ABridges: Scalable, self-configuring Ethernet campus networks

G. Ibáñez *, A. García-Martínez, A. Azcorra, I. Soto

Departamento de Ingeniería Telemática, Universidad Carlos III, Madrid, Spain

Received 21 December 2006; received in revised form 30 October 2007; accepted 30 October 2007
Available online 6 November 2007

Responsible Editor: T. Tugcu

Abstract

This article describes a scalable, self-configuring architecture for campus networks, the ABridges architecture. It is a two-tiered hierarchy of layer two switches in which network islands running independent rapid spanning tree protocols communicate through a core formed by island root bridges (ABridges). ABridges use AMSTP, a simplified and self configuring version of MSTP protocol, to establish shortest paths in the core using multiple spanning tree instances, one instance rooted at each core edge ABridge. The architecture is very efficient in terms of network usage and path length due to the ability of AMSTP to provide optimum paths in the core mesh, while RSTP is used to aggregate efficiently the traffic at islands networks, where sparsely connected, tree-like topologies are frequent and recommended. Convergence speed is as fast as existing Rapid Spanning Tree and Multiple Spanning Tree Protocols.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Ethernet; Routing bridges; Shortest path bridges; Multiple spanning tree protocol

1. Introduction

Ethernet technology dominates campus and enterprise networks due to the excellent price/performance ratios and backward compatibility among different versions and speeds. Additionally, bridges do not require any kind of address configuration to perform frame forwarding. These characteristics push for the deployment of Ethernet in bigger domains such as campus networks, although some issues arise due to the lack of scalability of standard bridge protocols, like the Spanning Tree Protocol. This limited scalability derives from two main factors: low link utilization and vulnerability of the bridged domain to network failures and configuration errors. On one hand, network infrastructure utilization is low because the loop prevention mechanism of the spanning tree protocol relies on the activation of just a subset of the available links. As a consequence, the resulting routes along the spanning tree are not pair-wise shortest paths due to the low connectivity. On the other hand, a hardware failure or configuration error may produce

* Corresponding author. Tel.: +34 609177932; fax: +34 913070143.
E-mail addresses: gibanez@aut.uah.es (G. Ibáñez), alberto@it.uc3m.es (A. García-Martínez), azcorra@it.uc3m.es (A. Azcorra), isoto@it.uc3m.es (I. Soto).

broadcast storms and even network meltdown of the switched domain.

Current wide-campus Ethernet deployments rely on IP routers to segment the network in order to limit the size of bridged domains and to prevent total disruption in case of failure. The drawbacks of this approach are the requirement of proper IP address and segment configuration, and the restrictions imposed to host mobility inside the network, since the IP address of a host must be modified when its point of attachment to the network changes.

To foster the deployment of all-Ethernet campus-wide networks several requirements have to be fulfilled, in particular those related with self-configuration and scalability to large networks (up to 20,000 hosts or more). Scalability is jeopardized by the traffic generated due to ARP broadcasts and link layer broadcast of IP multicasts, and by the inability of the bridges to manage the large number of MAC addresses. For large networks, ARP broadcasts result in hosts processing a significant number of ARP packets, most of them not targeted to it, which are responsible for substantial processing overhead and considerable bandwidth consumption [15]. The large amount of traffic generated by the layer-two broadcast of IP multicast traffic is a consequence of the inability of the switches to learn the multicast group addresses from frames carrying MAC addresses that cannot be used as source. Additionally, increasing the number of hosts may require a corresponding increase of the expensive cache memory of the switches, and produces frequent broadcasts of frames that can overload the hosts and the network. Finally, transparency to hosts and routers is essential, so that easy deployment can be achieved.

Protocols to extend Ethernet capabilities are currently under discussion at IETF and IEEE, with diverse approaches and requirements. On one hand, the so called RBridges [1] are proposed at the IETF as hybrid devices composed of routers and bridges, benefiting from the advantages provided by routers while preserving at the same time the automatic configuration capability of bridges. However RBridges currently do not explicitly aim or are required to scale up to campus-wide networks. On the other hand, the Shortest Path Bridging (SPB) proposal is under consideration at the IEEE working group 802.1aq [9], showing a slower protocol convergence than RSTP due to the mechanisms required to ensure symmetrical spanning tree instances. Additionally, the SPB configuration is complex and its scalability limited to 32 nodes.

In this paper we propose an architecture for high performance, scalable self-configuring Ethernet campus networks. It consists of a two-tier hierarchy of switches: a core of enhanced switches (ABridges) that interconnects a number of separate access networks (islands) formed by standard switches. The core provides failure isolation so that routers are not required. The whole campus network can then be deployed as a single IP segment and the hosts may move within the campus keeping the same IP address, without reconfiguration.

The remainder of this paper is structured as follows. Since AMSTP convergence mechanism, packet formats, etc. are partly derived from IEEE 802.1D and 802.1Q, these standards are presented in Section 2. Section 3 describes the overall network architecture, presenting the ABridge functionality, and the basic forwarding mechanism, along with other relevant components of the architecture such as ARP/ABridge servers. In Section 4 we detail the new and standard protocols used in this architecture, namely RSTP for the access layer, and AMSTP for the core layer. Since simple management is one of the key requirements of the paper, Section 5 deals with the management requirements imposed by the architecture. Section 6 contains a comparative analysis of AMSTP with other protocols and performance evaluation. Section 7 describes summarily the related work and finally Section 8 contains the conclusions.

2. IEEE 802.1 spanning tree standards

The most recent IEEE standards for the Spanning Tree Protocols exhibit fast reconfiguration and other advanced features that preserve and maintain compatibility with previous protocols. The standards of the IEEE 802.1 series related to the problem considered are 802.1D [3] for standard bridges and single spanning trees and 802.1Q [4] for VLANs and multiple spanning trees. The latest revision [3] of the 802.1D MAC Bridges standard has adopted the Rapid Spanning Tree Protocol as a replacement of the legacy and slow Spanning Tree Protocol (SPB, [5]).

2.1. RSTP

RSTP was defined to provide much faster convergence than the previous standard protocol STP.

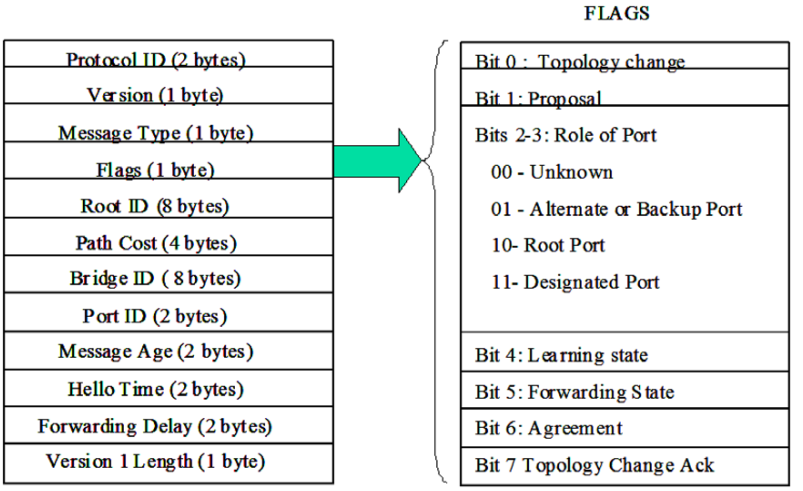

Fig. 1. RSTP BPDU layout.

RSTP achieves convergence in (typically) fractions of second because the old timer based mechanism to negotiate the transition of ports to the *forwarding* state, is replaced by a local negotiation between adjacent ports of neighboring switches. The ports are activated in the downstream direction of the spanning tree in a controlled way, enabling one level of links of the tree at a time. When the root port of a bridge agrees the transition to the *forwarding* state with its pair port (i.e. the designated port of its *parent* bridge), the designated ports of the (lower) bridge are previously blocked and stop forwarding frames to downstream bridges. This guarantees loop-free network start-up but requires point-to-point links to prevent loops. Fig. 1 shows the BPDU format and the flags byte used for port state negotiation.

2.1.1. MSTP

The Multiple Spanning Tree Protocol (IEEE standard 802.1Q) [3] uses RSTP to create different tree instances that are associated to sets of VLANs according to the configuration of the bridge. MSTP builds a set of multiple and independent spanning tree instances (MSTI) at each defined network region. Each region is interconnected via a common spanning tree (CST) to other MST regions. At each region, there is an Internal Spanning Tree (IST), identified with the number 0, which behaves as the basic spanning tree in order to provide compatibility. The Common and Internal Spanning Tree (CIST) or total spanning tree is comprised of the CST that connects all the regions, and the ISTs that provide basic connectivity inside each region. Inside a region, several VLANs can be mapped to each MSTI. The use of multiple tree instances on each region can improve the utilization because the links unused for a tree instance may be active for another tree instance.

Each region appears to the outside as a unique and separate "superbridge", since the whole region connects to the CST via one Regional Root Bridge port and a number of designated ports. Therefore, changes in the topology internal to a region do not affect to the higher level topology, and viceversa, as long as the connectivity through the Regional Root Bridge is preserved.

3. ABridges campus network architecture

We define a *campus network* as the set of network elements placed along separate buildings belonging to a common organization, connected to one or more WAN routers. The routers establish the limits of the campus network. Current campus network design practice is based on a model in which three infrastructure layers are considered: *core*, *distribution* and *access* (Fig. 2) [11]. This architecture differentiates the devices (switches) used at each layer in order to obtain optimum costs and to provide network scalability and predictability under reconfiguration [12]. Network segmentation is obtained using

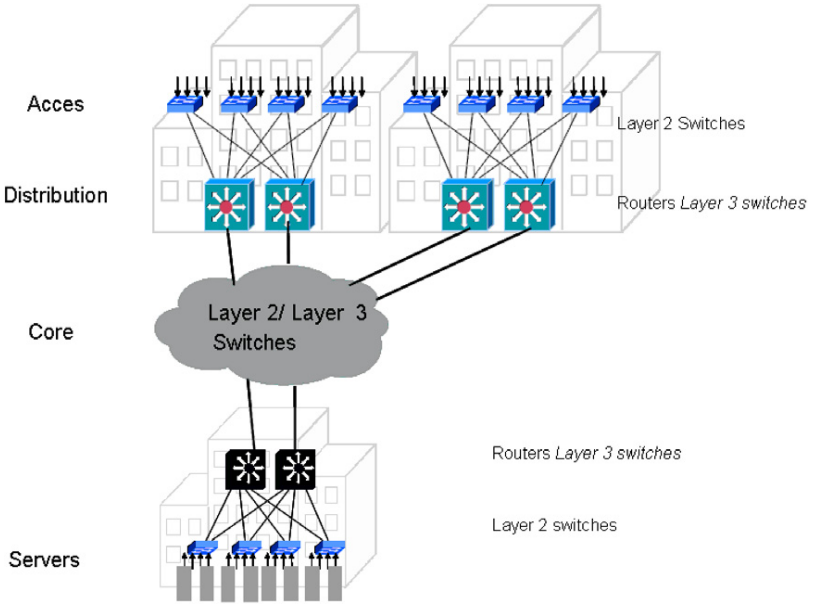

Fig. 2. Three layer campus network architecture.

routers or router-like devices called *multilayer switches* that split the network into IP segments or subnets.

The proposed ABridge campus network architecture consists of a two-level hierarchical link layer infrastructure in which segmentation is performed at link layer instead of splitting the infrastructure at network layer. In this way, IP routers are no longer required. Fig. 3 shows the generic network topology. A core of ABridges constitutes the campus backbone and interconnects different access networks or islands, formed by standard 802.1D bridges.

The upper layer behaves as a *core–distribution* layer (or *Core* for brevity) that connects the leaf access networks that are referred to as *access layer* (in short, *Access*). The core bridges (Abridges) use the Alternative Multiple Spanning Tree Protocol (AMSTP), a self-configuring and simplified version of MSTP protocol, to set up a network among

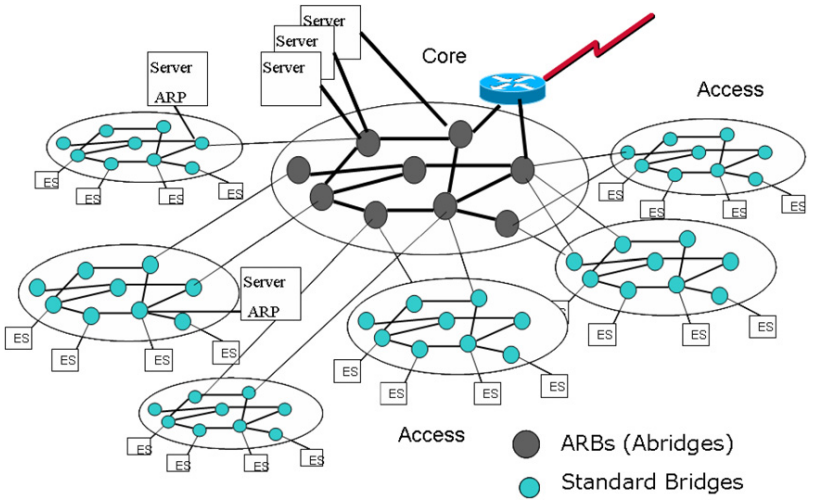

Fig. 3. Two-layer campus network architecture with ABridges.

them. ABridges require point to point links between them. Each ABridge builds a self-rooted tree instance used to forward frames toward the root ABridge in unicast and from the root ABridge in broadcast. Each *access network* or *island* is a layer-two sub-network made of standard 802.1D bridges using a standard Spanning Tree Protocol. The access network is connected to one or more ABridges operating in 802.1D mode. One ABridge of the access network is automatically elected as Root Bridge of the island spanning tree. The root elected ABridge behaves as a *gateway*, forwarding packets from the core to the island (access), and vice versa. Note that many Abridges may be connected to the access network, although only one is performing gateway functions at a given time. Communication among 802.1D bridges and between standard 802.1D bridges and ABridges does not require point-to-point connections.

The ABridge receiving an ARP frame from an island host obtains the ABridge (island) in which the destination is located by asking an ARP server where the host was previously registered by its island ABridge. This server stores the IP to MAC mapping and the island ABridge ID. The frames ingressing at the core are encapsulated with an additional layer two header that includes the destination and origin ABridge IDs. Frames are decapsulated by egress ABridge and forwarded into the destination island. The ARP servers distribute its load based on equal result of short hashing of the IP addresses served. The core self-configures and the operation is transparent to all hosts and standard switches at islands.

3.1. ABridges functionality

Fig. 4 shows the basic functional modules of an ABridge. These modules are

- the STD Bridge Module, that performs the standard bridging functions with the nodes of its island network,
- the AMSTP Routing Module, which routes between Abridges, and
- the Gateway module, that interconnects the STD module and the AMSTP Routing Module.

Frames access to the core through the STD Bridge Module, are processed by the Gateway module and enter the AMSTP Routing Module where they are forwarded towards the next ABridge in the core, till the Egress ABridge is reached. At this point, the frames are processed inversely.

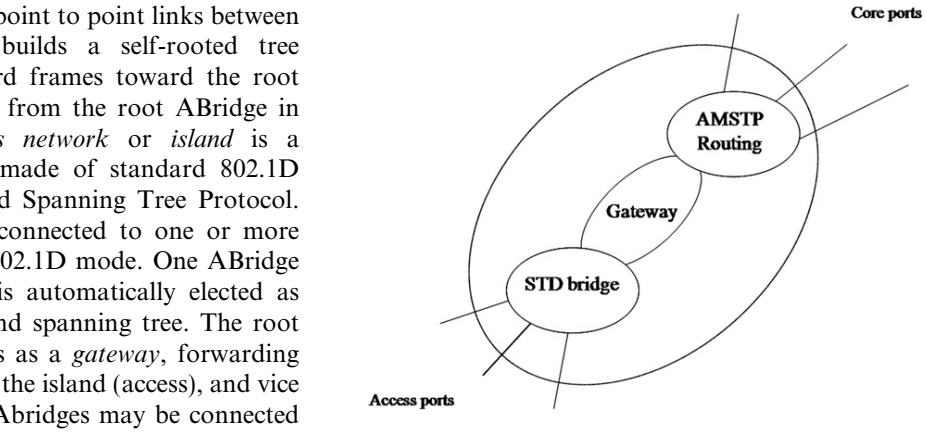

Fig. 4. ABridge functional decomposition.

The AMSTP Routing Module has *core ports*. Every core port connects the ABridge to another Abridge. The access functionality resides at *access ports* of the STD bridge module, and in this case the behavior is equivalent to a standard bridge acting as root bridge of its access network connected to the *access* ports. ABridges learn in which access ports are located the end nodes in the same way as standard bridges do.

In the AMSTP Routing module, ABridges learn root bridge IDs and root ports of the multiple core tree instances from the AMSTP BPDUs received, and store this information in their forwarding database (FDB). Frames with destination to the same access network of a given ABridge will be forwarded only between STD bridge access ports. Frames toward other access networks ingress the core via the Gateway Module, are encapsulated with destination address egress ABridge and forwarded to the AMSTP routing section. After this, the encapsulated frame is routed using the forwarding database constructed by the AMSTP protocol until the last ABridge is reached, where the outer header is eliminated and the packet is introduced in the destination access network.

ABridges auto-configure each port to be part either of the Core or of the Access network. The port auto-configuration mechanism is performed as follows: a port that is not connected using a point-to-point link to another ABridge configures itself as an *access port.* It executes the standard spanning tree protocol and provides connection to the Access Network. Ports directly connected to

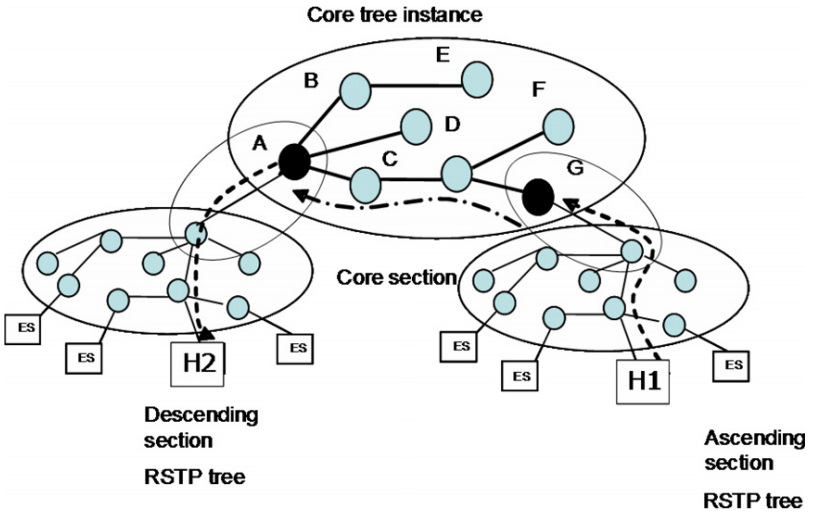

Fig. 5. End to end forwarding use case.

another ABridge are configured as *core ports*. The auto-configuration mechanism for ports is similar to the standard protocol migration state machine of RSTP [3] where ports execute STP or RSTP protocol according to the type of protocol BPDUs received by port. The version field indicates whether the protocol is STP or RSTP.

3.2. ARP and ABridge resolution

The broadcast-based ARP protocol is the standard method to obtain the link layer address associated to an IP destination at the same LAN segment. Broadcast frames are flooded by bridges to the whole campus network, resulting in excessive traffic flooding and processing load at hosts that should be minimized. To limit broadcast/multicast traffic, the use of distributed load ARP servers is recommended in the ABridges architecture, although its usage is optional.[1]

Consider the sequence for communication between hosts H1 and H2 depicted in Fig. 5: Host H1 first sends a broadcast ARP packet to get the resolution of the link layer address of host. The frame is distributed through the spanning tree of the access network and arrives at its root bridge, the island ABridge.

The flow of frames is detailed at Fig. 6. The root ABridge intercepts the ARP request packet, calculates *hash* (*IP destination address*) and with some bits of the hash it obtains from a table the link layer address of the ARP server responsible for the IP addresses of this hash value. ARP servers may be connected anywhere, although direct connection to different ABridges optimizes server traffic. The hashing mechanism used to select the ARP server for a given destination enables the distribution of load among the active servers. The IP to be resolved is hashed with a few bits hash length (i.e. three bits if eight servers are used). Once determined the corresponding server, the root ABridge encapsulates the ARP frame with its own address as origin and destination the server for that IP address. The server performs a look up using the IP destination address of H2 and obtains the LL address of H2 and the associated (egress, destination) ABridge ID of the destination access network, then encapsulates the ARP response and sends it to the ingress ABridge. The ABridge extracts the information, forwards a standard ARP response packet to host H1 and stores in its cache the pair *LL destination host–destination ABridge ID*.

The ingress ABridge also registers at the server, if required, the originating host by sending a registration packet containing the ARP packet to the corresponding ARP/ABridge server, obtained similarly by computing *hash*(*IP origin*), keeping the servers updated with latest host location information. The ABridge registers a host at the corresponding ARP Server/Registrar whenever it detects a frame

[1] An alternative approach to resolve ARP and destination ABridge is the exchange of host lists among the ABridges. This requires more processing, bandwidth and memory at the ABridges (50K tuple search for a 25 island network with 2K hosts per island).

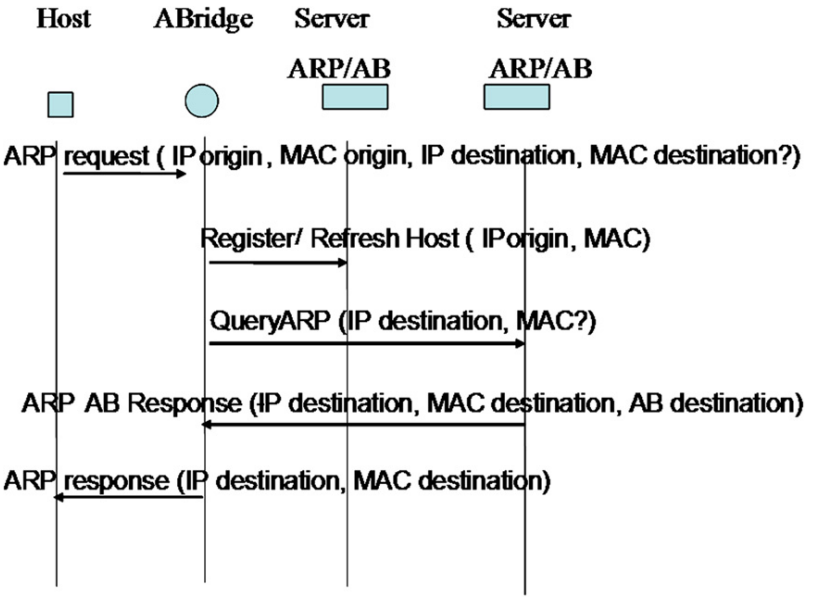

Fig. 6. ARP and ABridge resolution.

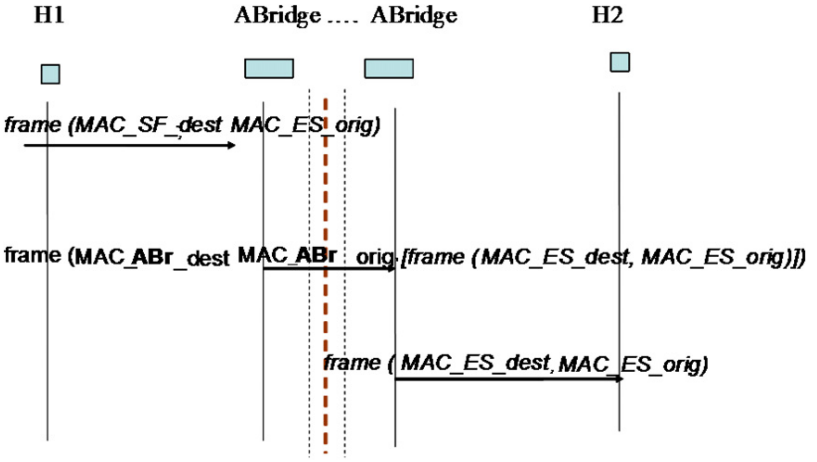

Fig. 7. End to end packet forwarding scenario.

from an unknown host connected at its access network.

Host H1 can then proceed to send packets with the LL address of host H2. Fig. 7 shows the flow of user packets. The process is detailed below in Section 3.3.

Note that upon negative or no response to its unicast ARP request to ARP server, the ingress ABridge will broadcast the ARP request across the core and, consequently, across all the access networks. The preservation of the standard ARP mechanism ensures host resolution in case of either ARP failures or host mobility.

It is worth to consider that when the destination host is connected to the same access network, the host will reply to the ARP Request directly by emitting an ARP response packet and the server response is not strictly necessary.

We finally discuss some issues related with ARP server management. Active servers announce periodically by multicasting to the ABridges multicast address its link layer address and the hash value

or values served. In order to do this, a new link layer multicast address, the *all-ABridge-and-ARP-servers*, is defined. Servers recently booted are included into the active set of ARP servers as follows: When a new server boots, it listens to the other server announcements and selects among them the most relatively loaded server to share its load with it, by taking over responsibility for one or several hash results from the loaded server. After a transition period of shared responsibility under old server control, the last host registration cache expires and the new server contains all valid host registrations so that the old server keeps no valid registration for that hash(IP) value(s). ABridges listen to the servers announcement at servers multicast address to learn the server in charge for every hash result.

The periodic announcements sent by each server to the all-Abridges-and-ARP-servers address allow the detection of failures by other servers. As a consequence, a failure of a server is detected by the remaining ARP servers and all the ABridges of the network. When an ABridge requires ARP resolution for an IP address whose hash(IP) points to the failed server, it recomputes the hash in this case using hash(IP+1), obtaining the new server that will attend this request (if the result points to a failed server, hash(IP+2) is computed, etc., until a valid server is obtained for the IP address. Note that this algorithm distributes among the remaining servers the load of the resolution of the addresses formerly resolved by the failed element.

3.3. Forwarding

Forwarding of encapsulated frames in the core is not based on the standard MAC address learning mechanism, but on forwarding through tree instance to destination ABridge. The first ABridge receiving the frame encapsulates it into an additional link layer header containing its Bridge ID MAC as source address, and the Bridge ID of the egress ABridge as destination MAC. This Bridge ID was obtained previously and cached from the response of an ABridge server, as described above. As shown in Fig. 8, the ingress ABridge forwards the encapsulated frame through the branch belonging to the spanning tree instance rooted either at ingress or egress ABridge. The tree instance rooted at egress ABridge is used to forward unicast traffic. Forwarding takes place by sending the frame through the ABridge root port for destination ABridge. This path is a shortest path because the tree is built by minimizing path cost from each root to the rest of the nodes. The tree instance rooted at the ingress ABridge is used to propagate multicast traffic, broadcast traffic, or frames to unicast destinations not known by ABridge. In this case, the frame is forwarded via all designated ports.

An encapsulated packet in the core network looks like an Ethernet frame but must be differentiable by ABridges from a native Ethernet frame. To accomplish this, a new link layer protocol type (“Ethertype”) is used. An encapsulated packet looks as shown in Fig. 9. Besides the additional layer two header, a standard 802.1 Frame Check Sequence (FCS) field is appended (for checksum verification) at the end of the original frame to complete the encapsulated frame that will transit through the core network.

4. Protocols

In this section we describe the protocols used in the proposed campus network architecture. The protocol for core layer is the Alternative Multiple Spanning Tree Protocol (AMSTP). AMSTP is an evolution of the standard Multiple Spanning Tree (MSTP) [4] and Rapid Spanning Tree (RSTP) [2,3] protocols. Different protocols can be used at the access layer networks, although the standard proto-

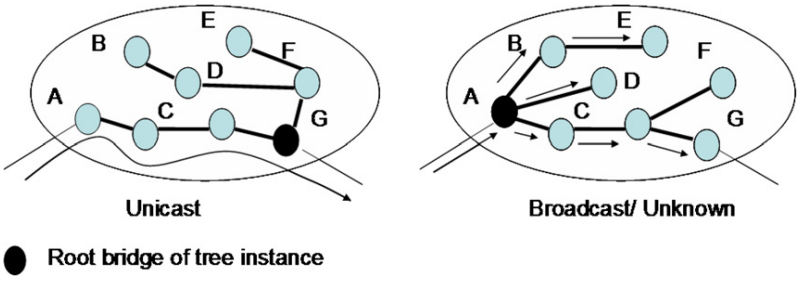

Fig. 8. Frame forwarding in core.

DA	*SA*	*Type*	DA	SA	Type	PDU	FCS	FCS
G	*A*	*AMSTP*	H2	H1	Type	PDU	FCS	FCS

Fig. 9. Frame encapsulation (core) format and frame example for Fig. 8.

col RSTP is the default and recommended protocol due to its efficiency and performance with the dominant client–server traffic.

4.1. Access layer protocol

To connect the access networks with the core, ABridges play a double bridging function, acting both as core bridges and as gateway bridges for the connected access networks.

When RSTP is used for the access network, an ABridge must be the root bridge of the tree built by the RSTP at its island, as shown for Abridges A and G. in Fig. 5. Note that being the root bridge of the spanning tree (ST) is an efficient way for a bridge to perform as gateway, since in this case all paths entering and exiting the core exhibit minimum cost.

To guarantee, without any configuration, the election of an ABridge as root bridge of the access network spanning tree, it is sufficient the default priority of ABridges to be lower (higher preference) than the default priority of standard bridges (mid-range value in the 802.1D standard). In case of failure of the root ABridge, other ABridge, the one with the lowest MAC address (assuming equal default ABridges priority), will take over as root bridge of the access network and will act as gateway to the core. So the root bridge election procedure at the access network is also used to determine the single ABridge that performs gateway functions to the core or *designated* ABridge.

4.2. Core layer protocol

In the architecture proposed, the new AMSTP Protocol provides both segmentation and shortest path interconnection between the islands or access networks. A preliminary version of AMSTP Protocol was proposed in [6] for metropolitan Ethernet backbones, that now is extended for campus networks with significant improvements. AMSTP is a simplified multiple spanning tree protocol that uses one tree instance rooted at each edge bridge in the core to forward frames. A *complete multi-tree* is the set of all the tree instances rooted at every edge bridge that interconnects the bridges in the backbone. We describe now how AMSTP builds and maintains the spanning trees that are used for frame forwarding in the core.

In order to build the trees, the AMSTP protocol relies in a basic tree, that is used to obtain the rest of the instances, named Alternate Multiple Spanning Tree Instances (AMSTI), until one tree instance per bridge is built as shown in Fig. 11 for the network of Fig. 10.

The process of building the main tree is the same as in RSTP. First, a bridge is elected as root bridge of the core network. Every bridge emits autonomously Bridge Protocol Data Units (BPDU) every Hello Time (configurable from milliseconds) to neighboring bridges. The bridge having the lowest Bridge ID (composed by the 2 byte configured priority plus the 6-byte MAC of an address of the bridge) is elected as root bridge of the main spanning tree. Every bridge receiving BPDU from the root bridge accepts it as its root bridge and propagates the root bridge identifier in the root bridge field of the BPDUs emitted (Fig. 12). For backward compatibility, the AMSTP protocol BPDUs use the same local multicast protocol addresses that the spanning tree protocol (Bridge Group Address 01-80-C2-00-00-00). These addresses are neither forwarded by bridges nor by ABridges.

These BPDUs emitted contain the minimum path cost from the emitting bridge to the elected root bridge. Each bridge builds its own BPDU with the

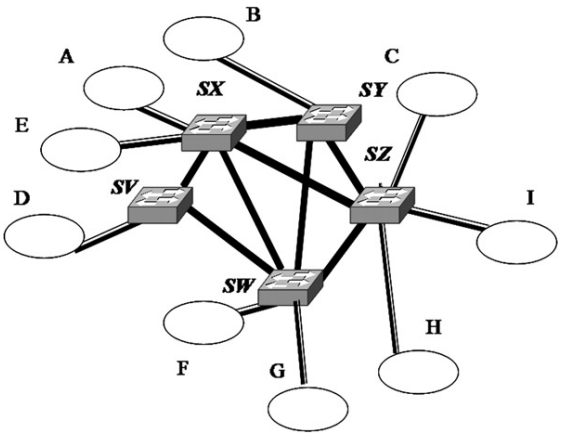

Fig. 10. Original five-node core network.

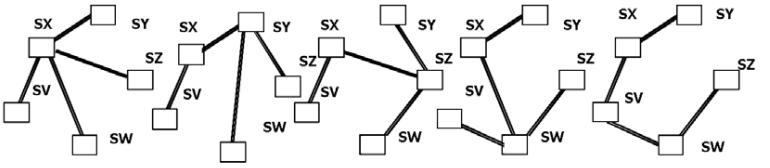

Fig. 11. The five spanning tree instances for the Fig. 10 network.

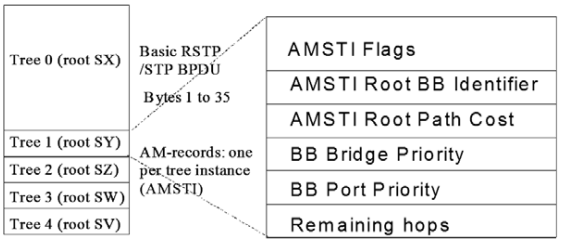

Fig. 12. AMSTP BPDU layout.

result of received BPDUs from other bridges, selecting "superior" BPDUs according to the standard STP criteria (lower Bridge ID, lower path cost, lower port priority, lower port ID), and emits BPDUs with this information to neighbor bridges for the continuous maintenance of the optimum main spanning tree. Every bridge attaches to each spanning tree instance by selecting as root port the port that is receiving the "best" BPDU. The "best" BPDU is the one that announces minimum path cost to root bridge.

AMSTP BPDUs have a structure that resembles MSTP BPDUs [4] since both are comprised essentially of a basic BPDU and several AM-Records, as it is shown in Fig. 12. The basic BPDU is used for the negotiation of the basic tree (0). Each of the AM-Records contains the data used to negotiate a specific tree instance (AMSTI). Every ABridge, with the exception of the elected root bridge, creates an AM-Record for its own spanning tree instance.

Every AM-record includes an octet flag identical to the one used by RSTP, shown in Fig. 1. These flags contain port role and state information, and the Topology Change Notification flags. They are used by connected ports of neighboring switches to negotiate the transitions of each tree instance with a proposal/agreement mechanism.

The process of building the rest of the tree instances, one per ABridge being root of an access network, takes place as follows: Each ABridge appends to the main spanning tree BPDU the information of all ABridge tree instances that he has notice from the BPDUs received, adding up its link costs, bridge priority, bridge identity and flags. This information, included in the AM-record is similar to the case of basic tree. Each tree instance is identified solely by the ABridge ID, information that is present both at AM-record and at encapsulated frames as destination address. One of the key differences with other spanning tree protocols is that in this case there is no root bridge election phase. In AMSTP the ABridge builds an AM-record for its own tree instance and accepts equally every other ABridge claim as root bridge of its own instance by processing the AM-records of other instances, each one originated by one ABridge, received. The bridge is accepted as the root by other bridges without negotiation. This self-rooted tree instance is identified by the bridge ID of the edge ABridge (root). The rest of the process is analogous to the building of the MSTI tree instances used by MSTP inside an MST region [4]: the tree is built by selecting tree paths at every bridge according to the same minimum path cost criteria that MSTP has, i.e. using port priority and port ID for tie breaking. A flag octet, identical to the one for building the basic tree instance, is used by the bridges to communicate and negotiate transitions of port states and roles per tree instance.

Once built, the tree instance rooted in a given ABridge is used to forward received frames toward that ABridge. The Forwarding Database (FDB) is created with one tuple per destination ABridge, This tuple contains as output interface to next hop the identity of port selected as root port for the spanning tree instance that is rooted at destination ABridge

5. Managing ABridges campus networks

Minimum or even zero configuration is an important requirement for campus networks, since it saves operating costs and minimizes network unavailabil-

ity caused by configuration errors. ABridges require, by design, minimum configuration. We now describe the ABridges configuration mechanisms and VLANs usage.

Regarding to IP configuration, the ABridges campus network constitutes a single IP segment, so the IP addresses of end nodes do not change when the point of attachment changes. Additionally, no IP configuration in the campus network is required, and in particular, IP addresses do not require segments to be planned and maintained when they grow.

ABridges do not require priority configuration for core operation. The core will operate correctly regardless the particular ABridge elected as root bridge. However, it is recommended practice to improve maintainability in bridged networks to explicitly configure the preferred root bridge and reserve root bridge of main spanning tree instance, making the network predictable upon failures. To do that it suffices to configure the target root with a low enough priority, and a slightly higher one for the root reserve bridges.

In relation to access layer priorities, ABridges connected to the core are configured to use a higher default priority than the one used by standard bridges at the access layer. As a consequence, the ABridge with lower bridge ID will be elected as RSTP root. In case of malfunction of ABridge core section, the priority may automatically descend to a low priority value to prevent being selected as root bridge of the access network. In this way, another ABridge with proper connection to the core, which was not acting as root for the access layer, could be elected as root in case of failure of the former root ABridge.

Another important self-configuration feature is that the ABridges core is formed automatically. The ports of ABridges that are not connected directly to another ABridge do not run the AMSTP protocol, falling back to the RSTP protocol and being kept out of the core forwarding mechanism. In this way, these ports auto-configure as access ports to interoperate with legacy switches running STP or RSTP. Port auto-configuration works as follows: each port detects, through the STP BPDU type (STP, RSTP or AMSTP) received on their link upon initialization, whether the device connected to the link is a standard bridge or an ABridge. If the BPDUs received are standard 802.1D BPDUs, the link will be assigned to the Access Network and the port will be automatically configured to access port mode. Any standard bridge connected to the ABridge is thus automatically excluded from the core function.

It is also worth to note that the most important auto configuration feature of ABridges is that VLANs and tree instances configuration are no longer required to achieve effective network infrastructure usage through multiple spanning trees, which is the case for MSTP operation. A self-rooted instance is automatically created by the AMSTP protocol per each ABridge, without requiring any configuration.

However, VLANs are used sometimes to separate traffic for many reasons, such as security, better network management, etc. Note that at the access layer, that normally has a tree-like structure, the use of VLANs does not improve significantly the utilization of the network infrastructure. VLANs at the access networks, including the access ports of ABridges that connect them to the core, operate in conformance to the IEEE 802.1Q standard. Access ports of ABridges may belong to VLANs. When VLANs are used in the access networks, standard bridges and access ports of ABridges need to be configured accordingly, specifying to what VLAN their ports belong to as in any regular VLAN network; alternatively, a dynamic VLAN server may handle VLAN assignment to hosts according to a stored VLAN-to-hosts list. ABridges may learn, as VLAN aware bridges, which port belongs to which VLAN by inspecting the incoming VLAN tagged frames. This may simplify VLAN configuration in ABridges but does not eliminate the need to configure VLANs in campus networks: Tagged VLAN frames must be generated either by manually configured bridges or by hosts originating the frames. If the hosts initially set the VLAN tag, a system to assign a VLAN to each host must be set up via a dynamic VLAN server, which also requires configuration.

The default operation of ABridges in the core is to operate core ports as VLAN trunk links, (that is, core links may transport frames of several VLANs) and tag the frames with the explicit VLAN tag corresponding to the access port where the frame entered. The VLAN tag is appended to the received frame according to the 802.1Q standard, and the frame is encapsulated with the additional link layer header. Note that the VLAN tag is not used to perform core forwarding. VLAN tags are recuperated at the Egress Bridge when the outer header is removed, and standard VLAN delivery to the appropriate ports is performed.

The difference with a core with the standard VLAN processing with MSTP is that with MSTP the broadcast of frames is performed only via the ports belonging to the VLAN whilst with AMSTP and servers, the unresolved addresses are broadcasted through the whole core. However, ABridges make limited use of broadcasts, since this mechanism is restricted to failures in address resolution mechanism of the ARP/ABridge servers or the use of broadcast by specific services.

6. Comparative analysis and evaluation of AMSTP

It is complex to evaluate and compare campus networks architectures and protocols due to the diversity of requirements and the relative importance assigned to them. Besides, campus networks normally combine several protocols, at least the standard bridge protocols STP and/or RSTP. The effective performance is a result of the combined action of all protocols, and not only core protocols, although these ones are the most critical components due to the higher performance involved.

The comparison of ABridges and AMSTP with alternative protocols, currently under discussion at IEEE and IETF provides a clear positioning of our proposal in the framework of current standardization work. However, it must be taken into account the difference in emphasis among the requirements addressed by the proposals. In our case, it is worth to remember that ABridges/AMSTP gives priority to the requirements of self-configuration and performance.

Taking this into account we first perform a qualitative evaluation and comparison of the AMSTP based architectures with alternative protocols RSTP, MSTP, RBridges and Shortest Path Bridging. This evaluation analyzes among other, the following criteria: self-configurability, scalability, infrastructure utilization, storage requirements and security. A quantitative comparison is performed between AMSTP and RSTP in the connectivity degree of the resulting topology.

6.1. Summary of alternative protocols

There are two standards and two draft proposals related with our work: The standards are IEEE 802.1D (RSTP) and IEEE 802.1Q (MSTP), and the drafts are RBridges (TRILL Working Group) at the IETF and Shortest Path Bridges at IEEE (802.1aq). RSTP has already being described.

6.1.1. RBridges

The problem considered in our paper is under discussion at the IETF and the IEEE with diverging approaches. At the IETF, so called RBridges [1,8] are proposed as a hybrid of routers and bridges, keeping the advantages of routers-like shortest paths and scalability while preserving at the same time the zero IP configuration capability of bridges. RBridges currently do not explicitly aim or are required to scale to large Ethernet campus networks. RBridges exchange between them the list of hosts they are responsible for. When a frame is received from the 802.1D network the RBridge is responsible for, the RBridge looks at the host table made aggregating all the host lists received, to find the destination RBridge, and looks up in the RBridge routing table to find the next hop RBridge towards a given one. The originating RBridge address and the next hop RBridge address are inserted in an encapsulation header added to the standard 802.1D frame received. RBridges use a modified version of the IS-IS routing protocol to propagate the MAC addresses of RBridges. The RBridge packet format includes also a TTL field to discard packets trapped in transient routing loops.

6.1.2. Shortest path bridges (SPB)

Shortest Path Bridging (SPB), is a recent proposal under discussion at the IEEE [802.1aq] [9]. SPB is aimed to operate in a Shortest Path Tree (SPT) Bridging Region and interoperate with RSTP and MSTP through configuration managed through management interfaces. In a SPT Region multiple tree instance are created, rooted at respective SPT capable Bridges. Each tree is linked to a specific VLAN. Learning of host MAC is supported per VLAN but sharing the MAC learning at opposite tree instances at each link.

The current draft (0.3) lists different alternative solutions regarding to the protocols to be used. Like AMSTP, SPB uses multiple tree instances rooted at the edge bridges to obtain shortest paths between bridges. The proposal aims for compatibility with VLANs and 802.1Q. Accordingly, a Shortest Path Region corresponds to a Multiple Spanning Tree region of the 802.1Q standard. As in MSTP, multiple regions may exist and are differentiated by a per region configuration identifier. Assuming two bridges in a region with one tree instance rooted at each bridge, the path between them on the two instances may coincide or not. If they coincide, it

is a symmetrical path. Note that MAC learning requires the path to be symmetrical.

SPB uses shared VLAN learning (SVL) of MAC addresses among VLANs for frames allocated in different spanning tree instances. The shared learning uses a common filtering information database (FID) for learning of MAC addresses by the two VLANs associated respectively each one to the two tree instances of a path in the campus network core. A restriction imposed by the MAC learning of SPB is the requirement for symmetrical spanning trees, i.e. the same path must be enforced for connecting two bridges in both possible directions, in order to properly perform address learning. The ties in the path costs of tree instances could be resolved with local information that may result in the selection of different paths. The *cut vector* mechanism is added to avoid the occurrence of asymmetrical paths, but it results in a degraded performance for the convergence protocol, because each link election during the tree instance generation must be notified across all the nodes previously added.

In the current draft (D0.3), three choices are contemplated for the computation of the set of symmetric shortest path trees between each of the bridges of an SPB region: A derivation of MSTP protocol (distance vector based), with addition of cut-bit vectors to ensure symmetry of tree instances; use of a extended IS–IS protocol with additional information and procedures and finally, the use of a new Link State Tree Protocol (LSTP).

6.1.3. Multiple spanning tree (MSTP)

MSTP has been described above and is the basis of AMSTP protocol. The ABridges architecture might be compared with a VLAN plus MSTP-based core. However, MSTP is difficult to compare with AMSTP in numerical terms because performance depends heavily on tree instance design. MSTP achieves shortest paths only when the tree instances are carefully designed and configured to do so, one per edge bridge, AMSTP always obtains shortest paths with self-configuration.

6.2. Configuration evaluation

We now review the amount of configuration required by the different protocols.

RSTP does not require IP-related configuration, and only requires VLAN configuration when this mechanism is specifically required. Root bridge priority configuration is optional, but recommended.

MSTP is complex to configure. Tree instances must be carefully planned and VLANs must be mapped manually to those tree instances. The configuration table must be exactly the same for all the bridges of the same region, or serious malfunction may occur. Erroneous definition of the tree instances associated to VLANs may cause network partitions. MSTP configuration requires region definition and delimitation, and explicit mapping of VLANs to tree instances (MSTIs) at each region defined, which has to be carefully planned and then configured at each bridge, a complex and error-prone process.

RBridges do not require IP configuration but require detailed VLAN configuration at the RBridges "core", which leads to potential multiplication of the routing "trees" at every RBridge.

Shortest Path Bridges do not require IP configuration, but require detailed VLAN configuration and tree instances mapping to VLANs to keep compatibility with MSTP. SPB also requires detailed configuration of VLAN IDs to be used for the shortest path tree instances. Complexity of configuration equals roughly to that of MSTP plus the added SPB functionality. SPB presents the advantage of not modifying the data plane (no encapsulation of data in core).

Abridges require minimum configuration, as described above, only slightly higher than RSTP, much simpler than MSTP, RBridges and SPB. Minimum configuration is, by design, an important advantage of ABridges and AMSTP protocol.

6.3. Scalability evaluation

The MSTP protocol is limited by the standard to 64 tree instances maximum per region and per BPDU. Note that if shortest paths are the objective, one tree instance per edge node is mandatory. The number of nodes is not limited.

RBridges do not aim to scale beyond current campus network sizes. They have limited scalability regarding the number of hosts and of bridges in the network. The number of hosts is limited by the use of flat MAC addresses for routing. Global host lists are needed at RBridges to perform the routing, and as a consequence, long host lists must be multicasted periodically among RBridges, and stored on them. The number of RBridges must be bounded according to link state protocol limitations to control complexity and overhead.

The RBridge packet format requires processing at each RBridge hop in order to set explicitly in

the destination field the link layer address of the next hop RBridge and decrement TTL.

The current draft for SPB suggests possible restrictions in the number of trees and link cost information to prevent excessive complexity and protocol overhead. This is reinforced by the increased complexity of the available link state protocols (i.e IS–IS and LSTP). When SPB uses a variation of MSTP,the maximum number of nodes is limited to 32 due to the increased info transmitted on BPDUs.

AMSTP has the same limits than MSTP. However, if AMSTP is implemented as N fully independent RSTP tree instances, then AMSTP has no BPDU length restrictions, since each BPDU carries info of only one tree instance.

6.4. Storage needs evaluation

We compare here the storage needs of the above mentioned protocols. ABridges use both AMSTP and RSTP protocols.

The MSTP storage needs are a caché memory per port is needed to learn hosts addresses separately per VLAN (independent VLAN learning), a table to list the VLAN IDs associated to each multiple spanning tree instance, and a forwarding table per tree instance.

RBridges must store tables at every RBridge the complete host list and designated RBridge that corresponds to each host. This requirement may be overkill in big campus networks with tens of thousands of hosts. Additionally, the use of a link state protocol imposes the requirement of maintaining the full topology at each RBridge.

SPB storage needs are those of MSTP plus the storage needs derived of the link state protocol used by SPB bridges. This means that the full topology must be known by SPB bridges.

Storage needs of AMSTP are very similar to MSTP. The differences are the following: ABridges do not use address learning in core ports. They do not need port cache in core ports. ABridges use a forwarding table based on the root ports of the bridge for the respective tree instances. The forwarding table at ABridges contains one entry per egress Abridge ID, containing the root port of the tree instance as output interface. Its size is then $O(N)$ because ABridges always create N tree instances, one per ABridge node, while MSTP create as many as configured, with a maximum of one instance per VLAN and a standard limit of 64.

At access ports, ABridges use the standard MAC addresses learning mechanism cache. The maximum number of MAC addresses to be learnt per access port is the number of nodes on the ABridge's access network, about several thousands for a very big campus network. ABridges also optionally store in a table the addresses of announced ABridge servers, like those in charge of the ARP host resolution/registration service and the hash values supported by each server, whose number is not expected to be high. The optional ARP servers use a table with destination host MAC, IP address, and associated ABridge covering the hash results supported by that ARP server.

Finally, the RSTP protocol uses for state negotiation and routing the so called port priority vectors. These vectors contain: the ID of the root bridge, root path cost, designated bridge ID, designated port ID and receiver port ID. The storage need per bridge is proportional to the number of ports $O(P)$, but independent of the number of network nodes M. This characteristic is typical of spanning trees. The ABridge ports connected to islands operate according to the standard IEEE 802.1D. ABridge ports learn the MAC addresses (SA of frame) of island hosts from local and egressing traffic. Egress traffic exiting to the island, being encapsulated, is not learnt by ABridge access ports, although it might be used to refresh short term caches of ARP server responses.

6.5. Complexity and processing evaluation of protocol message processing

The message complexity of AMSTP messages is similar to the MSTP case when the number of tree instances at MSTP equals the number of core nodes. Length of message is $O(AB)$ where AB is the number of edge ABridges (transit ABridges do not need a tree instance). The frequency of messages sent is the same as RSTP, being more frequent in case of reconfiguration. In RSTP, MSTP and AMSTP, the number of emitted BPDUs per second is limited by the TransmitHoldCount parameter TxHoldCount [3].

RBridges exhibit the bigger complexity of route calculation of link state protocols $O(AB^2)$, although the maximum number of RBridges is limited. The same occurs for Shortest Path Bridging when a link state protocol is used.

6.6. Convergence speed evaluation

When there is a bridge or link failure, reconfiguration takes place. In this section we describe the

factors affecting convergence speed of the protocols considered. In the ABridges proposal, convergence speed at core depends on AMSTP performance and at islands depends on RSTP speed.

MSTP convergence speed is similar to RSTP speed because the fast reconfiguration mechanisms used are the same. However additional mechanisms are used by MSTP at region level during tree instances reconfiguration to maintain consistency.

Convergence speed for RBridges in case of reconfiguration depends on the messages update period of the extended IS-IS routing protocol used. If subsecond reconfiguration time is the target, updates must be very frequent and this would increase the protocol processing load and overhead.

SPB convergence speed will be determined by the protocol used to build and reconfigure the spanning tree instances. Although it is currently optional, it is likely that an IS–IS like protocol is selected to create the spanning tree instances, so the considerations would be similar to the RBridge case.

Although not detailed evaluation of AMSTP convergence has been performed, it can be shown that it is equal or better than MSTP. This is because AMSTP is functionally a significant simplification of MSTP. AMSTP does not need to elect a root ABridge per core spanning tree instance (with the exception of the main spanning tree instance that is subject to root bridge election), and reconfiguration of a tree instance due to root bridge failure does not happen. Another key difference is that AMSTP operates in a single MST region so it does not require synchronization mechanisms at region level during transitions. Typical RSTP reconfiguration speeds are in the order of tens of milliseconds and less than 2 s, although there is a count-to-infinity situation identified [18] that may extend to several seconds the convergence process.

6.7. Evaluation of link utilization

RBridges can use all links available because the link state nature of the routing protocol. The standard bridged networks attached to them will likely use standard spanning tree protocols, so they are subject to the link blocking behavior of 802.1D spanning tree protocols.

MSTP allows, as AMSTP, full link usage inside the region, but needs careful planning and configuration of tree instances.

RSTP blocks all links that may create forwarding loops in the network, so the maximum number of links enabled is $N - 1$. RSTP has very poor utilization in networks with medium or high interconnection degree, although it has very high utilization in tree like network topologies. This is the case of the islands networks in the proposed architecture.

The degree of link utilization in the proposed architecture is a combination of link utilization at core and at access networks. At campus core, network infrastructure is used efficiently through multiple spanning tree instances.

In a core running the standard single spanning tree protocol, the maximum number of active links in the core is $N - 1$, while in a core running Alternative Multiple Spanning Tree Protocol the number of active links is only limited by the maximum of $N(N - 1)/2$ links of fully connected networks. The superposition of spanning tree instances per core node allows the utilization of all links. At each core node, the traffic is distributed among several links according to the destination. For an equivalent carried traffic, lower dimensioning of core links is possible as a consequence of this traffic distribution. The link bandwidth can be divided up to a factor of $N/2$ in case of a full connectivity topology. Fig. 13 shows the comparison on number of links enabled between the network with maximum number of links (a fully connected network with all links used) using AMSTP and the same network with single spanning tree protocol. Each real network running AMSTP will get a number of core links enabled somewhere in between the two lines, depending on its connectivity. The benefit increases with the degree of connectivity of the network.

6.8. Path length evaluation

We now compare the path costs of ABridges architecture with other protocols. In the ABridges

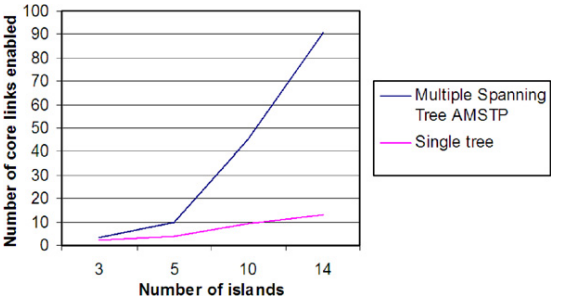

Fig. 13. Network infrastructure utilization range (max.: AMSTP full connectivity/min.: single tree).

architecture, total path length is the sum of the path at the core and the path at the access networks or islands. For comparison, we assume homogeneous link speeds at core and use hop count as a measure of path length. Since the protocols considered in this evaluation differ at the core forwarding, path length at island networks does not likely differ among them. We compare path lengths only at core.

6.8.1. Path length at core

AMSTP is a shortest path protocol, therefore the path lengths in the core are minimum. The same happens with SPB and RBridges, but not with MSTP, that requires careful planning of tree instances and configuration to obtain shortest paths. If RSTP is used in the core instead of AMSTP, path length increases significantly.

We choose the regular and strongly interconnected architecture of hypercube topologies to compare AMSTP with RSTP path length and other parameters. The reasons are explained below.

One might consider for comparison purposes that the topology of campus networks may be random, at least in theory. However, one important requirement for real campus network is *predictability*: this means that the network behavior, in case of reconfiguration due to link or node failure, is known in advance and that network performance stays above known and controlled limits. It also means that worst case path length and delay must be limited. Predictability is not possible with random topologies because the connectivity is random and absolute worst case path length equals $N-1$ hops, useless in practice.

Open mesh topologies are likely the most economical topology for metropolitan networks because the cost of connecting an additional node is minimized in terms of optical fiber interconnections between distant nodes. However, when applied to campus networks the additional cost of higher connectivity is low (link lengths of km. instead of tens of km.). When a high degree of connectivity is feasible, as in the core tier of local networks or in specific cases of metropolitan networks), highly connected topologies are advisable. Then we choose to perform a comparison of path lengths between AMSTP and RSTP in core by comparing them in high connectivity topologies like *k-ary n-cube* networks.

We analytically obtain average and maximum path lengths obtained with AMSTP and RSTP for n-ary 2-cubes topologies of 8, 16 and 32 nodes. The performance of these topologies is shown in Table 1.

A *k-ary n-cube* network (Fig. 14) has n dimensions with k nodes in each dimension. A node is identified by its position in each dimension, represented by a vector $(x_1, x_2, \ldots, x_n)$. Two nodes $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$ are neighbors if there exist an i such that $x_i = (y_i + 1) \bmod k$ and $x_j = y_j$ for all $i \neq j$.

We have evaluated the performance of 2-ary n-cube topologies.

Table 1 shows the results for the high connectivity 2-ary n-cube topologies.

The average path length is 1.40 times longer with RSTP than with AMSTP for 8 and 16 nodes topologies and 2.06 times longer for 32 node. Maximum path lengths are also longer, with a maximum of 1.40 times for 32 nodes.

6.9. Saturation traffic AMSTP vs single spanning tree

Table 1 also shows the computed relative traffic carrying capacities with RSTP and AMSTP. We assume a 10 Gbps input link to core per ABridge carrying 8 Gbps load as a reference traffic of $\lambda = 1$. Using cross-sectional bandwidth, the saturation traffic is calculated analytically for all topologies. For RSTP, the saturation traffic is reduced when the number of nodes increases, while with AMSTP the increment with the number of nodes is slow, due to the full utilization of the increasing connectivity of the n-cube topology. This means scalability of the core is possible at the cost of additional links. It

Table 1
2-Ary n-cube core features

2-ary n-cube	Average path length		Max. path length		Max. Relat. offered traffic		Link utilization (%)
	RSTP	AMSTP	RSTP	AMSTP	RSTP	AMSTP	RSTP
8 node	2.39	1.71	4	3	0.46	1.10	58.00
16 node	3.22	2.3	5	4	0.36	1.09	46.00
32 node	5.33	2.58	7	5	0.23	1.21	38.00

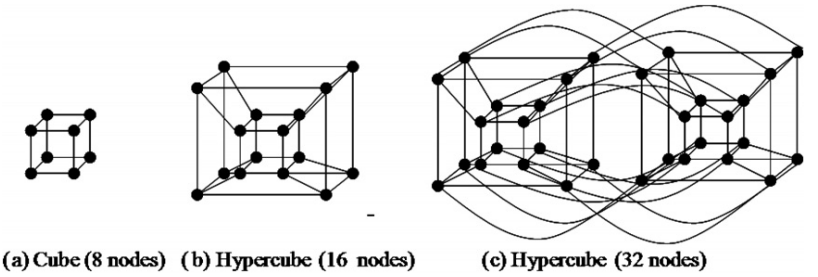

Fig. 14. High connectivity topologies (n-ary 2-cube).

is worth to note that network throughput under random traffic is always lower than the maximum theoretical cross-sectional bandwidth because the head-of-line blocking effect. We use cross-sectional bandwidth for comparison traffic simulations because saturation depends on the first link saturating, which respectively depends on the traffic distribution. Saturation of links appears first near the server location due to the low connectivity [16].

6.10. Evaluation of the infrastructure utilization

Table 1 also shows the percentage of links active of the network with RSTP. AMSTP is not shown because, with equal cost links, it achieves 100% link utilization. The links between neighbour nodes that are blocked by RSTP protocol to prevent loops are active with AMSTP protocol at instances rooted at respective nodes.

6.11. Security evaluation

This paper does not address specifically security issues, but we provide a perspective with some security considerations below. The main vulnerabilities identified at layer two are: MAC spoofing, saturation of bridges cache, attacks to the spanning tree protocols and ARP attacks. Note that the risk faced by bridged networks are aggravated in comparison to other type of networks, due to several reasons: the intruder detection systems sometimes do not monitor layer two attacks; the increase of size of layer two network and finally, provision of layer two access with a change of the meaning of the concept of *internal* attack for the ISPs.

As it is currently stated for RBridges and SPB, the security objective of ABridges is to keep at least the same security level of bridged networks, without introducing additional risks.

The key topological position of ABridges and their role as Root Bridges combined with the use of ARP Servers/Registrars enable the implementation of enhanced security measures with easy localization of attackers, fast detection of spoofed MACs by authentication between ABridges, etc. If IEEE 802.1X is used in link ports connecting ABridges, security is greatly enhanced in the network core, although it can not prevent malicious behavior of trusted authenticated ABridges. Authentication, however, requires some additional configuration, which opposes in part to the zero configuration objective of ABridges.

The security of ABridges architecture is similar or better than MSTP and RBridges by limiting broadcasts and allowing enhanced network control and traceability of attackers and malicious nodes.

7. Related work

In this section, we summarily present proposals related with Abridges aside from those already discussed in previous sections (i.e. RSTP, MSTP, RBridges and SPB).

Viking [16] is a per-VLAN (PVST) spanning tree architecture oriented to proprietary and Storage Area Networks. The Viking Manager calculates optimum routes between hosts, and alternative routes to be used in case of failure. These routes are mapped to VLANs. The bridges use the MSTP protocol and are configured accordingly to theses VLANs by the Viking Manager via SNMP. The hosts must run specific software to select the assigned VLAN. Viking may optimize load balance in network, but requires a complex management plane and requires the cooperation of the hosts.

The Scalable Spanning Tree (SST) [10] is a proposal for MSTP multi-region networks, oriented to arrange and adjust automatically the regions. Configuration is performed through the SNMP protocol. Its application scenario exceeds by far the campus area network domain.

Link State Over MAC (LSOM) [7] is a proposal for Metropolitan Ethernet backbones, but its link state routing based on host MAC addresses does not scale to big campus networks.

Global Open Ethernet (GOE, [14] is proposed by NEC to deploy hierarchical Layer-two Virtual Private Networks (L2VPN) as a replacement to Ethernet over MPLS. Ethernet over MPLS is considered too expensive due to the need for either IP routers, or for Q-in-Q encapsulation (802.1ad), which uses hierarchical stacked VLAN tags to forward packets on the provider network. GOE is based on a multiple tree forwarding topology. Each bridge is assigned a VLAN-ID and the VLAN tag is used as a routing address. GOE is compatible with the MSTP protocol, at the price of a complex configuration to align VLAN mapping in MSTP bridges with GOE bridges.

The main idea behind the Thin Control Plane proposal [15] is to abandon the Ethernet broadcast model. In this architecture, separate route calculation (decision plane) is performed at some servers, which spread the routes to the switches (dissemination plane). As drawbacks, we note that compatibility with Ethernet bridges is not considered, hosts have to be modified, and big campus networks may suffer from high bandwidth consumption due to route diffusion.

The Universal Ethernet Telecommunication Service (UETS) [17] is a recent proposal for high performance hierarchically addressed Ethernets through protocol stack simplification. Although UETS may be applied at all network usage scenarios, including campus networks, UETS is optimally suited to ISPs, metro Ethernet networks and high performance storage networks. Application of UETS principles to campus networks requires further work on zero configuration mechanisms and interoperability between UETS switches and standard bridges. The hierarchical scheme for assigning Ethernet addresses locally to the network, proposed by UETS, however, seems applicable to the ABridge campus networks architecture, but requires further study. This addressing scheme scales better than the currently predominant "MAC in MAC" encapsulation, first proposed in [13].

8. Conclusions

We have presented the ABridges architecture, a two-layer network architecture based on network islands running independently the simple and efficient RSTP protocol, that communicate through a core that uses the new AMSTP protocol to interconnect the root bridges (ABridges) of the islands. This architecture is very efficient in terms of network infrastructure usage due to the ability of AMSTP to provide optimum paths in the core mesh, while RSTP efficiently aggregates the traffic at islands networks, where low connectivity, tree-like topologies are frequent and recommended. Additionally, the architecture is also efficient in terms of failure recovery, since fast convergence of RSTP/MSTP protocols is preserved in the core using AMSTP.

Compared to existing core protocols, AMSTP equals to MSTP protocol when optimally configured for shortest paths, without the complexity and consequent unreliability of manual configuration. Compared with RBridges and Shortest Path Bridges, it provides scalability to bigger campus networks and faster convergence. The architecture provides shortest or close to shortest paths in most topologies and adapts well to traffic aggregation in switches. Regarding deployment strategies in existing networks, standard switches can be upgraded to ABridges via software migration.

In terms of configuration requirements, ABridges do not require any, while other architectures based on MSTP require VLAN and tree instance configuration in core, and VLAN configuration at ports of access networks. On one hand, the AMSTP protocol provides shortest paths among the bridges with zero configuration. On the other hand, the layered architecture allows the deployment of large networks with a single IP segment while maintaining segmentation in case of failure.

Interoperability with standard bridges and transparency to hosts and routers eases deployment significantly. A standard bridge connected inside the ABridges core is automatically excluded from core. An ABridge that gets disconnected from other ABridges, then unable to act as gateway to core, self-configures all ports as standard bridge ports and the edge function is taken by another ABridge. The continuity requirement in the core makes sense as any standard bridge interposed in the core would compromise any high performance guarantee.

When the architecture is compared to other architectures based on link state protocol, the distance vector nature of both AMTSP and RSTP clearly results in lower complexity and overhead. The proposal has similar computational complexity than Shortest Path Bridges,and lower than link state based RBridges. Computational complexity is of

same order than N spanning tree protocols. The architecture is applicable to arbitrary network topologies.

Acknowledgements

This work was supported in part by grants from Spanish Ministerio de Educación through Project CAPITAL (TEC2004-05622-C04-03/TCM) and from Comunidad de Madrid through Project E-MAGERIT (S-0505/TIC/000251). Special thanks to José Félix Kukielka who reviewed the manuscript.

References

[1] The rbridge archives. <http://www.postel.org/pipermail/rbridge/>.
[2] Rapid Reconfiguration of Spanning Tree. <http://www.ieee802.org/1/pages/802.1w.html>.
[3] IEEE 802.1D-2004 IEEE standard for local and metropolitan area networks – Common specifications – Media access control (MAC) Bridges.
[4] IEEE 802.1Q-2003 IEEE standard for Local and Metropolitan Area Networks – Virtual Bridged Local Area Networks.
[5] IEEE 802.1D.IEEE-1998 IEEE standard for local and metropolitan area networks – Common specifications – Media access control (MAC) Bridges.
[6] G. Ibáñez, A. García, A. Azcorra, Alternative multiple spanning tree protocol (AMSTP) for optical ethernet backbones, IEEE HSLN (LCN 2004) Tampa (November) (2004).
[7] R. García, J. Duato, F. Silla, LSOM: a link state protocol over MAC addresses for metropolitan backbones using optical ethernet switches, in: Proceedings of Second IEEE NCA'03.
[8] R. Perlman, J. Touch, A. Yegin, RBridges: transparent routing draft-perlman-rbridge-00.txt April 2004. <http://www.ietf.org/internet-drafts/draft-perlman-rbridge-00.txt>.
[9] Shortest Path Bridging. <http://www.ieee802.org/1/pages/802.1aq.html>, Draft 0.3, May 2006.
[10] K. Ishizu et al., APG-Report: SSTP: An 802.1s extension to support scalable spanning tree for mobile metropolitan area network, in: Proceedings of Globecom 2004, December 2004.
[11] Gigabit Campus networks Design. WhitePaper, Cisco Systems, 1999/2003. <http://www.cisco.com>.
[12] Cisco, Campus Network Design. <http://www.cisco.com/warp/public/779/largeent/design/campus_index.html>.
[13] I. Hadzic, Hierarchical MAC address space in public ethernet networks, IEEE Globecom (2001) 1563–1569.
[14] Iwata et al., Global open ethernet architecture for a cost-effective scalable VPN solution, IEICE Trans. Commun. E87-B (1) (2004) 142–151.
[15] Myers et al., Rethinking the Service Model: Scaling Ethernet to Million Nodes, HOTNETS III, November 2004.
[16] S. Sharma et al., Viking: a multi-spanning-tree ethernet architecture for metropolitan area and cluster networks, IEEE INFOCOM (2004).
[17] J. Morales, G. Ibáñez, Ethernet fabric routing (EFR): a scalable and secure ultrahigh speed switching architecture, in: High Speed Networking Workshop TCHSN INFOCOM 2006, Barcelone, April 2006. <www.ieee.org/ieee.explore>.
[18] K. Elmeleegy, A.L. Cox, T.S.E. Ng, On count-to-infinity induced forwarding loops in ethernet networks, in: INFOCOM 2006, Barcelone, April 2006.

Guillermo Ibáñez received his Telecommunication Engineering degree from Universidad Politécnica de Madrid in 1975 and the Ph.D. degree in Communication Technologies from Universidad Carlos III de Madrid in 2005. He worked at ITT R&D Laboratories in Madrid till 1984. He has long R&D experience in the telecommunication industry at Alcatel Spain, where he has held several technical and technical leading positions in the international development of Alcatel System 12 Switching System and Litespan 1540 Multiservice Access Nodes till 2002. He is an Associate Professor in the Telematics Engineering Area of the Universidad de Alcalá in Madrid. His current research interests are high performance and scalable Ethernet networks, wired and wireless. He is author of several publications on these subjects.

Alberto García-Martínez received his Ph.D. in Telematics Engineering from Universidad Politécnica de Madrid (UPM) in 1999. He is an Associate Professor in the Telematics Engineering Department of the Universidad Carlos III de Madrid, Spain He has participated in several national and international research projects on IPv6 and QoS. His recent research interests are related with multihoming in IPv6, protocol security and routing and addressing architectures.

Arturo Azcorra received a M.Sc. degree in Telecommunications from Technical University of Madrid (Spain), in 1986, and Ph.D. degree from the same university in 1989. In 1993 he obtained an MBA from Instituto de Empresa, Madrid. He was a lecturer at Technical University of Madrid from 1987 to 1990 and promoted to associate professor. In 1998 he joined U. Carlos III of Madrid (Spain), where he is now a full professor. Professor Azcorra has participated and directed over 20 European research and technological development projects from ESPRIT, RACE, ACTS and IST programs. He has also performed direct consulting and engineering work for institutions such as European Space Agency, MFS Worldcom, Madrid Regional Government, RENFE, REPSOL, and the Spanish Ministry of Science and Technology. He has been program committee member of international conferences in several editions of IEEE-PROMS, IDMS, QofIS, CONEXT and IEEE-INFOCOM. He is general co-chair for CONEXT'05, and TPC

co-chair for INFOCOM'06. His list of papers published in national and international magazines, books and conferences is over 100 titles. Current research projects include broadband networks, multicast teleservices, active networks, and advanced IP networks.

Ignacio Soto received a Telecommunication Engineering degree in 1993, and a Ph.D. in Telecommunications in 2000, both from the University of Vigo, Spain. He was a research and teaching assistant in Telematics Engineering at University of Valladolid since 1993–1999. In 1999 he joined University Carlos III de Madrid, where he has been an associate professor since 2001. His research activities focus on mobility support in packet networks and heterogeneous wireless access networks. He has been involved in international and national research projects related with these topics, including the EU IST Moby Dick and the EU IST Daidalos projects. He has published several papers in technical books, magazines and conferences, lately in the areas of efficient handover support in IP networks with wireless access, network mobility support, and security in mobility solutions. He has served as Technical Program Committee member of INFOCOM.