPOA devices attach a special TLV (i.e., generic Type–Length–Value information element) to LANE control messages, such as LE_ARP requests and responses. The TLV identifies the device type (e.g., MPS or MPC) and the Control ATM Address for the device. Thus, MPOA devices can learn which MAC addresses are associated with other MPOA devices by simply inspecting received TLVs. While this LAN Emulation enhancement produces an elegant solution, it comes at the cost of requiring LANE components that are compliant with the *LAN Emulation Over ATM Version 2 – LUNI Specification*, which was approved concurrently with the MPOA Specification in July 1997. Fortunately, not all of the devices on the Emulated LANs need to upgraded to LANE v2.0, only the LECs in MPOA devices and the LANE Service components.

The relationships between LANE and MPOA components are very flexible. Some of the possible relationships are illustrated in Figure 7 below. Note that: a single MPC can be associated with one or more LECs; there can be multiple LECs associated with the same MPS; and multiple MPSs can serve clients on the same ELAN.

When there is no shortcut available, intersubnet traffic is bridged to the MPS router over an Emulated LAN. For example, intersubnet traffic transmitted by stations residing behind MPC_1 would be bridged to MPS_1 over $ELAN_a$. Intrasubnet traffic is also bridged in MPOA, and intrasubnet traffic that traverses the ATM network is bridged over an ELAN. Since most MPOA edge devices include LAN switching hardware capabilities, intrasubnet traffic is handled very efficiently with end–to–end switching. This use of bridging, coupled with dynamic device discovery, enables the MPC to be independent of router topology and redundancy protocols, while maintaining the change management benefits provided by Virtual LANs (VLANs) (e.g., allowing a station to be moved from a segment behind one MPC to a segment behind another MPC without any reconfiguration).

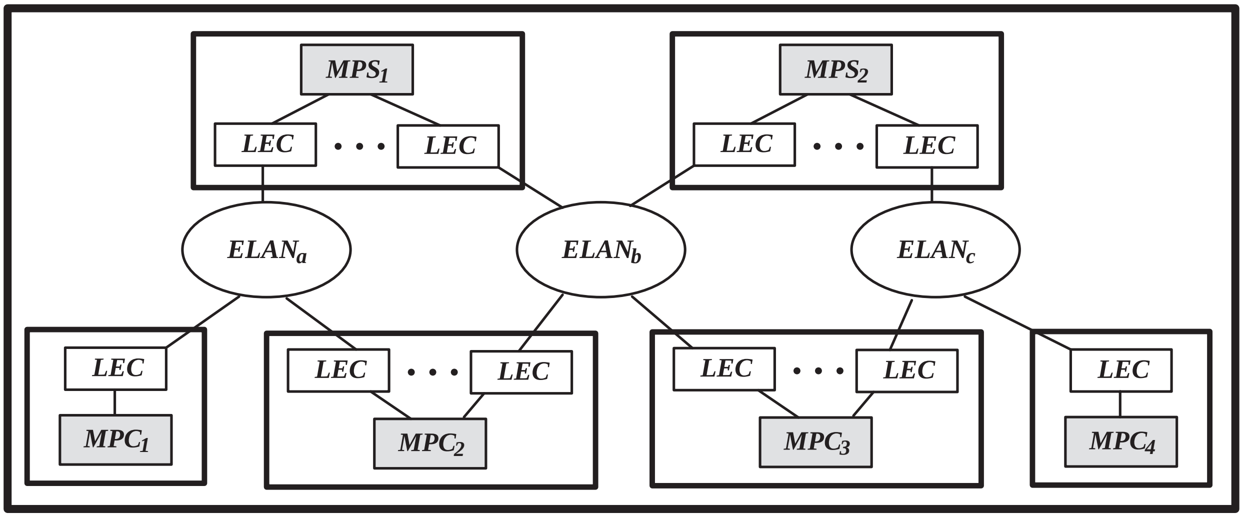

Figure 7. Relationships Between LANE and MPOA Components

3. 1. 2. Establishing Shortcuts

The MPC is responsible for initiating shortcut establishment. The first step towards establishing a shortcut is for the MPC to discover the MAC addresses of the MPS routers along with the corresponding Control ATM Addresses. The MPC then monitors the flow of traffic destined for these MPS MAC addresses. Flow detection is performed independently for each MPS. When the rate of traffic to a particular internetworking–layer destination address exceeds the configured threshold (which defaults to 10 frames/second), the MPC initiates shortcut establishment by sending a MPOA Resolution Request to the associated MPS.

We'll use the example configuration shown in Figure 8 below as a reference point for describing the shortcut establishment procedures. Assume that *IP* $Host_A$ is transmitting to *IP* $Host_B$ and that MPC_1 has determined that the flow exceeds the threshold for establishing a shortcut. MPC_1 then sends a MPOA Resolution Request for $Host_B$*'s* IP address to MPS_1. Upon receiving the request, MPS_1 converts the MPOA Resolution Request to a NHRP Resolution Request, and forwards the converted request along the routed path to MPS_2. MPS_2 will respond to the NHRP Resolution Request because it *serves IP* $Host_B$ (i.e., a MPS serves destinations that reside on subnets local to the MPS).

Before responding to the resolution request, MPS_2 imposes an egress cache entry on MPC_2. MPS_2 imposes the cache entry because it recognizes, through the discovery protocol, that the destination MAC address is associated with a MPC. The cache entry is imposed with a MPOA Cache Imposition Request that contains the Data Link Layer (DLL) header that MPS_2 would use to transmit frames to *IP* $Host_B$. MPC_2 records the cache entry and responds with a MPOA Cache Imposition Reply that includes an ATM address that can be used to set up a shortcut VCC for traffic destined to *IP* $Host_B$.

After receiving the imposition reply, MPS_2 inserts the ATM address into a NHRP Resolution Reply that is sent back to MPS_1. MPS_1 transforms the NHRP Resolution Reply into a MPOA Resolution Reply that is sent to MPC_1. MPC_1 then uses the ATM address to set up a shortcut VCC to MPC_2. Once the shortcut VCC is established, traffic destined for *IP Host*$_B$ is transmitted over the shortcut VCC, bypassing the intermediate routing hops at MPS_1 and MPS_2. When MPC_2 receives the frames over the shortcut VCC, it inserts the DLL header from the corresponding cache entry for IP address 9.1.3.1, and delivers the frames to *IP Host*$_B$ as if they were received via the routed path from MPS_2.

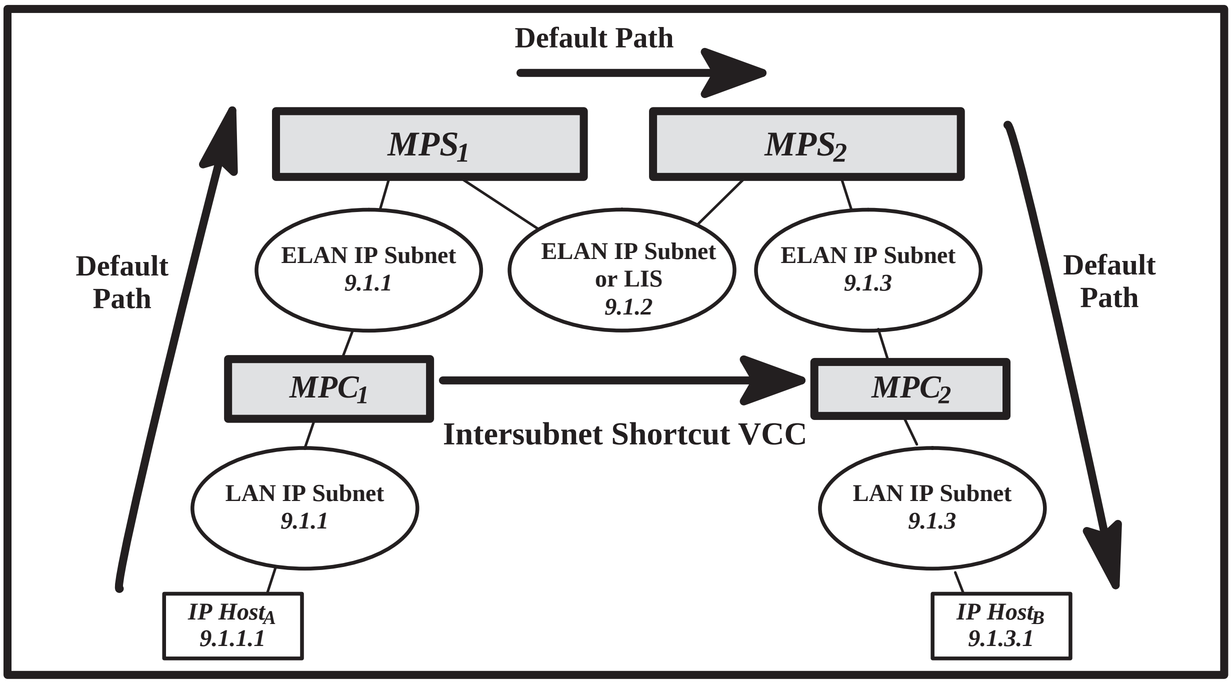

Figure 8. Bypassing Intermediate Router Hops with a MPOA Shortcut

The shortcut establishment flows described in the preceding paragraphs are depicted in Figure 9 below. Steps 1–6 would need to be repeated in the opposite direction for the reverse traffic flow, from *IP Host*$_B$ to *IP Host*$_A$, to be transmitted over the shortcut VCC. In general, the same VCC can be used in both directions, and for other flows between the two MPC devices.

MPOA devices do not exchange control information, such as resolution requests and replies, over ELANs; instead, control information is transmitted over Control VCCs. Each MPC and each MPS has a Control ATM Address that is advertised as part of the discovery protocol, and Control VCCs are established between these Control ATM Addresses on an as–needed basis.

By default, LLC/SNAP encapsulation is used for both control and shortcut data flows. MPOA control frames are encapsulated in the same manner as NHRP control frames, while shortcut data frames use either a routed protocol encapsulation format defined in RFC 1483 or an optional MPOA

Tagged Encapsulation format. The tagged format can be used to improve performance at the egress MPC. Ingress MPCs are required to include a MPOA Cache Tag Extension on MPOA Resolution Requests. This extension is passed to the egress MPC on the Cache Imposition Request, and the egress MPC may either ignore the extension or provide a 32–bit tag value of its choosing. If the egress MPC provides a tag value, the ingress MPC must use the tagged encapsulation format for the data flow. The tagged encapsulation includes the tag value in each data frame. Thus, the egress MPC can use the tag value, which it selected, as a key to efficiently locate the imposed cache entry containing the DLL header for the flow.

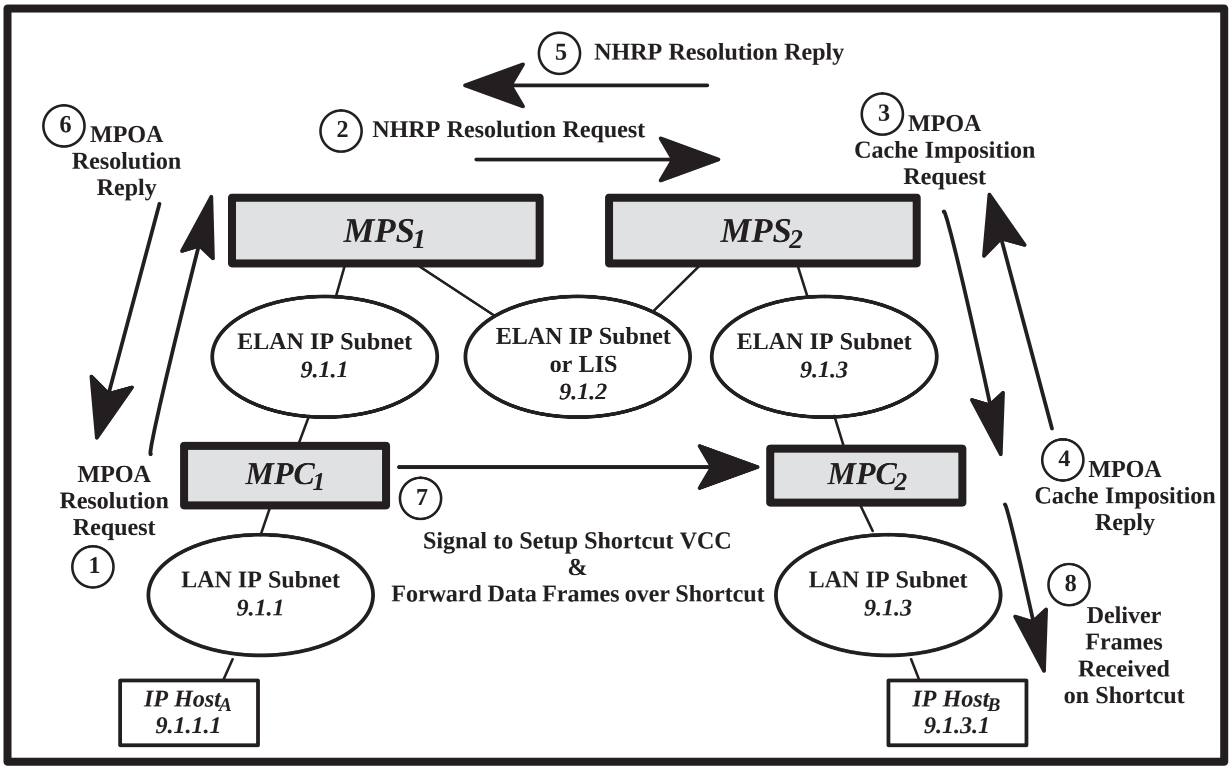

Figure 9. MPOA Shortcut Establishment Flows

3. 2. Example MPOA Configurations

Figure 10 below contains an example of a redundant MPOA configuration. The two MSS Servers provide the routing functions, with *MSS Server #1* acting as the primary MPS and *MSS Server #2* filling the backup role. Each 8371 contains dual OC–3 ATM interfaces. One LEC and one MPC instance is associated with each ATM interface. Since all the LECs join the same ELAN, the spanning tree protocol will block one LEC in each 8371. If the ATM interface associated with the active LEC should fail, normal spanning tree procedures will ensure that the other LEC/ATM interface takes over.

Note that there is a unique MPC instance for each ATM interface in Figure 10. The LECs that are defined on the same ATM interface as the MPC are *associated* with that MPC. By *associated*, we mean that a MPC instance will only establish shortcuts for flows that are being bridged to a MPS over one of its *associated* LECs. When LECs defined on different ATM interfaces are members of the same ELAN, then the active MPC instance for that ELAN is the one associated with the LEC that is in bridging forwarding state. Thus, MPC redundancy is achieved naturally in conjunction with layer–2 redundancy.

Also note that a single MPOA Shortcut VCC is established between the two 8371s in Figure 10. The VCC is used for all MPOA flows, in either direction, between the two 8371 switches. The MPOA Tagged Encapsulation format is used on the VCC. Given the choice, 8371 MPCs will always prefer to utilize the tagged encapsulation format.

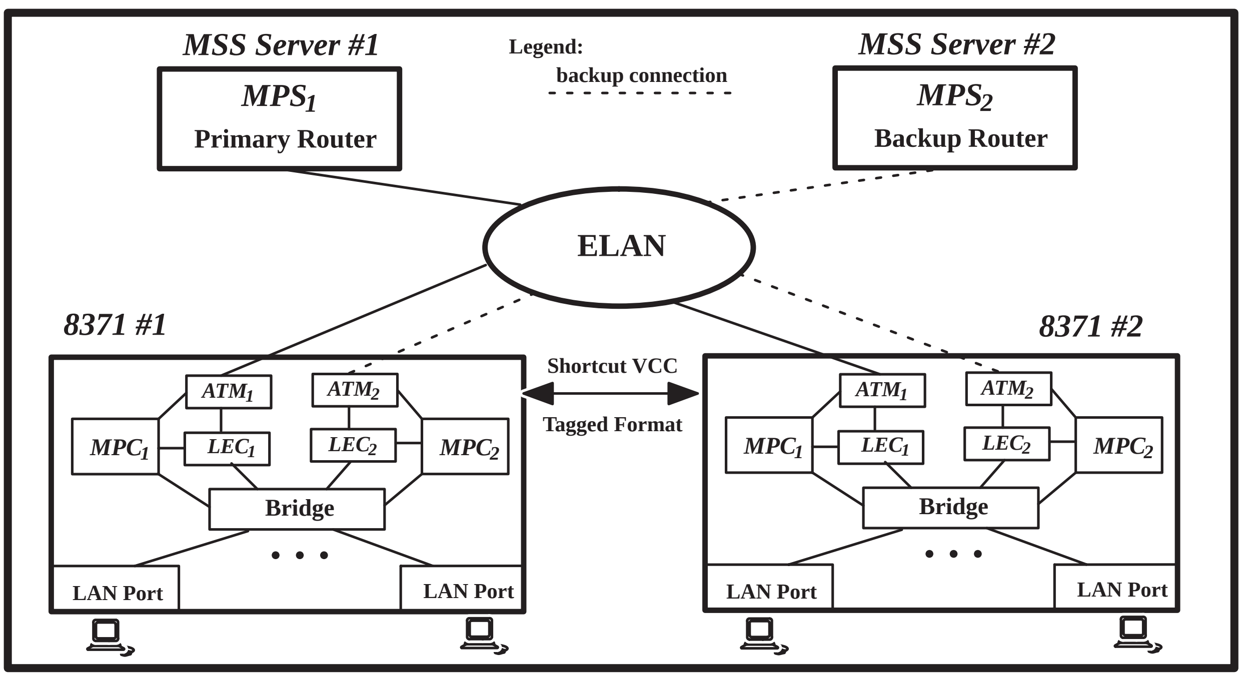

Figure 10. Example of a Redundant MPOA Configuration

Figure 11 below illustrates a configuration that augments the previously–described redundancy features with load–balancing function. In this example, some of the LAN stations downstream of *8371 #1* are members of $Subnet_1$, while others are members of $Subnet_2$. *MSS Server #1* is the default gateway for $Subnet_1$, and *MSS Server #2* is the default gateway for $Subnet_2$. Thus, two LECs are simultaneously active in *8371 #1*. The active LECs are defined on different ATM interfaces, with a LEC on interface ATM_1 providing connectivity to *MSS Server #1*, and a LEC on ATM_2 providing connectivity with *MSS Server #2*. Both MPC instances in the 8371 are also simultaneously active.

MPC_1 shortcuts traffic originating from *$Subnet_1$*, while *MPC_2* shortcuts traffic originating from *$Subnet_2$*. Although not shown in the figure, *8371 #2* is configured similarly, with the active LEC on interface *ATM_1* providing connectivity to an ELAN devoted to *$Subnet_3$* and the active LEC on interface *ATM_2* providing connectivity to the *$Subnet_4$* ELAN. As a result, multiple (up to four) shortcut VCCs may be setup between the 8371 switches.

There are two more items of note in Figure 11. First, notice that the MPOA and VLAN functions are complementary in nature. VLANs restrict the intrasubnet traffic to the appropriate domain, while MPOA shortcuts the intersubnet traffic.

Local shortcut handling, which is a feature of IBM's MPOA Client implementation, is the second item of note. A local shortcut occurs when both the source and destination stations are accessible via LAN ports of a single MPC (or via two MPC instances that reside on the same box) as depicted in Figure 11. In this case, the IBM MPC switches the traffic directly between the LAN ports without traversing the ATM network (in a loopback manner), which significantly increases the cumulative layer–3 forwarding capacity of the 8371 switch.

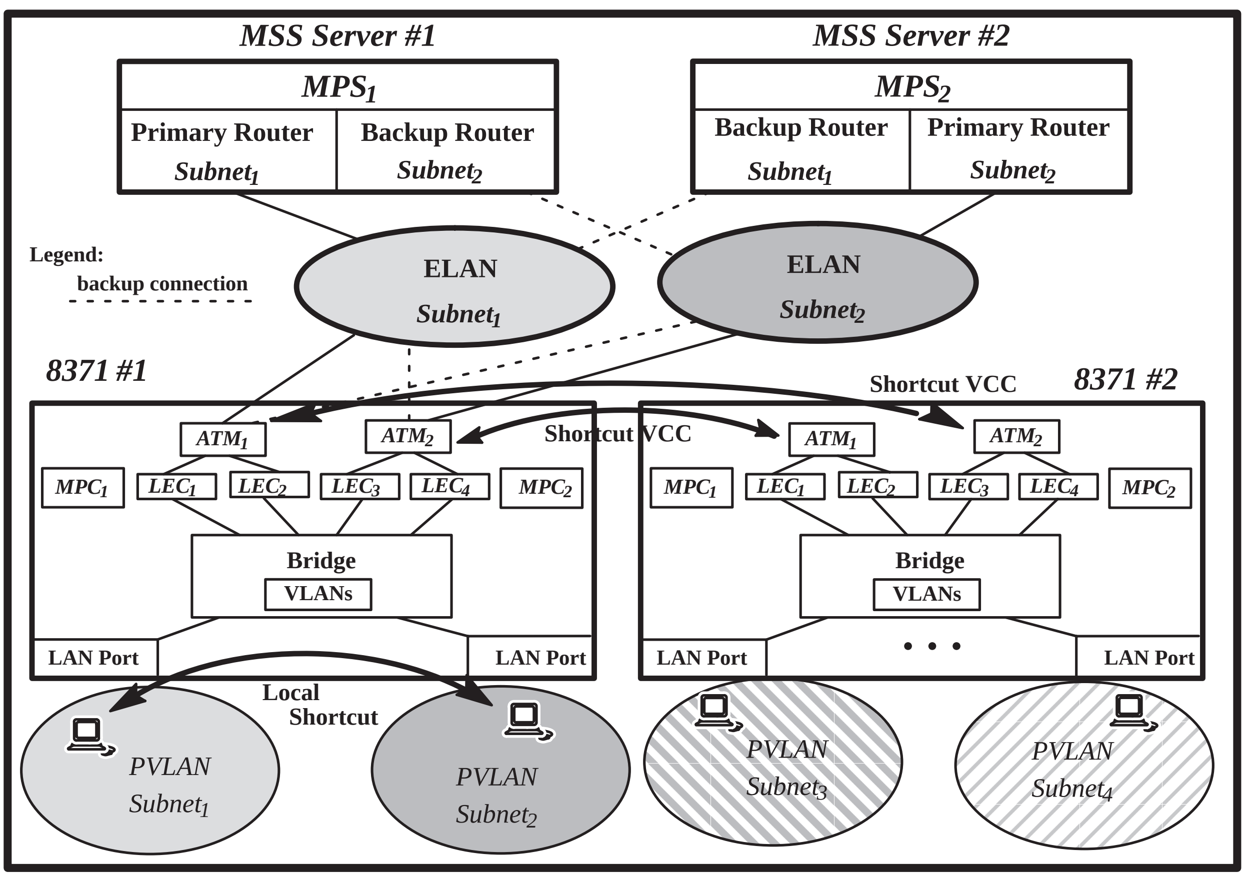

Figure 11. MPOA Load Balancing Example

Although the MPOA and NHRP Specifications do not define support for LANE–encapsulated data packets, they do define a generic mechanism for vendor–private extensions. The mechanism is interoperable because implementations that don't understand the extensions simply ignore them. IBM has leveraged this capability by designing an open set of extensions that support *LANE Shortcuts*. These extensions have already been implemented as part of the NHRP function delivered in Release 1.1 of the MSS Server, and as part of the MPOA function shipped in MSS Release 2.1. The extensions are an important value–add because they extend the reach of MPOA shortcuts to the large installed base of LANE equipment, including LAN switches with ATM uplinks. Consequently, the extensions have been implemented on the 8371. Figure 12 below illustrates MPOA shortcuts from the 8371 MPC to LANE devices on both Ethernet and Token–Ring ELANs.

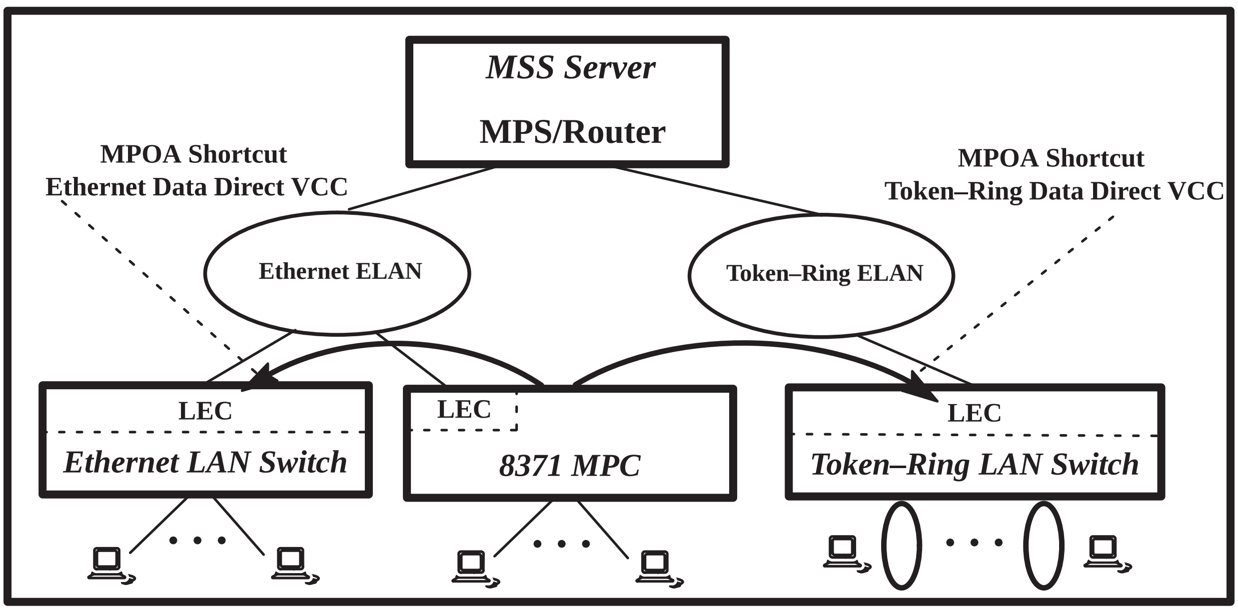

Figure 12. MPOA Shortcuts to LANE Devices

The 8371's LANE–shortcut capability also comes in handy when establishing MPOA shortcuts to Token–Ring MSS Clients. Figure 13 below shows two unidirectional MPOA Shortcut VCCs between a 8371 and a MSS Client. The two VCCs are established to accommodate the individual preferences of the two switch platform implementations. As previously discussed, the 8371 prefers to receive frames that are encapsulated in the MPOA Tagged format, while the MSS Client prefers the Token–Ring LANE encapsulation. The MSS Client prefers the LANE format for performance reasons, since the underlying platform can efficiently switch LANE frames in hardware.

The 8371 can handle reception of shortcut frames that are encapsulated in any of the following formats: (1) MPOA Tagged, (2) Ethernet LANE, or (3) the LLC/SNAP encapsulation for routed protocols defined in RFC 1483. The MPOA Tagged Encapsulation format is used between two 8371

MPCs and between the 8371 and non–IBM MPCs. The Ethernet LANE capability is useful when a NHRP Client (NHC) in a MSS Server shortcuts to the 8371, while the RFC 1483 format is employed on shortcuts with non–IBM NHCs (i.e., pure NHCs that are not MPCs).

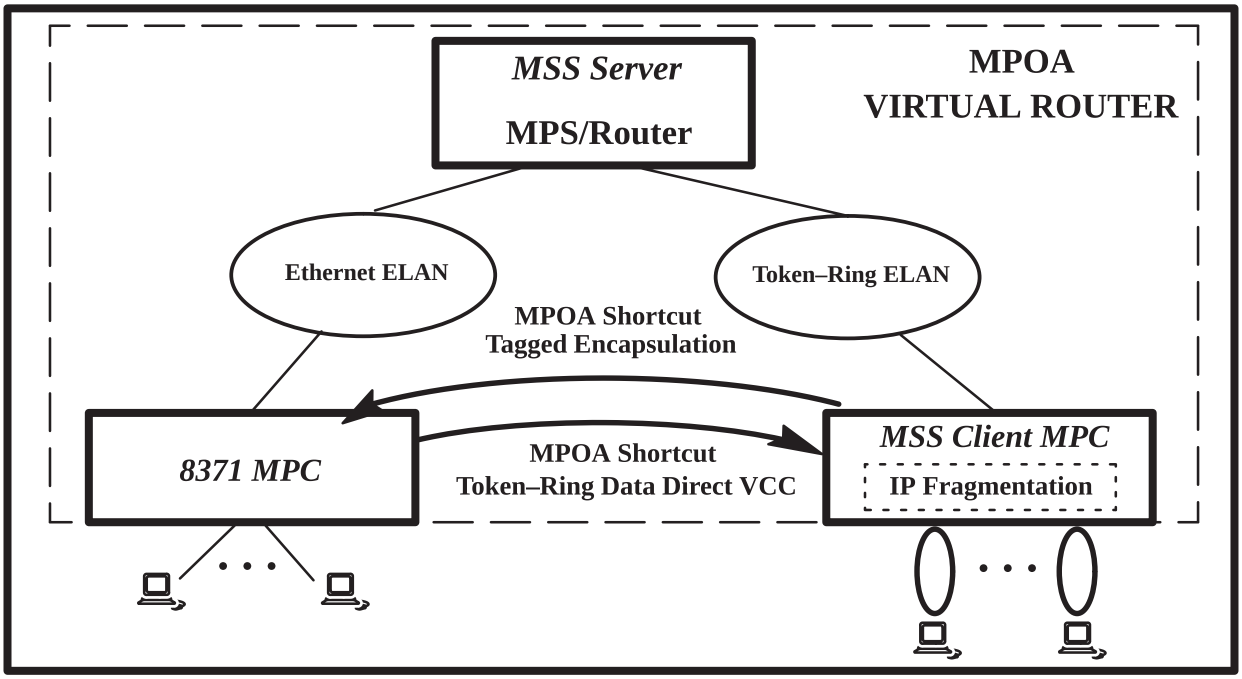

Figure 13. MPOA Shortcuts Between the 8371 and the Token–Ring MSS Client

Before leaving the example provided by Figure 13, note that the MPC in the Token–Ring Client is capable of performing IP fragmentation. This capability enables the MPC to handle MTU mismatches resulting from the fact that Token–Ring LANs support larger frame sizes than Ethernet LANs.

In addition to shortcutting to other MPCs and LANE devices, the 8371 MPC can also shortcut to NHRP Clients and Classical IP (CIP) Clients. Figure 14 provides an example of a shortcut to a CIP Client. In this case, the CIP Client resides on a MSS Server that is acting as the egress router to destinations that are not resident on the ATM network. In the particular example depicted in Figure 14, the destination, *IP Host*$_B$, is connected to the egress router via a FDDI LAN. Since shortcuts to all destinations on *IP Subnet D* will be terminated at the egress router, a *network route* can be returned in the resolution reply. The network route informs the ingress MPC that flows to all destinations on *IP Subnet D* can be sent on the same shortcut VCC, which reduces both resolution overhead throughout the network and storage requirements at the ingress MPC. Network routes for the IPX protocol are also supported in a similar manner.

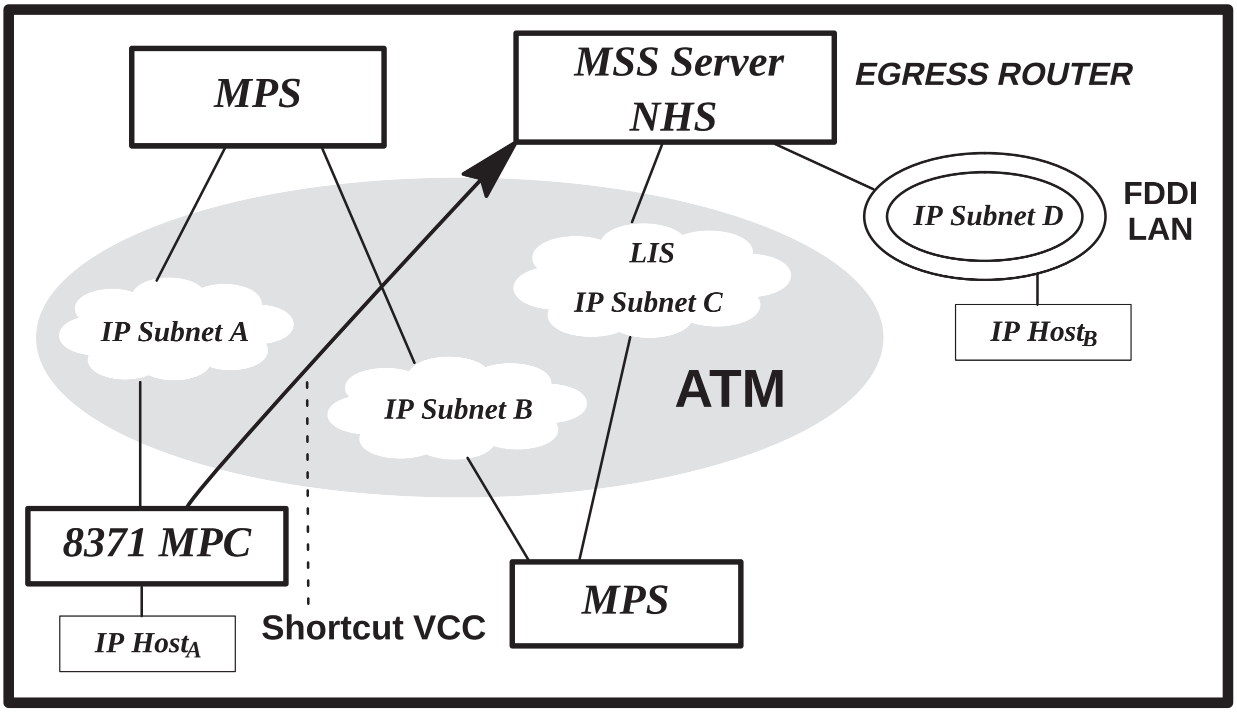

Figure 14. MPOA Shortcut to an Egress Router

3. 3. MPOA Client Distribution Techniques

Some of the fundamental techniques used to distribute the MPOA Client function are described in this subsection. The basic approach is patterned after the layer–2 design that was described in Section 2. By this, we mean that a *Master Database* is maintained by the MPC on the CPU, and *Local Databases* containing MPOA cache entries are kept at each of the intelligent port controllers.

3. 3. 1. MPS Discovery

When the MPC discovers a MPS, the MPC issues a *mpsMacAddrAddRequest* to request that all the Ethernet port controllers add a database entry for the MPS's MAC address. This is done because the Ethernet port controllers only monitor layer–3 flows destined for MAC addresses that have been identified as MPS MAC addresses.

3. 3. 2. Flow Detection and Distribution of Cache Entries

MPOA processing is initiated in an Ethernet port controller when the layer–2 lookup indicates that the destination MAC address is that of a MPS. If there is an entry for the layer–3 destination in the Ethernet port controller's Local Database, the frame is switched onto a MPOA shortcut. Otherwise, the frame is bridged to the MPS and a *flowDetectionIndication* is sent to the CPU as illustrated in Figure 15 below.

When the *flowDetectionIndication* is received by the MPC at the CPU, the MPC consults its Master Database to see if a shortcut already exists to the layer–3 destination. If a shortcut does exist, the MPC issues a *ingressCacheAddRequest* to download a forwarding cache entry as shown in Figure 15. Otherwise, the MPC updates the flow detection count for the layer–3 destination, which can trigger a MPOA Resolution Request to establish a shortcut if the flow detection threshold is reached.

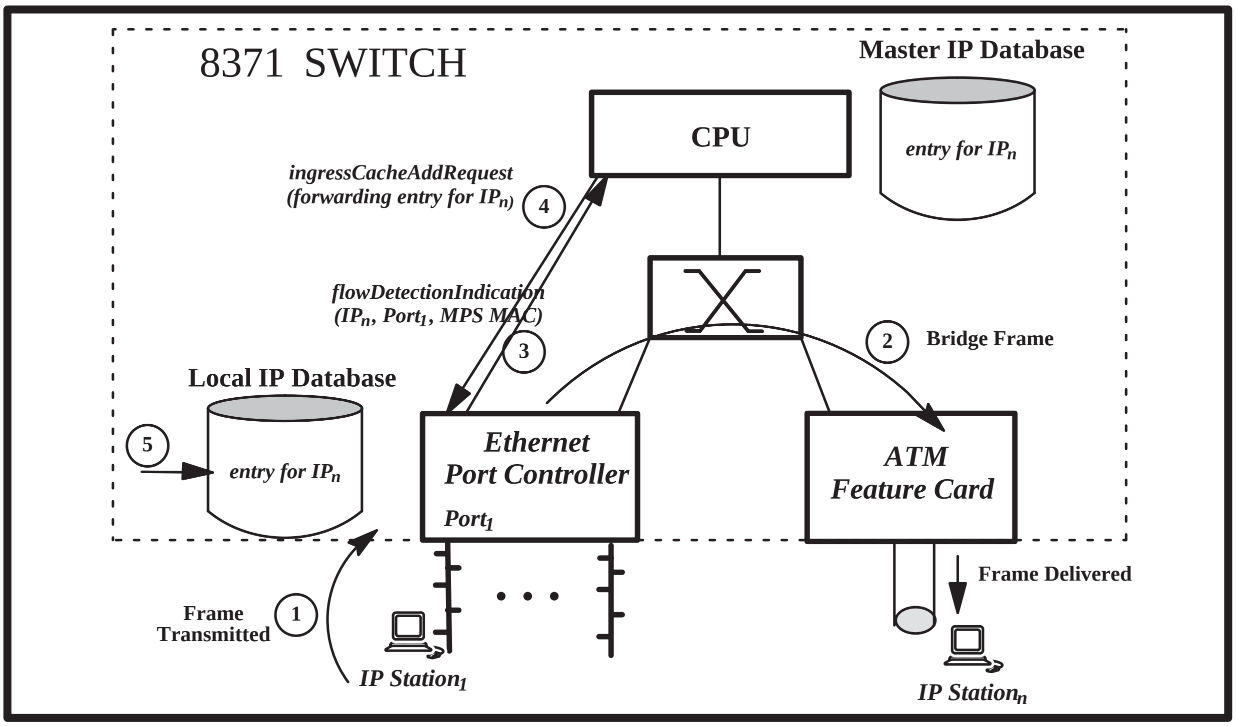

Figure 15. MPOA Flow Detection and Shortcut Distribution in the 8371 Switch

Ingress cache entries are only added to LAN interfaces after receiving a *flowDetectionIndication* as described above. Conversely, egress cache entries are only added to ATM interfaces. The MPC issues an *egressCacheAddRequest* to request that an imposed entry be added to the egress cache of a particular ATM interface. The egress cache entry contains the MAC header to be prepended to packets destined for the specified layer–3 address. In Release 1.0 of the 8371, the *egressCacheAddRequest* primitive supports the MPOA Tagged Encapsulation format. Shortcut frames received on Ethernet LANE VCCs are simply switched by the ATM port controller using the normal layer–2 procedures. Only pure RFC 1483–encapsulated frames need to be forwarded to the CPU for processing.

Each Ethernet port controller can support a maximum of 8K ingress cache entries, while the ATM port controller supports 16K tagged egress cache entries.

3. 3. 3. End–to–End Switching

Once the cache entries have been downloaded to the intelligent port controllers (e.g. ETH_1 and ATM_2), frames can be switched end–to–end as illustrated in Figure 16. Each of the steps identified in the figure is explained below.

Step 1: ETH_1 receives a frame transmitted by *IP* $Station_1$ and looks up the layer–2 destination, which is marked as being the MAC address of a MPS

Step 2: ETH_1 retrieves a forwarding entry for the layer–3 destination, IP_2, from the Local Ingress Cache Database, which maintains its entries on a per–MPS basis; in this case, the entry indicates that the MPOA Tagged Encapsulation is to be used, so ETH_1 replaces the MAC header with a tagged header (the tag value is stored in the forwarding entry) and prepends a small internal header before transmitting the frame into the 8371 switch fabric

Step 3: the frame is switched to ATM_1, which forwards the frame onto the external ATM network with minimal processing overhead

Step 4: the frame is switched through the ATM network to MPC_2

Step 5: the receiving ATM port controller, ATM_2, recognizes that the frame is encapsulated in the tagged format and uses the tag value as a key to retrieve the associated entry from the Local Egress Cache Database, ATM_2 utilizes the contents of the cache entry to build a MAC header for the packet, normal layer–2 bridging procedures are then used to determine the outgoing port associated with the destination MAC address and the frame is transmitted to that port via the 8371 switch fabric

Step 6: the frame is switched to ETH_2, which forwards the frame onto the external Ethernet LAN with minimal processing overhead

Step 7: the frame is delivered to the proper destination, *IP* $Station_2$.

If LANE encapsulation is used (instead of the tagged encapsulation format), the primary difference in the above procedures is that the MAC header is supplied by the ingress MPC in *Step 2* (as opposed to being supplied by the egress MPC in *Step 5*). Local Shortcuts (see Figure 11 and accompanying discussion) handling is similar to that for LANE encapsulation (i.e., the MAC header is supplied by the ingress EPIF), with the exception that the frame never traverses the ATM network.

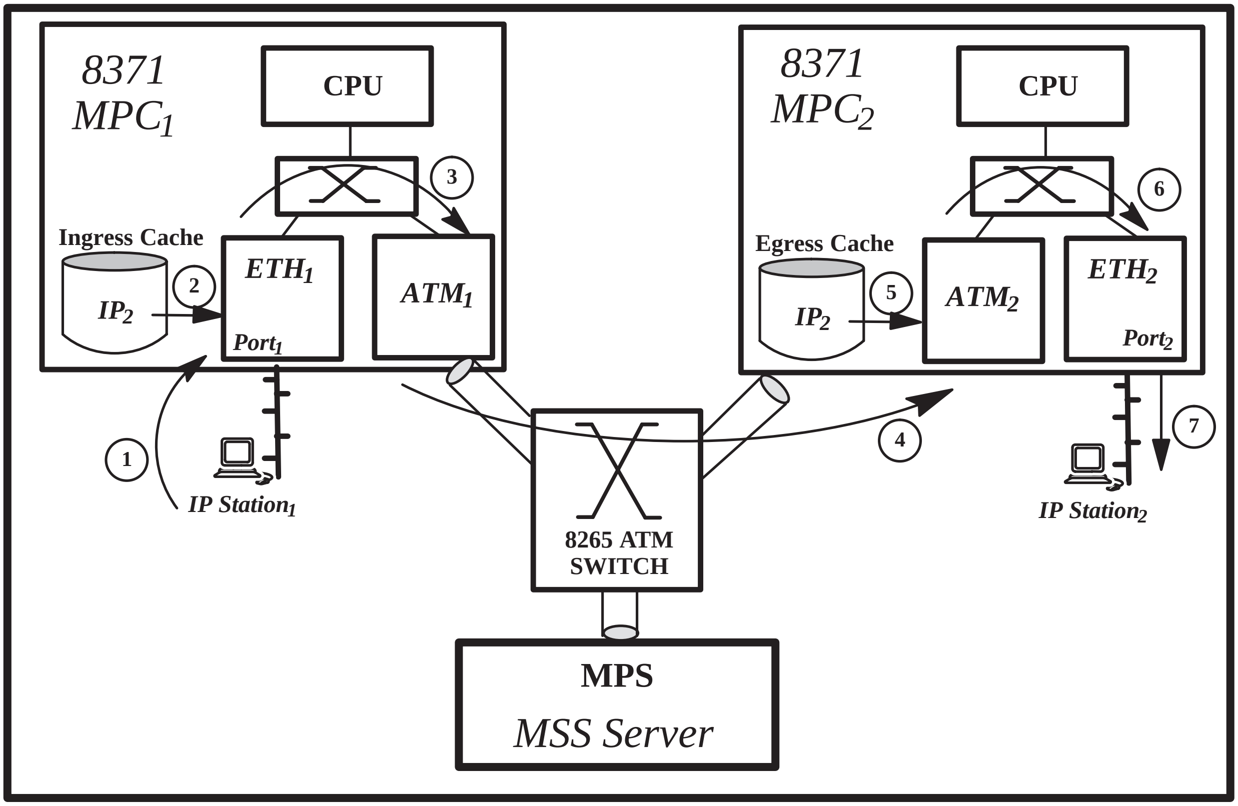

Figure 16. End–to–End Switching with MPOA

3. 4. Highlights of IBM's MPOA Client Implementations

The following bullets highlight features of IBM's MPOA Client implementation.

- Fully compliant with the ATM Forum's MPOA Specification
- Provides manageability and scaleability benefits of virtual router model
- Available on Ethernet (8371) and Token–Ring (MSS Client) LAN platforms
- Support provided for both IP and IPX protocols
- Complete auto–configuration support, with no MPC configuration required and the capability of receiving configuration parameters from the LAN Emulation Configuration Server
- Load balancing and redundancy support for dual ATM uplinks via multiple MPC instances on 8371 platforms
- Support for the optional network route feature, which offers significant scaleability advantages
- Support for performing IP fragmentation on the Token–Ring MSS Client platform

- Extends MPOA shortcut capability to the large installed base of LAN Emulation equipment including devices on both Token–Ring and Ethernet ELANs
- Provides the flexibility of shortcutting to other MPOA Clients, NHRP Clients, Token–Ring LAN Emulation Clients (LECs), Ethernet LECs, and Classical IP Clients when coupled with IBM's MSS Server
- Utilizes high–performance hardware switching capabilities of 8371 platform to achieve media–speed frame forwarding
- ○ Support for local shortcuts, which significantly increases cumulative forwarding capacity
- Utilizes high–performance hardware switching capabilities of MSS Client Token–Ring LAN platform for egress frame delivery
- Support for reserved bandwidth ATM service categories
- Extensive manageability capabilities including several message logging categories, packet tracing, and support for the MPOA Client MIB

4. SELF–LEARNING IP

The Self–Learning IP feature is similar to MPOA in a number of ways. Both are router acceleration techniques that enable scaleable solutions by distributing layer–3 forwarding functions out to LAN switches. Self–Learning IP also relies on many of the same implementation mechanisms as MPOA.

One way that the two features differ is that Self–Learning IP is intended for all–Ethernet environments. Another difference is that MPOA has specified protocols for discovering routers and resolving shortcut routes to layer–3 destinations, while Self–Learning IP improvises to learn this same topology information by *snooping* on the contents of selected packets.

As a result of its snooping activities, Self–Learning IP is able to identify adjacent routers (i.e., routers that are attached to ports of the 8371 switch) and build an IP forwarding table. Each forwarding table entry includes the following information:

- IP address of the station,
- MAC address of the station,
- interface through which the station can be accessed,
- type of LAN encapsulation used by the station (e.g., DIX or LLC/SNAP),
- MAC address of the station's default router, and
- a timeout value indicating when the entry is to be aged–out (e.g., if it's not being used).

With this database, Self–Learning IP is able to identify packets destined for routers and offload the IP forwarding function from these routers. Figure 17 below illustrates one of the configurations where Self–Learning IP is useful. In this configuration, stations are moved from an Ethernet port on the router to an Ethernet port on the 8371 switch, and then the router port is also connected to a port on the 8371 switch. With this simple wiring change, the Self–Learning IP function of the 8371 switch can begin to offload IP forwarding functions from the router. For example, routing traffic between $Station_1$ and $Station_N$, which frees up forwarding capacity that the router can use for WAN traffic. Furthermore, this benefit is achieved without making any changes to the router's configuration.

The benefits provided by Self–Learning IP can also be realized in conjunction with the advantages availed by micro–segmentation as depicted in Figure 18 below. In this example, stations are moved from shared router segments to dedicated 8371 segments. The 8371 can then switch both the intrasubnet traffic and the intersubnet traffic.

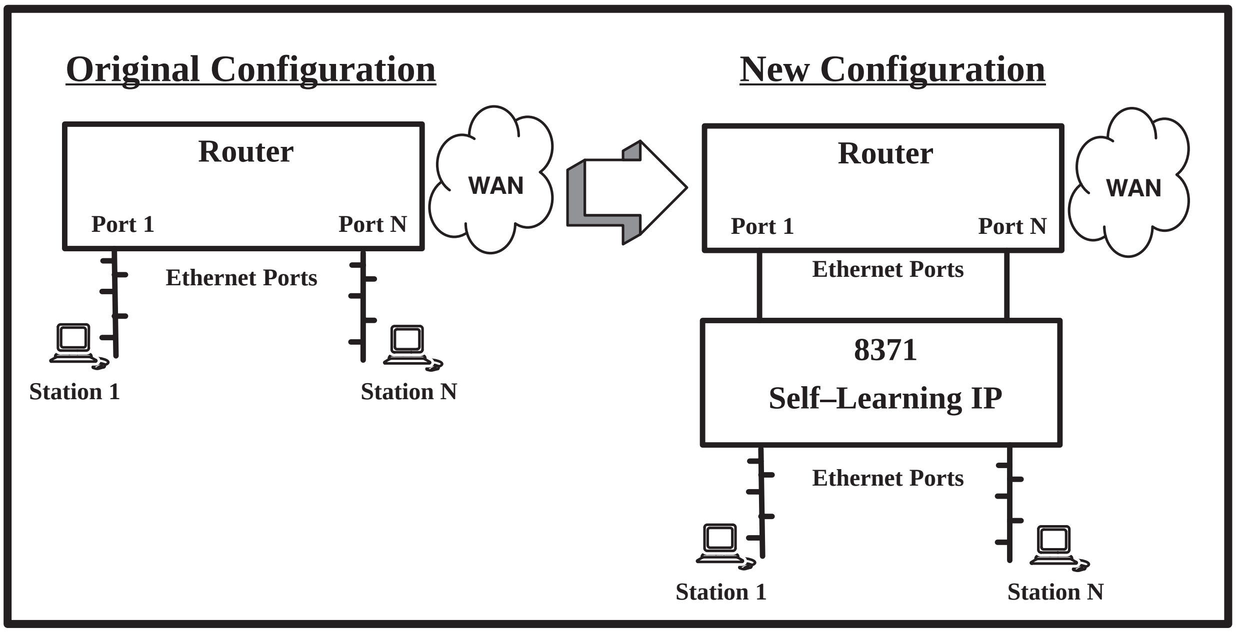

Figure 17. Automatic Router Offload with Self–Learning IP

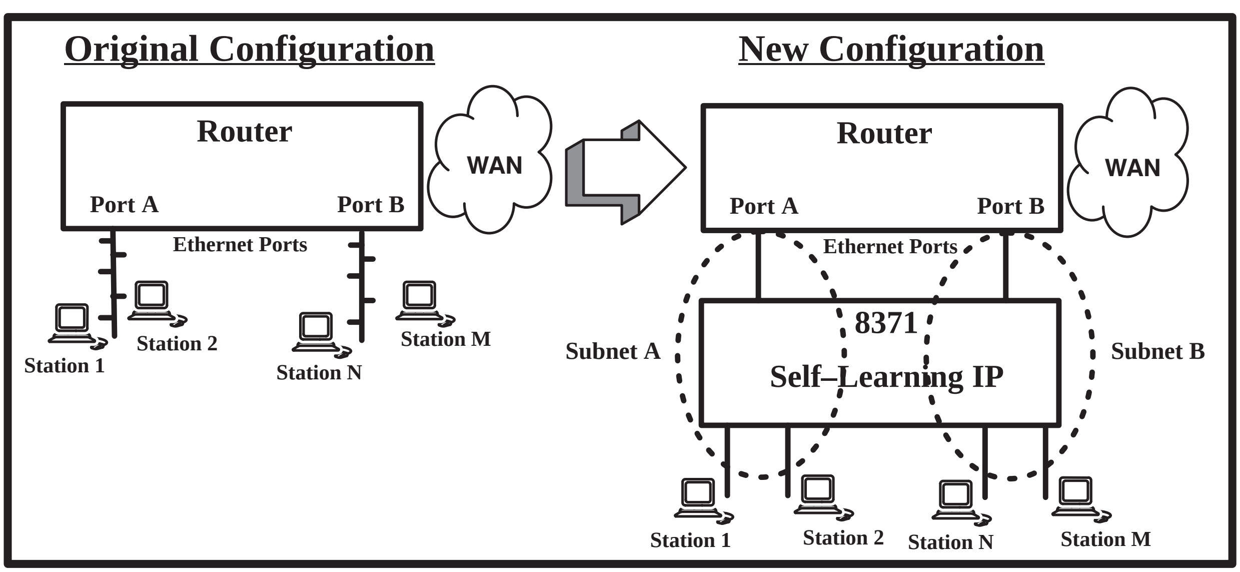

Figure 18. Combining the Benefits of Micro–Segmentation and Self–Learning IP

Figure 19 shows another configuration where Self–Learning IP is advantageous. Here, stations are moved from an Ethernet port on the router to an Ethernet port on the 8371 switch, but the router is only connected to the 8371 switch with a single Ethernet port, thereby reducing the number of expensive LAN ports that are required on the router. Furthermore, the connection to the router can be a 10 Mbps link, while all the downstream ports are upgraded to 100 Mbps performance. Setting up such a network would, however, require changes to the router's configuration.

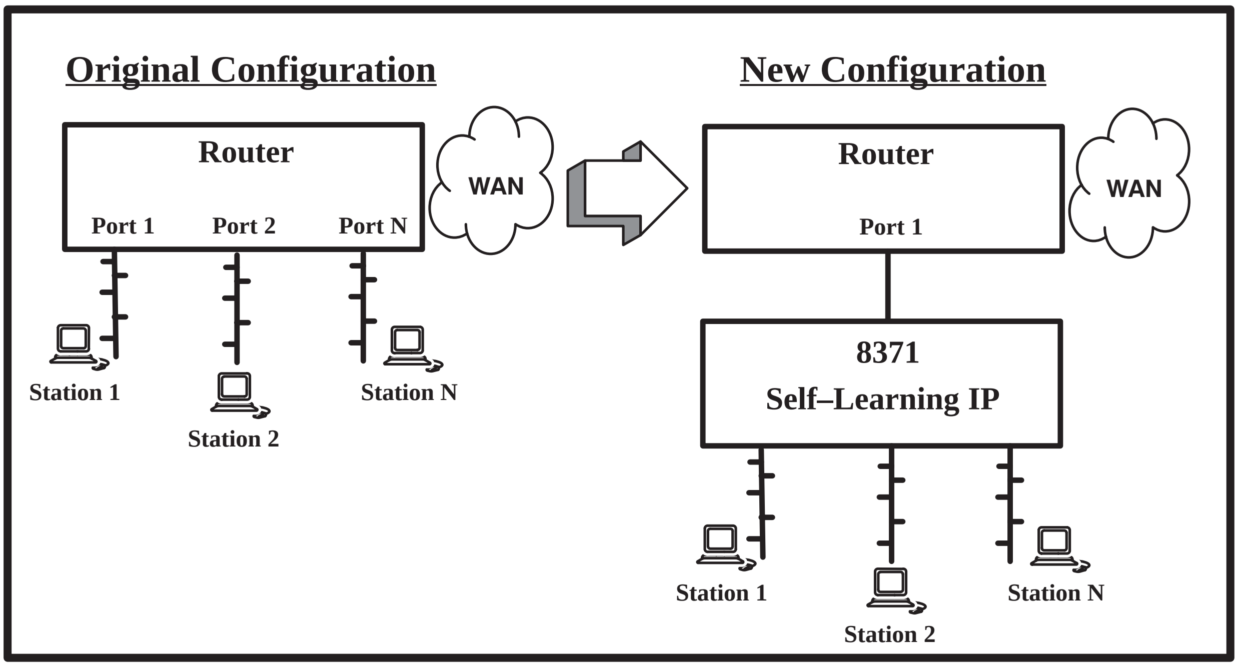

Figure 19. Consolidating Router Ports with Self–Learning IP

Figure 20 below provides a detailed illustration of Self–Learning IP forwarding operations. Assume that Self–Learning IP has discovered two router MAC addresses, MAC_X and MAC_Y, and that the forwarding database already contains entries for $Station_A$ and $Station_N$. In step 1, $Port_3$ receives a frame transmitted by $Station_A$. At layer–2, the frame is addressed to $Station_A$'s default router. When $Port_3$ looks up MAC_X, it will recognize that the MAC address is that of a router, which will trigger a search of the forwarding database for the destination IP address, IP_N. The entry for IP_N is found in step 2. Then, in step 3, a new MAC header is built using information from the forwarding database entry, and the frame is switched out $Port_4$, which completes the forwarding operation. Note that when the frame is received by $Station_N$, the frame appears exactly as it would if it had been transmitted by the router (e.g., the TTL is decremented and the checksum updated accordingly).

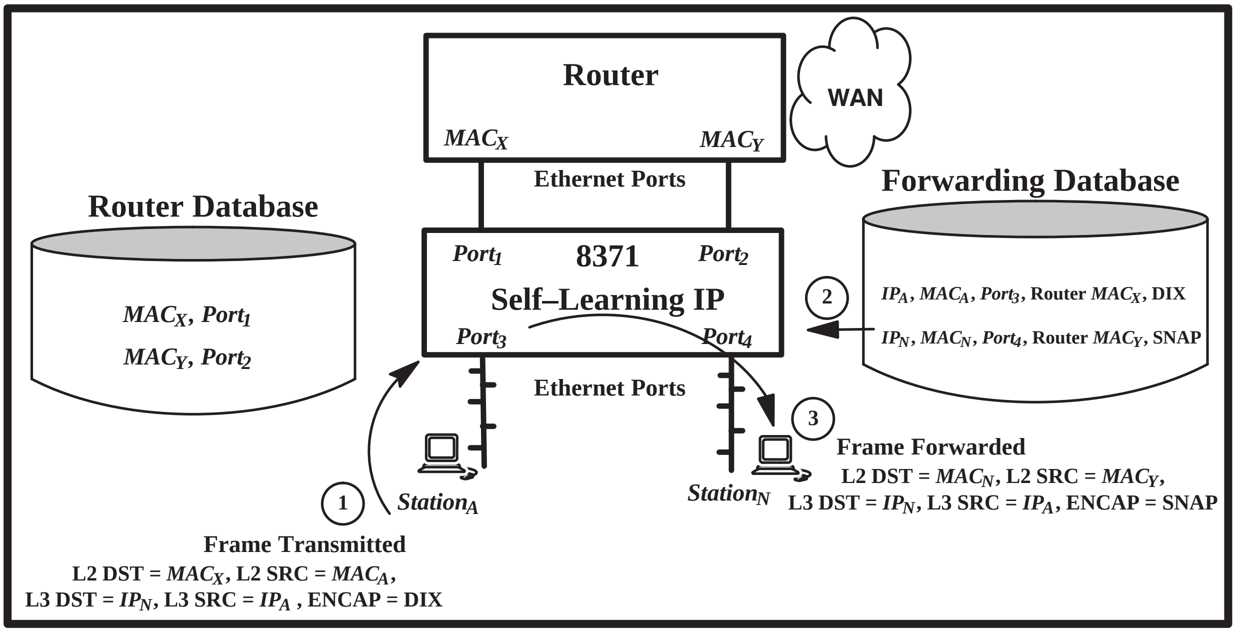

Figure 20. Self–Learning IP Forwarding Operation Example

More specifically, the CPU maintains a global Router Database and a global Forwarding Database. The Router Database contains an entry for each router that has been discovered. The complete contents of the Router Database is replicated at each port. The Forwarding Database is a master table that contains all of the routes that have been learned on any of the ports. Only a subset of the Forwarding Database contents is replicated at each port. Routes are downloaded to ports only when they are needed to forward frames. As an example, let's consider Figure 20 again and examine what would happen if $Port_3$'s local database did not contain an entry for IP_N. In this case, $Port_3$ would bridge the frame to the router via $Port_1$ and send a destination flow detection indication to the CPU. The flow detection indication would trigger the CPU to download the forwarding information for IP_N. Subsequent frames destined for IP_N that are received by $Port_3$ would then be shortcut out $Port_4$.

Self–Learning IP learns routes that are inserted into the Forwarding Database by snooping on (i.e., examining the contents of) ARP Reply frames. Each time a new forwarding entry is learned from an ARP frame, Self–Learning IP sends a *probe* to try and determine whether this new station is a router. The probe is an ICMP Request with a TTL of 1. If either a *Time Exceeded* or a *No Route to Host* response is received from the station, then the station is identified as a router, and an entry for the station is inserted in the Router Database.

Self–Learning IP also learns new forwarding entries by snooping on the contents of routing protocol frames transmitted by adjacent routers. Information is gleaned from both RIP and OSPF protocols.

Self–Learning IP will periodically delete entries in the Forwarding Database that are not actively being used to forward packets. Similarly, entries in the Router Database will also be deleted if the association of the MAC address with a router is not refreshed over time.

Self–Learning IP is implemented as a new *feature* of the Nways Common Code. As such, it has a configuration interface and a console interface. The configuration interface is very simple, consisting of *enable* and *disable* commands. Self–Learning IP can also be dynamically enabled or disabled via the console interface, subject to the constraint that Self–Learning IP was enabled at boot–time. Additionally, the console interface provides commands for displaying the contents of the Router Database and the IP Forwarding Database.

The configuration and console commands can be accessed from both the command line and web interfaces.

To facilitate plug–and–play operation, Self–Learning IP provides a mode that partitions the Ethernet ports into pairs representing separate bridging domains. Ports 0 & 1 are paired, ports 2 & 3 are paired, and so on. Broadcast and unknown frames received by port 0 are only forwarded to port 1, and vice versa. This mode of operation is tailored to support the configuration illustrated in Figure 17, where a router is connected to one port and the stations that were previously connected to the router are moved to the paired port of the 8371 switch. Naturally, this partitioned mode of bridging operation can be disabled in order to support configurations such as those depicted in Figure 18 and Figure 19.

5. CONFIGURATION AND CONSOLE INTERFACES

A variety of tools are available for configuring and managing the 8371 switch. The tools include a command line interface, a web browser interface, and a SNMP interface to network management stations.

Both in–band and out–of–band access methods are supported. Out–of–band access can be made locally through the serial service port or remotely via an external modem. Out–of–band access supports both SLIP and TTY connections.

The command line interface provides complete configuration, as well as comprehensive monitoring and status reporting facilities. It can be accessed using a TTY connection, or via Telnet over a SLIP or in–band IP connection.

The web interface enables the command line functions to be accessed via a more user–friendly graphical interface. Web browsers may access the 8371 via SLIP or in–band IP connections.

A wide range of monitoring and management functions are also supported by network management stations that interact with the SNMP agent in the 8371 (see Section 6. for additional information on network management).

While the 8371 obviously possesses extensive configuration capabilities, it was designed to support *plug–and–play* operation through a set of default configurations. All possible network interfaces are automatically configured at boot–time. Thus, the *Device ADD* and *Device DELETE* commands are no longer needed (and have been removed from the configuration menus). Similarly, the *ADD* and *REMOVE* commands have also been deleted from the LE Client configuration menu. A list of the network interfaces available on the standalone 8371 switch is provided in Table I below.

Table I Network Interfaces Automatically Configured on Standalone 8371 Switch

Device Type	Slot	Ports	Interface Net Numbers
10/100 Ethernet (fixed ports on base switch)	0	1–16	0–15
10/100 Ethernet Feature Card	1	1–8	16–23
10/100 Ethernet Feature Card	2	1–8	24–31
Reserved	3	1–4	32–35
ATM	1	1–2	36–37
ATM	2	1–2	38–39
Ethernet LAN Emulation Client	3	5–29	40–63

Note that all of the above interfaces cannot be active at the same time. For example, if Ethernet feature cards are installed in both slots 1 and 2, then there is no place to install ATM interfaces. Similarly, a maximum of two ATM interfaces are supported in Release 1.0 of the 8371. On the standalone 8371 product, an ATM feature card with two OC–3 interfaces may be installed in either slot 1 or slot 2, but ATM feature cards are not supported in both slots simultaneously.

On the 8371 blade module, the OC–12 connection to the 8265 backplane occupies slot 1, and only Ethernet feature cards are supported in slot 2. This alters the list of available network interfaces as indicated in Table II below.

Table II Network Interfaces Automatically Configured on 8371 Blade

Device Type	Slot	Ports	Interface Net Numbers
10/100 Ethernet (fixed ports on base switch)	0	1–16	0–15
Reserved	1	1–8	16–23
10/100 Ethernet Feature Card	2	1–8	24–31
Reserved	3	1–4	32–35
ATM (connection to 8265 backplane)	1	1	36
Reserved	1	2	37
Reserved	2	1–2	38–39
Ethernet LAN Emulation Client	3	5–29	40–63

In addition to the network interfaces, the transparent bridge is also automatically configured. By default, all the Ethernet interfaces and all of the LECs are configured as ports on the bridge. Configured interfaces that are disabled, or not actually present, are simply inactive bridge ports.

Only one of the 24 automatically configured Ethernet LEC interfaces is enabled by default. The LEC with an interface number of 40 is enabled, while the LECs with interface numbers 41–63 are disabled. Each of the LECs is configured as follows, where *i* is the interface number of the LEC and *j* = (*i* –39):

ATM Interface: 36
ELAN Name: ELAN*j*
LECS Auto–Config: YES
MAC Address/ESI: burned–in MAC address for net *i*
Selector: 2

Note that:

- all of the LECs are initially associated with the same ATM interface (interface number 36),
- the LECs are setup to use the auto–configuration facilities of the LECS to locate their LES,
- each LEC is assigned a unique MAC address, and
- the MAC address is also used as the ESI component of the LEC's ATM address.

As part of the autoconfiguration effort, a new *assign–lec* console command has been created. Since LECs can no longer be *ADDed*, the *assign–lec* command was created to allow LECs to be assigned to particular ATM interfaces.

Not only is each LEC given a unique burned–in MAC address, but so is each Ethernet interface and each ATM interface.

Autoconfiguration also includes the MPOA layer–3 switching function. Layer–3 autoconfiguration occurs when no configuration records are found. When this situation exists and there is an ATM interface installed, the MPOA Client is automatically enabled.

As a result of the above–described autoconfiguration features, the 8371 switch comes up in a plug–and–play manner with no configuration required for either bridging or MPOA operation.

6. NETWORK MANAGEMENT AND RMON

A comprehensive set of standard and enterprise–specific MIBs is available for monitoring and managing 8371 resources. All of this network management data is accessible via SNMP. The 8371 uses the same SNMP agent software as the MSS products. The 13 MIBs supported by the 8371 are listed below; of these, 8 are IETF or ATM Forum standards, and 5 are enterprise–specific MIBs.

6. 1. IETF Standard MIBs

- MIB II, specified in RFC 1213
- MIB II Extension, specified in RFC 1573b
- Ethernet MIB, specified in RFC 1650
- AToM MIB, specified in RFC 1695
- RMON MIB, specified in RFC 1757
- Bridge MIB, specified in RFC 1493

6. 2. ATM Forum Standard MIBs

- LAN Emulation Client MIB
- MPOA Client MIB

6. 3. Enterprise–Specific MIBs

- IBM LAN Emulation Extension MIB
- IBM Common Routing MIB
- IBM Netview 6000 MIB
- IBM CPU Utilization MIB
- IBM RMON Private MIB
- Proteon MIB

6. 4. RMON–lite

This subsection contains more–detailed information about the RMON functions provided in the 8371 switch.

In Release 1.0 , the 8371 implements *RMON–lite*, which is defined to be Groups 1, 2, 3, and 9 of the RMON I MIB (defined in RFC 1757).

Group 1 is the *Statistics Group*, which includes the following counters in the *etherStatsTable*:

DropEvents:	number of events in which packets were dropped due to lack of resources
Octets:	number of received octets of data
Packets:	number of received packets (including bad packets, broadcast packets, and multicast packets)
BroadcastPackets:	number of good broadcast packets received
MulticastPackets:	number of good multicast packets received
CRCAlignErrors:	number of properly sized packets received with either a CRC error or an alignment error
UndersizePackets:	number of well formed packets received that were less than 64 octets in length
OversizePackets:	number of well formed packets received that were greater than 1518 octets in length
Fragments:	number of undersized packets received that also had either a CRC error or an alignment error
Jabbers:	number of oversized packets received that also had either a CRC error or an alignment error
Collisions:	estimate of the total number of collisions
Pkts64Octets:	number of packets received that were 64 octets in length
Pkts65to127Octets:	number of packets received that were between 65 and 127 octets in length
Pkts128to255Octets:	number of packets received that were between 128 and 255 octets in length
Pkts256to511Octets:	number of packets received that were between 256 and 511 octets in length
Pkts512to1023Octets:	number of packets received that were between 512 and 1023 octets in length
Pkts1024to1518Octets:	number of packets received that were between 1024 and 1518 octets in length

The preceding statistics are maintained in hardware for each Ethernet interface.

Group 2 is the *History Group*, which controls sampling of the interface statistics. Two tables are used: *historyControlTable* and *etherHistoryTable.* Entries in the *historyControlTable* specify the interface for which samples are to be maintained, the interval at which samples are to be taken, and the number of buckets that are available for storing history records. The Ethernet statistic records are actually recorded in the *etherHistoryTable* (i.e., an entry in the *historyControlTable* contains counters corresponding to counters in the *etherStatsTable*). Up to 16 entries can be created in the *historyControlTable*. All of the entries share a common pool of 1024 history record buckets.

Group 3 is the *Alarm Group*. which controls the monitoring of MIB values. Both rising and falling thresholds can be specified for a variable, and the threshold comparisons may be based on either absolute or delta values. As an example, assume that the *Collisions* variable is being monitored for a particular interface, and that an alarm is desired if the number of collisions increases by more than 1000 from one sample to the next. This can be achieved by simply inserting an entry in the *alarmTable* that specifies delta values and a rising threshold of 1000. Then, when the condition is encountered, a *rising–alarm event* will be generated. Indices of the events to be generated in the case of rising or falling alarms are also stored in the *alarmTable*. In Release 1.0 of the 8371, the number of *alarmTable* entries that can be created is limited to 30.

Group 9 is the *Event Group*. which controls the action taken when particular events are generated. The action may be: (1) create a *logTable* entry, (2) send an SNMP trap, or (3) create a *logTable* entry and send an SNMP trap. Entries in the *logTable* identify the event that generated the entry, specify the time that the entry was created, and include a user–defined character string description of the event. 8371 Release 1.0 supports up to 75 *eventTable* entries, and a maximum of 32 *logTable* entries per event.

Both configuration and console interfaces are available for enabling/disabling the RMON function.

7. MISCELLANEOUS FUNCTIONS

This section covers miscellaneous functions that have not been addressed in any of the previous sections.

7. 1. LAN Emulation Client QoS

The 8371 switch supports reservation of ATM bandwidth on the Multicast Send VCC that a LEC establishes to the BUS and on Data Direct VCCs established between two LECs. The capability for a LEC to reserve ATM bandwidth is currently available on multiple IBM products including the MSS Server, MSS Client, 2216, and 2210, in addition to the 8371. The LAN Emulation Client QoS function, which was first introduced in MSS Server Release 1.1, is the feature that enables bandwidth to be reserved on Data Direct VCCs. With this feature, traffic parameters can be defined on an ELAN–wide or individual–LEC basis.

7. 2. Hot Swap

A *hot–swap* capability is provided as part of 8371 Release 1.0. This capability supports feature card replacement while the remaining 8371 interfaces are still functioning. The capability is termed *hot–swap* instead of *hot–plug* because of the following two restrictions: (1) a feature card that is hot–swapped must have been present when the 8371 was most–recently booted, and (2) a feature card can only be hot–swapped with another feature card of the same type.

7. 3. Mutual Exclusion of MPOA and Self–Learning IP Functions

The MPOA Client and Self–Learning IP functions are mutually exclusive in that only one of the functions may be enabled at any given time. The decision regarding which layer–3 function to activate is made at boot–time and cannot be altered dynamically.

8. PERFORMANCE DATA

This section contains a sampling of performance measurements for the 8371. The data clearly indicates that the 8371 is a high–performance LAN switch.

Throughput for layer–2 switching between two 100 Mbps Fast Ethernet ports is shown in Figure 21 below. Media speed, 148.8 Kpps, is achieved for all frame sizes. The aggregate layer–2 switching throughput of a 8371 equipped with 32 Fast ethernet ports exceeds 4.7 Mpps.

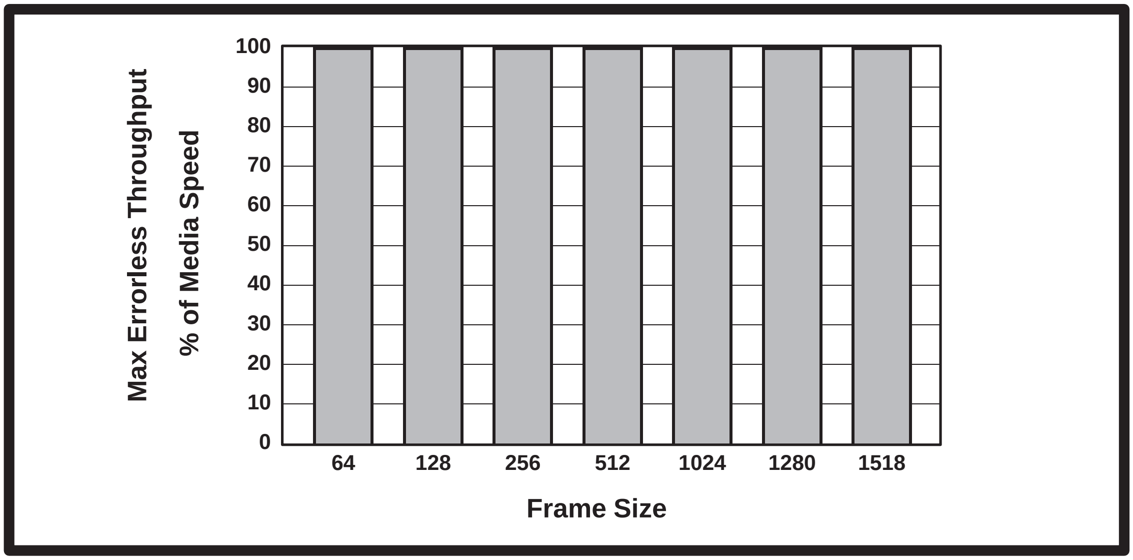

Figure 21. Layer–2 Switching Throughput

Throughput for MPOA layer–3 switching between two 100 Mbps Fast Ethernet ports is illustrated in Figure 22 below. The data shown is for an IP local shortcut (i.e., for IP data being switched between two Ethernet ports on the same 8371 switch). Performance is similar across an ATM backbone (i.e., when the Ethernet ports are on different 8371 switches connected via ATM). Local shortcuts are noteworthy because they significantly increase the cumulative layer–3 forwarding capacity of the switch (by avoiding unnecessary traversal of the ATM interface/network). The aggregate MPOA layer–3 switching throughput of a 8371 equipped with 24 Fast Ethernet ports exceeds 3.2 Mpps.

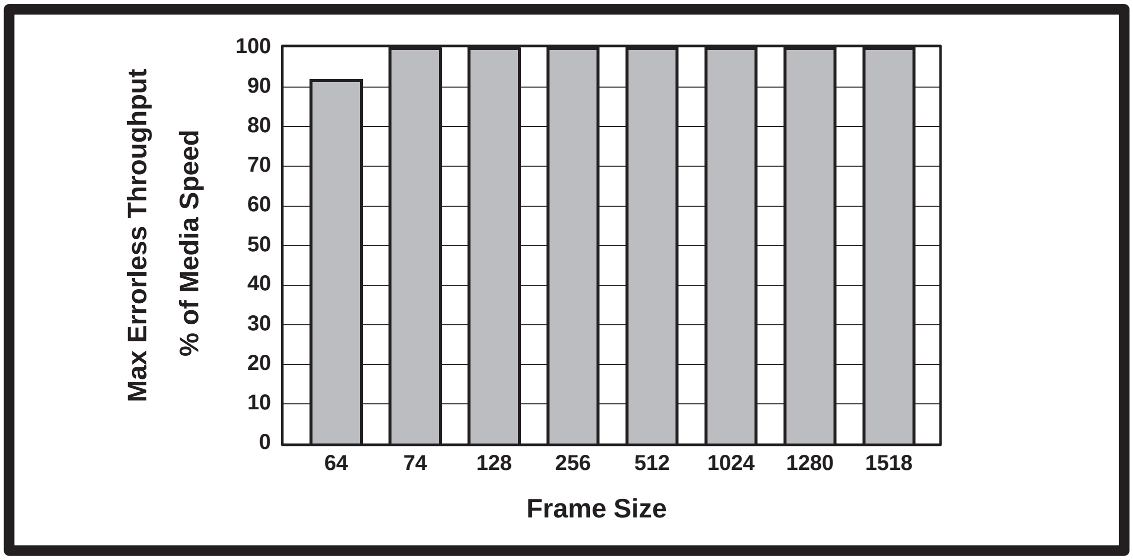

Figure 22. MPOA Switching Throughput

Throughput for Self–Learning IP layer–3 switching between two 100 Mbps Fast Ethernet ports is shown in Figure 23 below. Media speed is achieved when frame sizes are at least 128 bytes. The aggregate Self–Learning IP switching throughput of a 8371 equipped with 32 Fast Ethernet ports exceeds 3.1 Mpps.

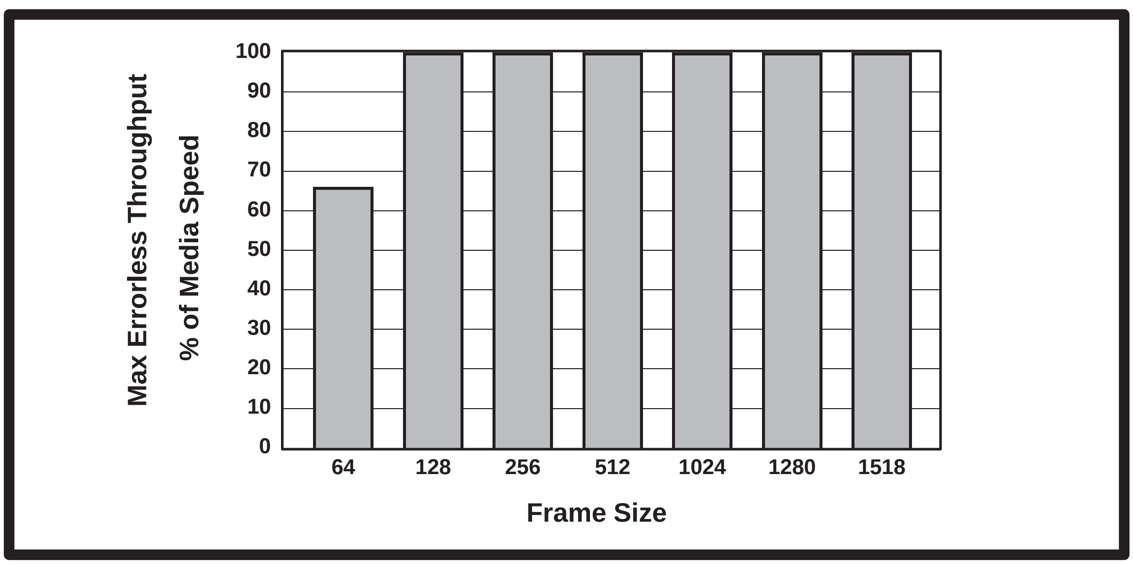

Figure 23. Switching Throughput for Self–Learning IP

To further illustrate the potential performance improvements that can be realized with the Self–Learning IP feature, consider offloading an IBM Nways 2210 Model 24E Router with a 8371 switch as shown in Figure 24 below.

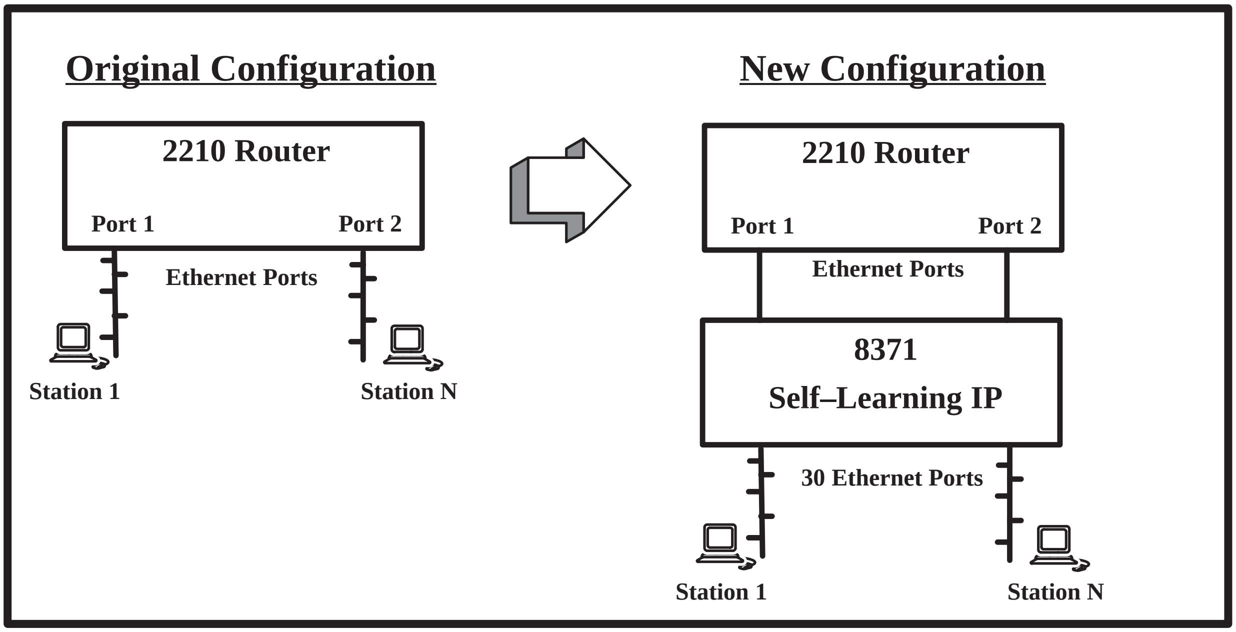

Figure 24. Offloading an IBM 2210 Router with Self–Learning IP

This model of the 2210 can route 64–byte IP frames between two 10 Mbps Ethernet ports at media speed (~15 Kpps), but doing consumes 95% of the CPU. Offloading the 2210 with a 32–port 8371 increases the layer–3 forwarding capacity by a factor of 30, while also reducing the 2210 CPU utilization to 5%. If the end–station segments are also upgraded to 100 Mbps, then the performance improvement factor increases to 200 and the 2210–8371 combination forms a 3 Mpps router.