Introduction to Port Extension

Joe Pelissier
Cisco Systems
3 West Plumeria Drive
San Jose, CA 95134
+1.503.628.0801
jopeliss@cisco.com

***Abstract* – Data centers today are experiencing a rapid proliferation of switches as a direct result of the deployment of high-density server platforms. Given the layered switching architecture found in today's data centers, many of the switches are performing a simple aggregation function; that is, the majority of traffic is moving between downlinks and uplinks. However, despite this simple function, these switches contribute to a significant portion of the capital expenditure and ongoing administrative and management costs of the data center. This paper proposes a new technology that replaces these aggregating switches with a device that extends the ports of the switch in the next higher layer. This technology has the potential to reduce significantly the number of switches that must be managed in the data center as well as reduce the upfront capital expenditure costs.**

I. Introduction and Motivation

Data centers today are experiencing a dramatic increase in the number of installed Ethernet switches as a direct result of the deployment of high-density servers and blade servers. In addition, deployment of virtualization technology within servers has resulted in an even further increase in the number of installed switches. These switches are typically embedded within the actual server itself. It is important to note, however, that virtualization is not the sole source of this explosion in switch proliferation, but it has added significantly to this phenomenon.

The growth in switch deployment has resulted in the corresponding growth of associated costs.

It may also be observed that many of these switches serve little function other than to act as aggregation points within the network. They frequently perform minimal frame relay outside of the simple passing of frames between uplinks and downlinks. In effect, these switches are simply acting as additional ports for higher-level switches. Despite the fact that these switches are performing a relatively simple function, they account for much of the initial capital expenditure and ongoing management and administrative costs.

To address these costs, a new technology has been proposed referred to as "Port Extension". The Port Extension technology introduces a new device called a "Port Extender" that effectively acts as additional ports for the switch to which it is connected. The switch to which the Port Extender is connected is referred to as the "Controlling Switch". The Controlling Switch and a set of Port Extenders connected to it form a single logical switch. Port Extenders are not individually managed but instead are managed as part of the combined switch entity. To the extent possible, all switching functions are performed in the Controlling Switch. This keeps the functionality of Port Extender limited and therefore the cost low. This also helps to keep the functionality of the Controlling Switch consistent across all ports realized using Port Extenders.

II. PROBLEM STATEMENT

The deployment of hundreds to thousands of switch devices with varying capabilities and performance as a result of high-density server technology (including but not limited to server virtualization) creates the following challenges that are addressed by the technology proposed by Port Extension:

- High network management complexity and administrative cost
- High initial capital expenditures
- Stressed scalability limits and responsiveness of network applications due to:
  - Quantity of points of management
  - Quantity of management messages required for each point
  - Latency in response to each management message
- Lack of visibility of internally switched frames
- Server and network management conflicts

III. REQUIREMENTS

It was considered extremely important that the Port Extension technology provides exactly the same externally observable behavior of that provided by switches deployed in networks today. This is fundamental to ensuring interoperability. Deviating from such behavior opens the door for many unforeseen consequences. Furthermore, any requirement that an application or device be aware it is connected to a switch using Port Extension technology was seen as a significant barrier to acceptance and therefore highly undesirable.

It was also considered highly desirable to drive complexity towards the Controlling Switch and out of the Port Extenders. This complexity would include functions such as address lookups, learning and aging functions, VLAN functions, and access control list processing. The most obvious reason for this requirement is to reduce overall cost. Since Port Extenders will outnumber Controlling Switches, it clearly makes sense to focus on reducing the cost of the Port Extenders. Equally important, however, is the fact that simplifying the Port Extender results in greater data center design and dynamic reconfiguration flexibility by reducing inconsistency of functionality of the network infrastructure.

It is also required that the ports instantiated by Port Extenders operate with any device that could normally be connected to a standard switch port. This includes other switches, end stations, and other Port Extenders. In other words, the ports provided by Port Extenders must operate as any other port that is part of the Controlling Switch.

Additionally, maintaining simplicity in the Port Extenders is a requirement to provide simple and efficient management capabilities. As a Controlling Switch becomes a single point of management for itself and all attached Port Extenders, the total number of points of management in the network is substantially reduced. More importantly, by keeping the majority of the functionality in the Controlling Switch, few management operations require the Controlling Switch to initiate additional protocol operations with the Port Extenders. For example, configuration of an ACL on a group of ports affects only the Controlling Switch and not the associated Port Extenders. Thus, the total number of management messages required and the latency of each of these messages is significantly reduced.

Finally, to promote and simplify the development of Controlling Switches, Port Extension technology is designed to operate using the fundamental architecture common in existing switches today.

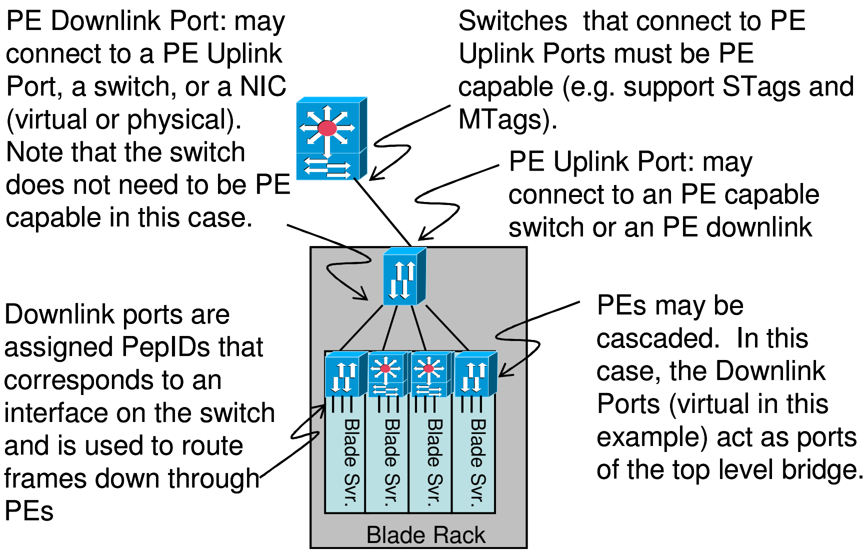

PE Downlink Port: may connect to a PE Uplink Port, a switch, or a NIC (virtual or physical). Note that the switch does not need to be PE capable in this case.
Switches that connect to PE Uplink Ports must be PE capable (e.g. support STags and MTags).
PE Uplink Port: may connect to an PE capable switch or an PE downlink
Downlink ports are assigned PepIDs that corresponds to an interface on the switch and is used to route frames down through PEs
PEs may be cascaded. In this case, the Downlink Ports (virtual in this example) act as ports of the top level bridge.
Blade Svr.
Blade Svr.
Blade Svr.
Blade Svr.
Blade Rack

Figure 1 – Port Extension Anatomy

IV. An Approach

Port Extension technology proposes to meet the previously stated requirements by providing the capability to combine distributed network components into a single logical 802.1Q compliant switch. These components consist of:

- A Controlling Switch
- Distributed Port Extenders (that may be cascaded)
- A protocol enabling control of the Port Extenders by the Controlling Switch

Figure 1 illustrates a network utilizing Port Extension technology.

The top switch in this figure represents the Controlling Switch. One of the ports on this switch is connected to a Port Extender. The Port Extender port that is connected to a Controlling Switch, or to a higher level Port Extender within a cascade, is referred to as an uplink port. This Port Extender is connected to four additional devices below it. Two of these devices are additional Port Extenders. The other two devices are conventional switches. The Controlling Switch and the three Port Extenders shown in this figure combine to form a single logical switch. The ports of the Port Extenders that connect to lower level Port Extenders within a cascade, or to other devices outside of the logical switch, are referred to as downlink ports. In a cascade of Port Extenders, the downlink ports of Port Extenders in one layer of the cascade connect to the uplink ports of the Port Extenders in the next lower layer of the cascade. Thus, the topology formed by a cascade of Port Extenders is a loop-free tree.

Figure 2 illustrates the logical network that is achieved by the physical network illustrated in figure 1.

As previously mentioned, to the greatest extent possible, all switching functions are performed in the Controlling Switch. However, many of these functions are port based, that is, they require knowledge of the ingress port or require an explicit indication of the egress port. To convey this knowledge, an additional tag is added to the frame much like a VLAN tag. In addition, each Port Extender Port is assigned a 12-bit Port Extender Port ID (PepID) by the Controlling Switch (this occurs upon port creation in the case of ports connected to virtual machines). Finally, to facilitate flooding, multicast, and broadcast, each Port Extender may be programmed with multiple port lists by the Controlling Bridge. A Port Extender Port List ID (PepLID) identifies each Port list.

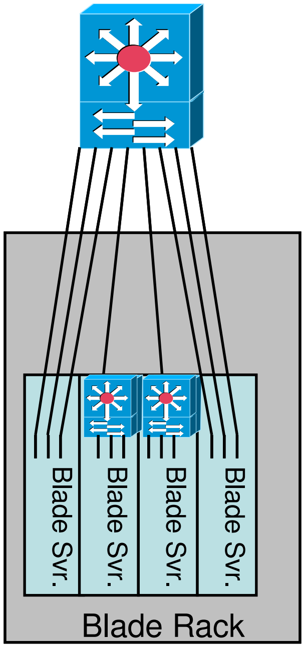


Figure 2 – Equivalent Logical Network

The assigned PepIDs of all Port Extender ports accessible through a given Controlling Switch port must be unique. PepIDs are qualified by the Switch port through which they are accessed; therefore, PepIDs of Port Extender ports accessible through independent switch ports are not required to be unique.

V. Controlling Switch and Port Extender Architecture

Figure 3 illustrates the architecture typical of many Switches deployed in data centers.

These Switches consist of line cards that provide the individual ports and a crossbar function that provides the relay of frames between line cards.

When a frame enters such a switch, an indication identifying the ingress port is passed along with the frame to the memory controller and frame processing functions.

The frame processing functions utilize the indication of the ingress port along with various fields in the frame headers to perform the necessary processing to determine to which port or ports the frame should be relayed. This processing may include functions such as validation of the frame's VLAN, verification that the frame passes the requirements imposed by access control lists for the ingress port, and look up of the frame's MAC address and the VLAN identifier within the switch's filtering database. In addition, learning functions are performed to associate the frame's source MAC address and the VLAN identifier with the ingress port.

After determination of the egress port (or ports) is made, an indication of the egress port is associated with the frame data. The frame is then scheduled for transmission through the crossbar function. The actual scheduling algorithm used may be based on a wide variety of factors but frequently includes the ingress and egress port indications.

At the appropriate time, the frame, along with its ingress and egress port indications, is transmitted to the crossbar function. The crossbar forwards the frame to the line card containing the port specified in the egress port indication.

The egress line card and port performs additional frame processing. This may include egress access control list processing, egress VLAN processing, and the addition and removal of tags. The frame may then be scheduled for transmission from the port specified in the egress port indication.

Figure 4 illustrates the same switch architecture with the addition of Port Extenders in both the ingress and egress paths. The key point to note is that no change to the fundamental switch architecture is required to support the Port Extenders.

In this case, the frame enters the Port Extender and a STag is added to the frame.

The format of the STag is defined in IEEE 802.1ad [1]. The STag format is very similar to that of a QTag. It contains a three-bit Priority Code Point indication, a Drop Eligible Indicator bit, and a 12-bit "Service VLAN ID" (SVID). The priority field is simply copied from the frame's QTag, or set to the Port Extender's port default priority if a QTag is not present. The Drop Eligible Indicator bit is set or reset based on a static programming of QTag priority. The SVID is set to the PepID of the ingress port.

After the frame is tagged with a STag, the Port Extender forwards the frame to its uplink port. Only the first Port Extender on an ingress path adds the STag; successive Port Extenders forward the frame toward the Controlling Switch.

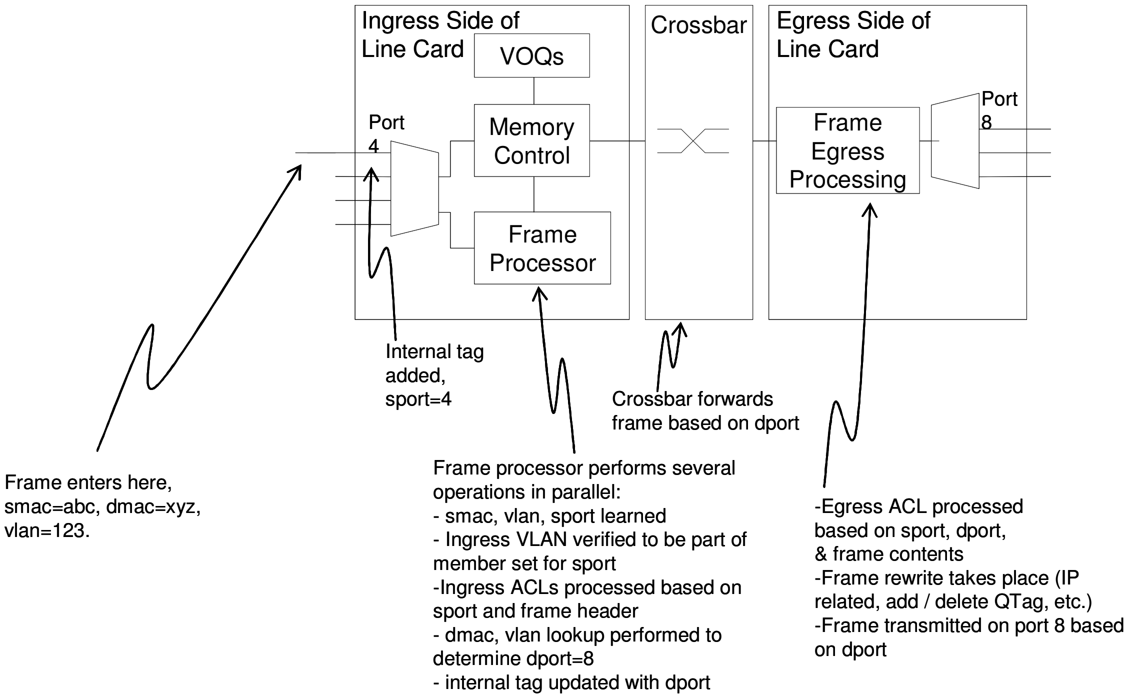
Upon ingress into the Controlling Switch, the Port Extender ingress port PepID is combined with the indication of the Controlling Switch ingress port. This combined value is then used for all ingress port identification in the switch frame processing functions. In particular, the learning function learns the combination of the ingress Port Extender PepID and the Controlling Switch ingress port indication.

The frame processing functions determine an egress port. However, in this example the egress port is a port on a Port Extender. Therefore, the egress port indication returned by the frame processing function contains the combination of an indication of the Controlling Switch egress port and the Port Extender egress port PepID. This combination is used as the egress port indication for all frame processing functions within the Controlling Switch.

Upon transmission from the Controlling Switch, the STag is updated with the Port Extender egress port PepID. Each Port Extender contains a forwarding table that is indexed by PepID that is programmed by the Controlling Switch. Each entry in the table contains the physical port index through which a frame with the corresponding PepID is to be transmitted.

On egress, the frame transits through one or more Port Extenders. At each Port Extender, the PepID is used as an index into the forwarding table and the egress Port Extender port is obtained. The frame is then transmitted from that port. The last Port Extender removes the STag.

To support multicast and frame flooding, each Port Extender contains a Port Extender Port List table. Each entry in the table indicates a unique group of ports to which a frame on egress is to be replicated. If the Controlling Switch determines that the frame is to be flooded, contains a group MAC address, or needs to be forwarded to multiple ports for other reasons, the Controlling Switch replaces the STag with a new tag called an "MTag". Like the STag, the MTag contains a Priority Code Point field and Drop Eligible Indicator bit. These values are simply copied from the STag. In addition, the MTag contains a PepLID field and a source PepID. The source PepID is also copied from the STag of the original frame. To cause the frames to be forwarded to the proper ports, the Controlling Switch populates the PepLID field with the appropriate value. The frame is then forwarded to the appropriate Port Extender. Each Port Extender along the egress path forwards the frame to the ports indicated in the port list pointed to by the PepLID. The final Port Extenders remove the MTag. In addition, the final Port Extenders check the source PepID in the MTag and discard the frame if it matches that of the egress port. This prevents a frame from being forwarded on the port from which it was received.



Internal tag added, sport=4

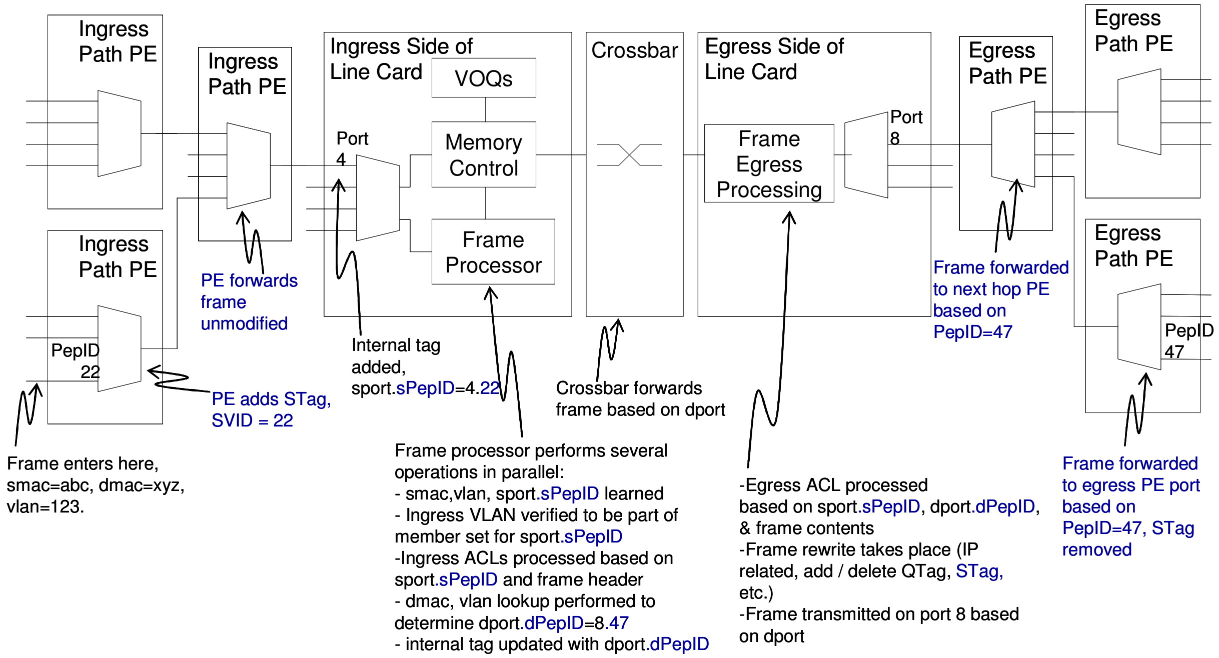
Crossbar forwards frame based on dport

Frame enters here, smac=abc, dmac=xyz, vlan=123.

Frame processor performs several operations in parallel:
- smac, vlan, sport learned
- Ingress VLAN verified to be part of member set for sport
-Ingress ACLs processed based on sport and frame header
- dmac, vlan lookup performed to determine dport=8
- internal tag updated with dport

-Egress ACL processed based on sport, dport, & frame contents
-Frame rewrite takes place (IP related, add / delete QTag, etc.)
-Frame transmitted on port 8 based on dport

Figure 3 – Typical Switch Architecture



PE forwards frame unmodified

PE adds STag, SVID = 22

Frame enters here, smac=abc, dmac=xyz, vlan=123.

Internal tag added, sport.sPepID=4.22

Crossbar forwards frame based on dport

Frame forwarded to next hop PE based on PepID=47

Frame forwarded to egress PE port based on PepID=47, STag removed

Frame processor performs several operations in parallel:
- smac,vlan, sport.sPepID learned
- Ingress VLAN verified to be part of member set for sport.sPepID
-Ingress ACLs processed based on sport.sPepID and frame header
- dmac, vlan lookup performed to determine dport.dPepID=8.47
- internal tag updated with dport.dPepID

-Egress ACL processed based on sport.sPepID, dport.dPepID, & frame contents
-Frame rewrite takes place (IP related, add / delete QTag, STag, etc.)
-Frame transmitted on port 8 based on dport

Figure 4 – Typical Port Extension Capable Switch Architecture

VI. Data Center Network Design Hierarchy

The design of modern data center networks is based on a proven layered approach, which has been tested and improved over the past several years in some of the largest data center implementations in the world. The three layers of a data center network are:

- Core layer, the high-speed packet switching backplane for all flows going in and out of the data center.
- Aggregation layer, providing important functions such as the integration of network-hosted services: load balancing, intrusion detection, firewalls, SSL offload, network analysis, and more.
- Access layer, where the servers physically attach to the network and where the network policies (access control lists [ACLs], quality of service [QoS], VLANs, etc.) are enforced. The access-layer network infrastructure can be implemented with either large, modular switches, typically located at the end of each row, providing connectivity for each of the servers located within that row (the end-of-row model,) or smaller, fixed configuration top-of-rack switches that provide connectivity to one or a few adjacent racks and have uplinks to the aggregation-layer devices (the top-of-rack model.) Bladed server architectures modify the access layer by allowing an optional embedded blade switch to be located within the blade enclosure. Blade switches, which are functionally similar to access-layer switches, are topologically located at the access layer; however, they are often deployed as an additional layer of the network between access-layer switches and computing nodes (blades), thus introducing a fourth layer in the network design [2].

Sever virtualization technologies logically divide a single server (or blade) into multiple servers. To achieve this, a switch (implemented in software, hardware, or a combination of both embedded within the server) provides connectivity to each virtual machine. Thus, a fifth layer is added to the network architecture that is effectively a third sub-layer of the access layer.

This architecture results in an extensive proliferation of switches that are providing little service other than the simple forwarding of frames between uplinks and downlinks. Yet these are full function switches and each must be fully managed (if only to disable much of the switch functionality). This creates an excessive management and capital expenditure burden with marginal return.

However, the functionality provided by a Port Extender is exactly the functionality required in these switches, i.e. the ability to forward traffic between uplinks and downlinks. Thus, these switches may be replaced by Port Extenders effectively collapsing the switch hierarchy into the next higher-level switch. As a result, the number of switches that are required to be deployed and managed is dramatically reduced.

Additionally, the capability of such a collapsed network is frequently much greater. In a traditional network design, the switches located near the edge of the network are those that face the greatest cost pressure. This is understandable given that they occur in the greatest quantity. Yet this is precisely where the most advanced functionality is required. Conversely, the switches that are located away from the edge tend to be fewer in number, and therefore the cost pressures are not as great. Thus, these switches tend to provide much greater capability. Ironically, these switches are not optimally located to provide the advanced functionality.

Port Extenders correct this “capability inversion”. By replacing switches at the edge of the network with Port Extenders, the higher layer more capable switches effectively become the edge switches. As a result, their advanced capabilities may be fully deployed at the edge where they are most needed and effective.

VII. Port Extension and Virtualization

Port Extension provides unique benefits in server virtualization environments in addition to the benefits provided in traditional data center networks.

A key capability of server virtualization is the ability to move an active virtual server from one physical machine to another (referred to as virtual machine migration). This capability is commonly provided by virtualization software.

The network complicates the migration process. Each virtual machine requires certain characteristics of the switch port that connects it to the fabric. This could include parameters such as flow control, congestion notification, VLAN assignment, access control lists, etc. This collection of parameters is commonly referred to as the virtual machine's "port profile".

There is no standard process for transferring a port profile from one switch to another in synchronization with the migration of a virtual machine. Virtual machine migration often involves a manual step of pre-provisioning the switch port within the target physical server. Thus, the efficiency of the migration is negatively impacted.

If the migration is to take place between two ports of a given switch, it is quite trivial for the switch to simultaneously transfer the port profile. Immediately after a migration, a virtual machine typically broadcasts a frame to "announce" its new location and to allow switches in the network to update their filtering databases (a RARP frame is typically used for this purpose). Since the virtual machine has moved to a new port on the same switch, upon reception of the announcement frame, the switch may simply move the port profile from the old switch port to the new one. Since no coordination is required between switches, no standard protocols or procedures are required.

Unfortunately, virtual machines in today's data centers nearly never migrate between ports on a given physical server. Recall that each physical server contains its own embedded switch. It rarely makes sense to migrate a virtual machine within a given server (such a migration is essentially meaningless). Thus, migrations are nearly always between physical servers, and by definition, between switches.

However, if the switch embedded within each physical server is replaced by a Port Extender, and these Port Extenders are connected to a common Controlling Switch, then all of the virtual machines within all of the virtual servers are, in effect, connected by a single switch. As a result, migration between the physical servers is enabled without manual pre-provisioning of the port profiles.

VIII. Summary

Because of high-density server technology, modern data centers are experiencing a dramatic increase in the number of deployed switches. This results in increased capital expenditure and management costs while stretching the scalability limits of the management applications. Many of these switches perform little frame relay other than between adjacent layers in the network architecture. Port Extension allows these switches to be removed from the network resulting in significantly fewer switches being acquired and managed. In addition, port extension extends the reach of the more capable higher layer switches to the edge of the network allowing these capabilities to be more effectively utilized. Finally, port extension enhances the efficiency of virtual machine migration by eliminating the need for manual pre-provisioning of network port resources and configuration.

IX. References

[1] IEEE Computer Society, *IEEE Std 802.1ad™-2005, IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks, Amendment 4: Provider Bridges,* 3 Park Avenue, New York, NY 10016-5997, USA, 26 May 2006.

[2] Cisco Systems, Inc., "Cisco VN-Link: Virtualization-Aware Networking, A Technical Primer", http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns892/ns894/white_paper_c11-525307_ps9670_Products_White_Paper.html, page 1.