

# Bridges

Zartash Afzal Uzmi

Tariq Mahmood Jadoon

Computer Science and Engineering

Lahore University of Management Sciences, Pakistan

Email: {zartash,jadoon}@lums.edu.pk

November 21, 2006

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Transparent Bridges</b>	<b>6</b>
<b>3</b>	<b>Bridge Operation</b>	<b>8</b>
3.1	Learning and Forwarding Operation . . . . .	9
3.2	Loop Free Operation . . . . .	9
<b>4</b>	<b>Spanning Tree Algorithm (STA)</b>	<b>10</b>
4.1	Description . . . . .	11
4.2	Protocol Operation . . . . .	11
4.3	Protocol Algorithm . . . . .	13
4.4	Address Learning and Frame Forwarding . . . . .	16
4.5	STP Example . . . . .	16
4.6	Reconfigurations and STP . . . . .	17
4.7	Reconfigurations and Loops . . . . .	19
4.8	BPDU Format . . . . .	20
<b>5</b>	<b>Bridges and LAN Multicast</b>	<b>21</b>

<b>6</b>	<b>Generic Attribute Registration Protocol (GARP)</b>	<b>23</b>
6.1	GARP Operation . . . . .	23
6.2	Virtual LANs and GVRP . . . . .	24
<b>7</b>	<b>Summary and Recent Trends</b>	<b>25</b>
<b>8</b>	<b>Glossary</b>	<b>26</b>

**Keywords:** LAN, Local Area Network, Bridge, Switch, Spanning Tree Algorithm, MAC Bridge, Extended LAN, VLAN, GARP, Expedited Forwarding, Ethernet, 802.1D, 802.3.

### Abstract

Bridges are devices used to scale a local area network (LAN) in terms of range and number of connected hosts or stations. A LAN segment is restricted to span a small geographical area and only a small number of hosts or end stations can be connected to it. However, LAN segments can be joined together, without an appreciable performance degradation, by means of a bridge. Using many bridges, a conglomerate of LAN segments, called an extended LAN, can be created that may cover a reasonably large geographical area, typically a campus, supporting a larger number of end stations. Thus, Bridges constitute an integral part of today's LANs. The term LAN is used in the literature to refer to either a LAN segment or an extended LAN.

A bridge is used to extend the range and capacity of a local area network by joining LAN segments, forming an extended LAN. The use of a hub or a repeater also extends the range and increases the capacity of a single LAN segment, but does not form an extended LAN. A single LAN segment, whether or not it uses hubs or repeaters, is fairly limited in geographical scope and typically covers a small floor. On the other hand, an extended LAN, formed by the use of bridges, may cover an entire campus.

Connecting LANs over a wide area, say two different campuses of an enterprise, generally requires the use of routers for interconnection. However, in this chapter, our discussion will be restricted to an extended LAN, formed by using bridges. We will further restrict our discussion to transparent bridges as those are the ones commonly used in practice. Bridges are transparent in the sense that two stations on two different segments of an extended LAN, formed by using transparent bridges, communicate as if they are on the same LAN segment.

Some other major functions performed by Bridges include allowing LAN traffic differentiation by assigning *priority* to certain types of LAN traffic and allowing the formation of Virtual LANs (VLANs). We will also cover the mechanisms that are used by bridges for the formation of VLANs and the role played by bridges in LAN multicast.

## 1 Introduction

A LAN segment is usually restricted to span a small geographical area and may only support a small number of hosts. Adding more hosts can cause the performance of the LAN segment to degrade. Furthermore, if the geographical span of the LAN segment is increased, end stations may no longer be able to communicate.

The end stations or hosts, as they are sometimes called, are connected to a local area network (LAN) using one of the many LAN technologies, several of which have been standardized by the 802 committee of the Institute of Electrical and Electronic Engineers (IEEE). The most commonly used LAN technology is based on the Carrier Sense Multiple Access with Collision Detect (CSMA/CD) protocol and was standardized by the IEEE 802.3 subcommittee. Another LAN standard based on CSMA/CD is Ethernet (also referred to as Ethernet II or DIX Ethernet, standardized by Digital Equipment Corporation, Intel and Xerox), and is quite similar to the one standardized by the IEEE 802.3 subcommittee. In fact, the CSMA/CD based LANs in use today are those which have been standardized by IEEE 802.3 and are backward compatible with Ethernet, and both of them are usually referred to as ‘ethernet’. Therefore, we will also use the terms 802.3 LAN and Ethernet LAN interchangeably unless it is required to highlight the differences between the two. Since CSMA/CD based LANs are the most prevalent these days, these will be our primary focus in this chapter and any reference to a LAN will implicitly mean an 802.3 or an Ethernet LAN.

The ethernet standardization included two layers of the networking stack: the physical layer and the data link layer. Since Bridges are meant to connect various LAN segments, they also implement the physical and data link layers of the networking stack. The data link layer of the networking stack is subdivided into the media access control (MAC) sublayer which depends upon the actual physical technology (e.g., ethernet or token ring), and the logical link control (LLC) sublayer which is independent of the physical technology used to implement the LAN and allows sharing of the data link resources by providing a unified interface supporting a common set of services. A brief description of tasks accomplished by some relevant subcommittees of the IEEE 802 committee is given in Table 1, and Figure 1 shows a picture of the networking stack standardized by this committee.

Subcommittee	Assigned/Accomplished Tasks
802.1	Common Issues
802.2	Logical Link Control (LLC)
802.3	CSMA/CD-based LANs
802.4	Token Bus LAN
802.5	Token Ring LAN

Table 1: Tasks of Various IEEE 802 Subcommittees

A LAN segment uses a physical medium that is shared by all the end stations connected to that segment. That is, the physical medium is of a broadcast type and the end stations

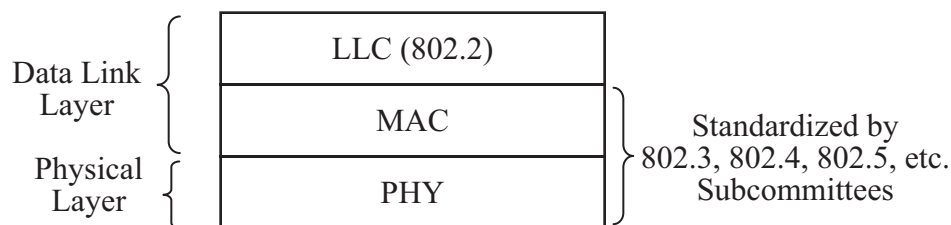


Figure 1: Networking Stack Showing the Layers Standardized by IEEE 802 Committee.

(or hosts) contend with one another to transmit a chunk of data, referred to as a frame, on the LAN segment. All such stations are said to be in the same collision domain. If two stations transmit at the same time, a collision happens and a back-off mechanism is used by the hosts to retry the transmission of the frame.

Sometimes, either a hub or a repeater is used to connect many LAN segments together. Many such devices use store-and-forward technology, while others may forward the frame as it is being received. In the former case, each LAN segment has its own independent collision domain while in the latter case, LAN segments connected through the device are part of the same collision domain. In either case, both LAN segments belong to the same broadcast domain. Unfortunately, distinguishing a hub from a repeater has become quite vague these days with the proliferation of devices that enjoin LAN segments. We will not attempt to differentiate between these devices, especially because most devices available these days use store-and-forward technology. A bridge also enjoins LAN segments and uses store-and-forward technology, and may well be confused with a hub or a repeater, but we will clearly differentiate a bridge from other LAN devices as being a LAN device that has the capability to run the spanning tree algorithm. Most bridges also learn end station addresses and perform selective forwarding based on such learning.

The terms MAC bridge, ethernet bridge, LAN switch, ethernet switch all refer to a bridge. Note, however, that a switch is a generic term and may sometimes refer to devices other than bridges. The term LAN segment may refer to a single LAN segment or different segments connected through a hub or repeater within the same collision or broadcast domain. An extended LAN is a topology of many LAN segments connected using one or more bridges.

Every broadcast medium must use a mechanism to identify the sender and the receiver. On CSMA/CD based LANs, a LAN address, also known as MAC address, ethernet address, hardware address, or physical address, is used to identify the sender and the intended receiver. Every ethernet frame contains a source LAN address and a destination LAN address. IEEE has standardized 48-bit addresses for use with IEEE 802.3 LANs. LAN addresses are globally unique and are *burned* into a LAN card by the manufacturer. The first three octets of each

48-bit LAN address represent the vendor code, also referred to as organizationally unique identifier (OUI), and indicate a block of  $2^{24}$  LAN addresses. A vendor, who may purchase one or more blocks of LAN addresses, assigns the last three octets of a block to uniquely identify the complete 48-bit address. Note that a single vendor may buy address blocks, each of which has different OUIs. An example ethernet address using a well-accepted representation is 00:0F:1F:B2:0E:CC, where 000F1F represents the manufacturer (DELL) and the remaining portion is selected by the manufacturer, uniquely for each card.

The sending host includes the source and destination LAN addresses within each frame. The last bit of the first octet of the destination LAN address is the first bit that is sent on the physical medium. A 0 value of this bit means that the desired destination is a single host, while a 1 in this bit position designates a multicast destination address indicating that this frame is intended for multiple hosts. Because of the broadcast nature of the LAN, every connected station receives every frame transmitted in the same broadcast domain and immediately examines the destination address. If the destination address is a unicast LAN address different from the receiving station's address, the frame is discarded, otherwise it is processed further. Bridges, on the other hand, must process all the frames received at all their ports, a port being a physical point connecting a bridge to a LAN. We refer to this mode of listening, where a bridge listens to (and processes) every frame, as promiscuous listening. For efficiency reasons, end hosts do not usually listen promiscuously but bridges must listen promiscuously since they are meant to forward frames from one segment to other segments.

## 2 Transparent Bridges

Bridges can be broadly classified as either transparent or source-routing. Bridges are transparent in the sense that two stations on two different segments of an extended LAN, formed by using transparent bridges, communicate as if they are on the same LAN segment. That is, end stations and other bridges are unaware of the presence of a transparent bridge in the network. On the other hand, source routing bridges explicitly forward frames to other source routing bridges in the network by specifying a path included within the frames. In this chapter, we will only consider transparent bridges as those are the ones commonly used in practice. Figure 2 shows a number of end stations connected to a LAN segment while Figure 3 indicates the same end stations divided onto two LAN segments connected through a bridge. If the bridge is transparent, the end stations will not notice any difference in topology in the two cases.

The two obvious benefits of bridges are an increased number of supported end stations

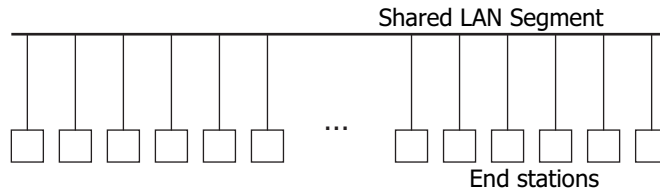


Figure 2: Stations Connected to a LAN segment.

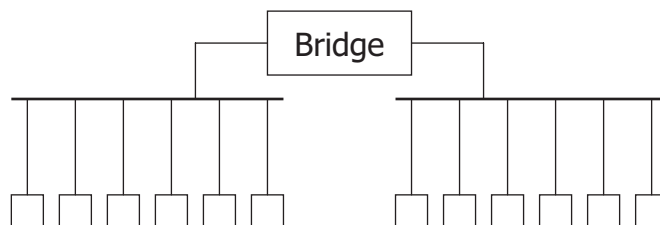


Figure 3: A Bridge connecting two LAN segments.

and an extension in the geographical range over which a LAN can be used. A third benefit which is not so obvious is the potential for increased transmission rate per LAN segment. This benefit stems from the fact that bridges may forward frames *smartly* such that a frame is not blindly forwarded to every LAN segment as is done in other store-and-forward LAN devices (such as repeaters and hubs). Instead, a bridge forwards the traffic only to those LAN segments where necessary. This limits the traffic on a given LAN segment resulting in the possibility to accommodate more traffic per segment. To illustrate this benefit, consider a LAN segment with an average traffic  $R$  packets per second. If we divide the stations on this LAN segment into two groups, one on each LAN segment, on either side of a smart bridge as shown in Figure 4, then the average traffic on each segment will be  $\frac{R}{2}$  packets per second. On average, half of the traffic generated on each segment is directed for the same segment while the other half is destined for the other segment. Thus the average traffic on each segment is  $\frac{3R}{4}$ , and it must not exceed the throughput capacity  $C$  of the segment, i.e.,  $\frac{3R}{4} < C$  which means that the effective transmission rate may exceed  $C$  up to a value  $\frac{4C}{3}$  which is in contrast with the case when a single segment was used and the effective transmission rate was limited to the value  $C$ .

The increase in effective transmission rate is not only dependent upon the number of segments and bridges used to make an extended LAN topology but also the configuration in which the segments and bridges are connected. Note that the increase in transmission rate is possible because we assume that the bridges do not unnecessarily forward frames to all segments; such is not the case with other LAN devices which do not perform any learning, whether or not they use a store-and-forward mechanism.

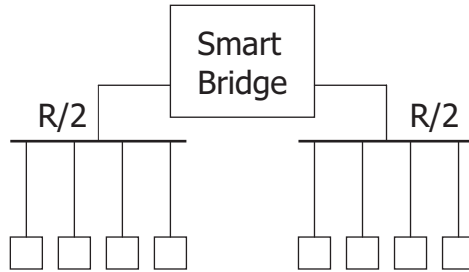


Figure 4: A Transparent Bridge with Smart Forwarding.

### 3 Bridge Operation

From earlier discussion, we learned that the end stations or hosts are connected to LAN segments, and LAN segments are connected to one another by means of bridges to form an extended LAN or simply a LAN. We may use a router to connect two LAN segments but this results in the two segments belonging to two different broadcast domains. In such a case, end hosts on different segments will need to communicate using a different mechanism compared to what would be used if they were on the same segment.

Transparent bridges listen to every frame promiscuously on each physical interface (port) and either filter (i.e., do not forward) or store-and-forward the frame to some of the other ports based on a local database called filtering database or forwarding database (FDB). The entries in an FDB indicate the port (or ports) where a frame for a given destination should be forwarded. To populate the forwarding database, for every received frame, the source LAN address is stored together with the port on which the frame is received. This is indicative of the fact that the transmitting end station is approachable through that particular port. The forwarding database is created in a memory area called the station cache. Sometimes, the terms forwarding database and station cache are used interchangeably. It is likely that a station might move from one place to another necessitating that the FDB entries be refreshed periodically.

For each received frame, the bridge examines the destination LAN address and if it is found to be a unicast address, the forwarding database is searched for the destination LAN address in the frame. If the destination LAN address is not found in the FDB, the frame is forwarded to all the ports except for the one on which it is received. However, if an entry for the destination LAN address is found in the FDB, the frame is forwarded to the corresponding port in that FDB entry. The frame is not forwarded at all if the port number in the FDB is the same as the one on which the frame is received, a situation which arises when the source and destination stations reside on the same LAN segment.



### 3.1 Learning and Forwarding Operation

To illustrate the learning and forwarding mechanism, consider the topology shown in Figure 5, adapted from [1]:

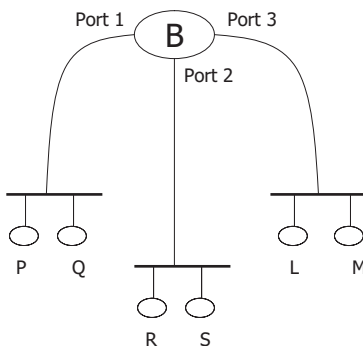


Figure 5: Learning and Forwarding in a Transparent Bridge.

Suppose that first frame transmitted in this topology is directed from P to L. To begin with, we assume that the station cache is empty. The bridge forwards the frame to ports 2 and 3 and makes an entry in the FDB that station P is approachable through port 1. Next, assume that M transmits a frame destined for P. Since the station cache now includes the station P, the frame is only forwarded to port 1. If the next frame is transmitted from station Q with P as the destination, the frame is not forwarded at all on any of the ports since the bridge determines that the source and the destination both reside on the same LAN segment. Note that the broadcast nature of the LAN segment ensures that station P would have received this frame directly. Finally, if station R transmits a frame with S as the destination, the frame is forwarded to ports 1 and 3 since there is no entry for station S in the FDB at that time.

The same learning and forwarding mechanism as described above is independently used at each bridge in the extended LAN. In fact, just as the end stations do not notice the presence of transparent bridges in an extended LAN, other bridges also remain unaware of the presence of transparent bridges in an extended LAN topology. Each bridge builds its own FDB based on the frames received at each port (or through manual intervention by the network administrator) and forwards or filters frames according to the FDB entries, irrespective of the FDB entries of other bridges.

### 3.2 Loop Free Operation

It is clear from the bridge operation described in the previous section that a bridge may make copies of a frame received at one of its ports for further transmission to some of its other

ports. This means that an extended LAN topology with loops is prone to the proliferation of frames, and frames may run in loops forever. Furthermore, the presence of loops may also cause the learning process to become unstable. Thus, it appears that loops in extended LANs should be avoided at all cost. However, loops may provide path redundancy when a portion of the extended LAN fails and, therefore, enforcing a loop-free topology by careful deployment of network resources (bridges and LAN segments) is not necessarily desirable. Instead, bridges in the topology may be designed to run a distributed protocol which logically prunes part of the topology in order to avoid loops. This pruning is obtained by blocking some ports of bridges and is temporary in the sense that the blocked ports may be unblocked if some portion of the network fails.

In the normal operation of bridges, some ports remain part of the extended LAN such that data frames are received from and transmitted to these ports while some ports are blocked for data frames. Here we differentiate between data frames which are generated by the end stations and control frames which are generated by bridges and communicated to other bridges to ensure proper working through a distributed protocol. This protocol reduces the original topology into a spanning tree which, by definition, connects all nodes (i.e., bridges) in a loop-free manner. While data frames are not allowed across blocked ports, bridges may receive control frames on such ports.

The distributed protocol must use an algorithm to automatically reconfigure an arbitrary topology into a loop-free topology in a short bounded time without significant overhead i.e., without transmitting too many control frames. The reduced topology is loop-free and is obtained by eliminating the blocked ports from the complete original topology. We call this reduced topology an *active topology*, which necessarily is a spanning tree. A popular algorithm for finding the active topology, originally designed by Radia Perlman, is called the spanning tree algorithm (STA), and it implicitly includes the protocol for exchanging control frames between the bridges. Sometimes, the STA is also referred to as the spanning tree protocol (STP); we will use both STA and STP interchangeably. Newer versions of STA have also been proposed and are used in practice. The spanning tree algorithm (STA) is the key ingredient of bridge operation and is given in detail in the following section. Subsequent sections will cover applications and variations of this algorithm.

## 4 Spanning Tree Algorithm (STA)

In an extended LAN (a.k.a. Switched Ethernet), each bridge (or ethernet switch or layer 2 switch) runs a distributed protocol – the spanning tree protocol (STP) sometimes referred to as the spanning tree algorithm (STA) – to facilitate the formation of a spanning tree of

the extended LAN. The resulting spanning tree is used by the Bridges to forward (or filter) the frames from one LAN segment to another.

The spanning tree is formed when bridges exchange control information amongst themselves and this step does not require any end station to be attached to any of the LAN segments. However, the (address) learning process requires that end stations are attached to LAN segments. Furthermore, learning is solely based on the information originated by the end stations and is not dependent upon the information originated by bridges.

## 4.1 Description

Although the terms spanning tree algorithm (STA) and spanning tree protocol (STP) are usually used interchangeably, and we will also use them as such, there is a clear difference between the two. An implementation of the spanning tree algorithm runs on each bridge in the extended LAN, while the spanning tree protocol is used to exchange information amongst bridges to create a spanning tree in order to eliminate loops that may exist in the topology. Thus, STA and STP both support each other during the formation of a spanning tree. As already mentioned, a spanning tree is needed in a topology of LAN segments connected by bridges because loops in the network topology create instability in learning besides the endless circulation of multicast and broadcast traffic.

The spanning tree is created by an exchange of special messages between bridges called Configuration Bridge Protocol Data Units, or Configuration BPDUs. We will use the term BPDU to refer to Configuration BPDUs throughout this chapter. In the spanning tree protocol, bridges exchange and process the BPDUs. Towards this end, one bridge is selected to be the root of the spanning tree. The spanning tree is then created which consists of shortest paths from each bridge to the root.

To explain the spanning tree algorithm and protocol, we assume that the BPDU message has the following form:

<Root ID>.<Cost>.<Transmitting Bridge ID>.<Transmitting Port ID>

Thus, a BPDU contains all of the information given in Table 2.

## 4.2 Protocol Operation

As described above, the protocol serves to build a tree in which one bridge is determined to be the root, while the other bridges make up a spanning tree. Bridges can only forward data frames towards the root bridge or away from the root bridge.

Data	Explanation of Data
Root ID	ID of the bridge currently believed/known to be the root
Transmitting Bridge ID	ID of the bridge transmitting this BPDU
Cost	Cost of the least cost path from the transmitting bridge to the currently known root
Transmitting Port ID	ID of the port which transmits the BPDU

Table 2: Information Contained in BPDUs

Each bridge is assigned a unique bridge ID, typically by an administrator based on the ethernet addresses of its ports, and the root bridge is determined to be the one with the lowest bridge ID. When a bridge first boots up, it has no information about other bridges on the LAN, so it believes that it is the root bridge itself. Therefore, it builds a BPDU of the form  $\langle \text{its bridge ID} \rangle . \langle 0 \rangle . \langle \text{its bridge ID} \rangle . \langle \text{Port ID} \rangle$  and transmits this BPDU on each of its ports. A BPDU transmitted by a bridge on a given port is multicast to all the bridges on the LAN segment to which it is connected via that port. Likewise, a bridge receives BPDUs sent by all the other bridges on that segment. The bridge must process the received information and reevaluate which bridge it now believes to be the root bridge, and its distance to the new root bridge. Depending on the BPDUs received, the bridge may *block* some of its ports and *designate* others. Finally, the bridge sends out this newly determined information on all designated ports. The process of sending, receiving, and processing BPDUs continues until a stable spanning tree is formed. After the algorithm converges, one bridge has been elected as the root bridge, while all other bridges have calculated their shortest path cost to the root bridge. An example of a spanning tree in a given topology is shown in Figure 6.

The bridge with bridge ID 20 (or simply bridge 20) is determined to be the root bridge since it has the lowest bridge ID. Thus, all of its ports are designated. Every other bridge has found its root port, and either designated or blocked all other ports. A root port is the port that leads to the root bridge along the shortest path, and a designated port is an unblocked port connected to a segment. A blocked port is not part of the spanning tree, and no data traffic flows through it.

In Figure 6, bridge 50 has two ports that lead to the root bridge with the same cost. In this case, the transmitting port ID of bridge 20 is used to determine that port 1 of bridge 50 should be the root port (details later in Section 4.3).

There can only be one designated bridge per LAN segment, and it is the bridge with lowest cost to the root. Thus, root bridge is designated on all the LAN segments it is connected to, and all ports of the root bridge are designated. In case of a tie in determining

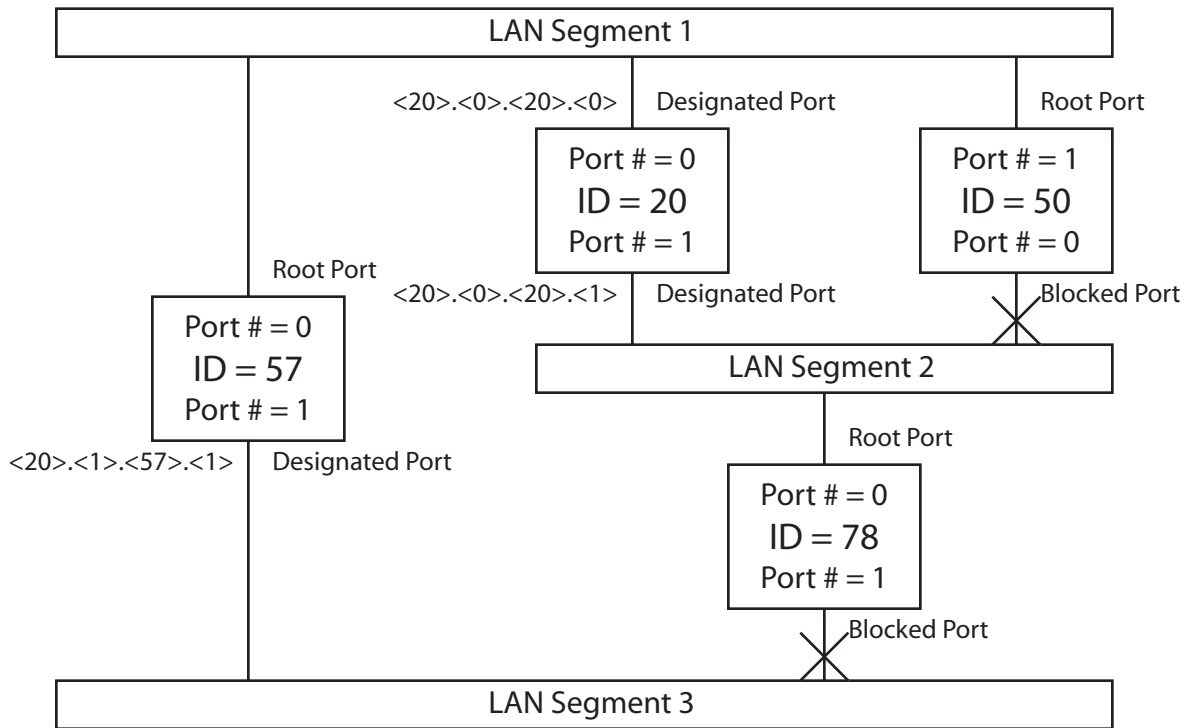


Figure 6: Spanning Tree Example

the designated bridge for a LAN segment, the bridge with the lowest bridge ID is chosen. Thus for segment 3, bridge 57 (rather than bridge 78) is chosen as the designated bridge.

### 4.3 Protocol Algorithm

The protocol algorithm is described below:

- 1) Initially each bridge assumes itself to be the root bridge and transmits the following BPDU on each of its ports:
 

$\langle \text{Transmitting Bridge ID} \rangle . \langle 0 \rangle . \langle \text{Transmitting Bridge ID} \rangle . \langle \text{Transmitting Port ID} \rangle$
- 2) During a period of time (called a cycle which can be arbitrarily short), each bridge receives BPDUs on each of its ports. For each port, the bridge saves the “best” amongst all BPDUs received and determines the BPDU that it would transmit on that port. When comparing the BPDUs, we use the following rules:
  - A BPDU with a lower root bridge ID is “better” than any other BPDU with a higher root bridge ID.
  - If the above rule results in a tie (i.e., the root bridge ID is the same), then a BPDU with lower cost is “better” than any other BPDU with higher cost.

- If both the above rules result in a tie (i.e., the root bridge ID and cost are the same), then a BPDU with a (numerically) lower transmitting bridge ID is better than any other BPDU with a higher transmitting bridge ID.
- If all three rules above result in a tie (i.e., the root bridge ID, cost, and transmitting bridge ID are the same), then a BPDU with a lower transmitting port ID is “better” than any other BPDU with a higher transmitting port ID.

Examples:

< 29 > . < 15 > . < 35 > . < 0 > is better than < 31 > . < 12 > . < 32 > . < 1 >  
 < 35 > . < 15 > . < 80 > . < 2 > is better than < 35 > . < 18 > . < 38 > . < 1 >  
 < 35 > . < 80 > . < 39 > . < 1 > is better than < 35 > . < 80 > . < 40 > . < 0 >  
 < 35 > . < 15 > . < 80 > . < 0 > is better than < 35 > . < 15 > . < 80 > . < 1 >

- 3) If a bridge receives a BPDU on a port that is better than the one that it would have transmitted on that port, it no longer transmits BPDUs on that port (that port will be named as either root port or blocked port). Therefore, when the algorithm stabilizes, only one bridge on each segment (the designated bridge for that segment) transmits BPDUs on that segment.

Based on the received BPDUs from all interfaces, each bridge independently determines the identity of the root bridge (which will eventually converge to a single identity). After receiving BPDUs, the bridge ID of the “newly determined” root bridge is, therefore, the minimum of a bridge’s own bridge ID and the root ID contained in all other received BPDUs. Each bridge is able to determine its lowest cost to the root, its root port, and its designated ports, if any.

As an example, assume that bridge B has a bridge ID of 18. Suppose that the “best” BPDU received by bridge B on each of its ports is as given in Table 3:

Port Number	Best BPDU Received on the Port
Port 1	< 12 > . < 85 > . < 51 > . < 0 >
Port 2	< 12 > . < 85 > . < 32 > . < 2 >
Port 3	< 81 > . < 0 > . < 81 > . < 1 >
Port 4	< 15 > . < 31 > . < 27 > . < 1 >

Table 3: Example: Best BPDUs received on each port

From the best BPDUs received by bridge B on its ports, it is clear that both port 1 and port 2 received BPDUs containing the lowest root ID. Bridge B has bridge ID

18 which is greater than 12. Therefore, bridge B is not the root bridge, and after B receives the BPDUs listed in Table 3, bridge B now believes that bridge 12 is the root bridge. Bridge B must also determine the cost to the root bridge; this cost is calculated by adding 1 to the lowest cost reported to bridge 12 by any of the best BPDUs given in Table 3. Likewise, bridge B must also select its root port—the port through which the root bridge can be reached via the smallest cost. Since the BPDUs received at port 1 and port 2 report the same cost to root (i.e., 85), the cost to root for bridge B is  $1 + 85 = 86$  and since the transmitting bridge ID is lower in the BPDU received at port 2, bridge B selects port 2 as its root port. Bridge B no longer transmits BPDUs on port 2.

Bridge B now constructs a BPDU  $\langle 12 \rangle . \langle 86 \rangle . \langle 18 \rangle . \langle \text{possible port number on bridge B} \rangle$ . This BPDU is better than the ones received at ports 3 and 4, so bridge B becomes a designated bridge for the two LAN segments, one connected to port 3 and the other to port 4. Port 3 and port 4 are the designated ports, and bridge B continues to transmit BPDUs on these ports.

Finally, the BPDU bridge B would send out is not better than the BPDU received at port 1. Port 1 is, therefore, blocked and bridge B no longer transmits BPDUs out of this port.

Thus, bridge B has determined the following:

- the root bridge and bridge B's root port (bridge 12 and port 2)
  - its own distance (cost) to the root (86)
  - ports for which it is designated (3, 4)
  - blocked ports (1)
- 4) At each step, a bridge must decide which ports to include in the spanning tree and which ports to exclude. The root port and all designated ports are included, whereas all blocked ports are excluded. Thus, in the above example, ports 2, 3, and 4 are included in the spanning tree while port 1 is excluded.

Bridges place all ports included in the spanning tree in the “Forwarding” state. A bridge receives data frames from ports in the forwarding state and transmits frames on ports in the forwarding state. Bridges do not, however, accept data from blocked ports nor do they transmit data on blocked ports (this doesn't apply to BPDUs).

By having all bridges running the same algorithm, a spanning tree will be formed that connects every LAN segment into a loop-free tree. Data frames can, therefore, be transported along the branches of the tree.

## 4.4 Address Learning and Frame Forwarding

When a data frame arrives at a port on a bridge, the bridge must determine on which outgoing port the frame should be transmitted. The decision is made using the forwarding database (FDB) which, as already mentioned, is simply a table containing the destination MAC addresses and the appropriate output port that should be used to reach those destinations.

To maintain the FDB and forward frames correctly to the destinations, a bridge will use the following procedure:

- When a bridge receives a frame, the bridge stores the frame's source address in the forwarding database along with the port on which the frame was received.
- The bridge then looks up the frame's destination address in the forwarding database:
  - If the destination address is found, the bridge forwards the frame on the outgoing port specified in the forwarding database (except when the frame is received from the same port).
  - If the destination address is not found, the bridge forwards the frame on all the outgoing ports except for the one on which the frame was received.

## 4.5 STP Example

We notice that the spanning tree algorithm is simply a distributed version of the shortest path algorithm in which once a root bridge is determined, the shortest paths from every other bridge to the root bridge are included in the *active topology*. To illustrate the application of the STP, consider the topology shown in Figure 7.

For this topology, we demonstrate the step-by-step procedure the spanning tree protocol and algorithm use, in a sequence of diagrams. In each diagram, we show one full cycle of the exchange of BPDUs. Progressing through these cycles, the topology converges to a final loop-free topology shown as the last in the sequence of diagrams. In each cycle, we also indicate BPDUs exchanged between bridges (without being particular about the port numbers for selecting the best BPDU). The sequence of diagrams are shown in Figures 8 and 9.



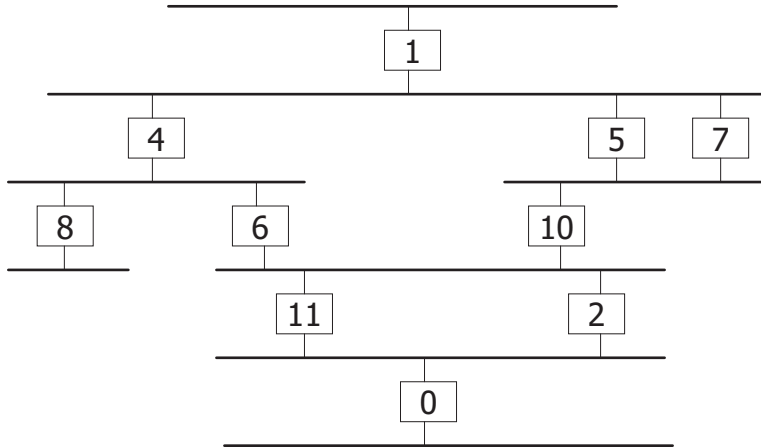


Figure 7: Example to Illustrate the application of STP.

## 4.6 Reconfigurations and STP

One goal of the spanning tree protocol is to allow for the formation of a new spanning tree in case the extended LAN topology changes. Topology changes are part of regular LAN operation in which segments and bridges are added, removed or moved around in the topology. Topology changes may also result by upgrading a segment to run at a higher speed (e.g., from 100 Mb/s to 1 Gb/s). Sometimes, a topology change also occurs when a segment or a bridge (or another LAN device) malfunctions. The STP should deal with topology changes efficiently and should quickly find a new spanning tree.

For spanning tree maintenance, the STP acts as a soft-state protocol which means that bridges must be assured of the health of the LAN topology from time to time. For this purpose, in addition to the best configuration message received at each blocked or root port, bridges store an *Age Field*. The bridge keeps on incrementing the Age Field until a configuration BPDU is received, at which time the Age Field is reset. Otherwise, the Age Field reaches the *MaxAge* at which point the bridge assumes that the topology has changed and, consequently, runs the STA from the start.

Under normal operation, the root bridge generates and then transmits the configuration BPDUs at every *hello time* with the Age Field set to zero within the message. When any Bridge in the downstream receives a message coming from the root (received at the root port), it transmits a BPDU on designated ports with Age Field set to zero within the message. Thus, if the root bridge fails or the path to the root bridge becomes unavailable, a bridge stops receiving fresh messages and eventually times out (Age Field becomes equal to MaxAge) at which point the root bridge, cost to the root bridge, and the root port are all calculated from scratch. While timing out on root ports will always cause everything (root,

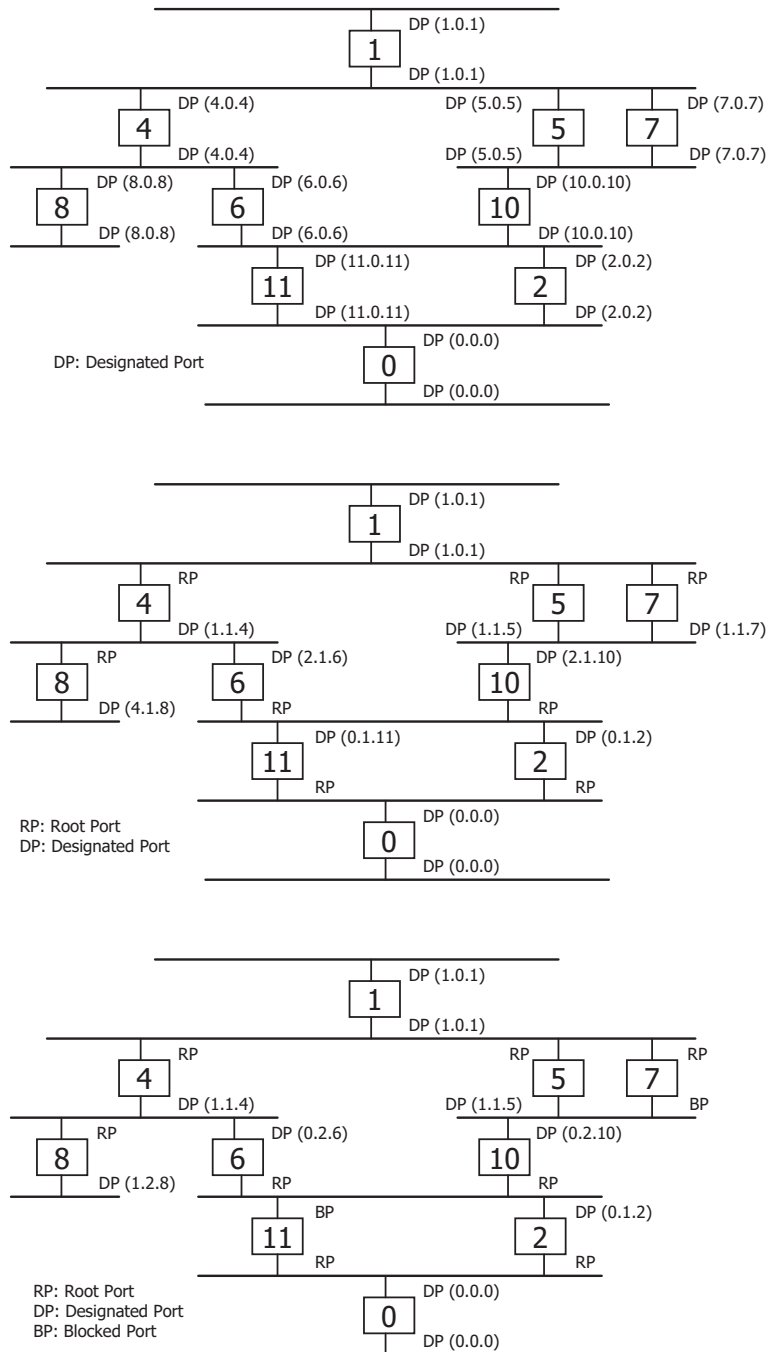


Figure 8: Example to Illustrate the application of STP (continued to next figure).

cost to root and root port) to be recalculated, timing out on a blocked port may or may not indicate a significant change in topology, and thus may not warrant a fresh running of the spanning tree protocol.

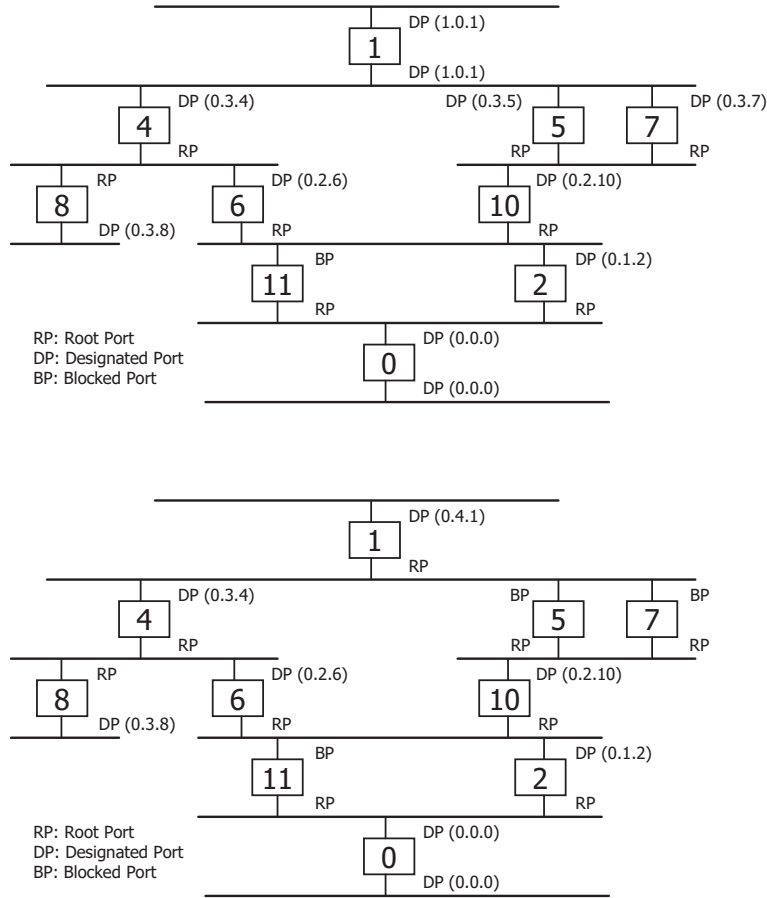


Figure 9: Example to Illustrate the application of STP (continued from previous figure).

## 4.7 Reconfigurations and Loops

When an extended LAN undergoes a topology change and reconfiguration, different ports of the bridges change their state. The ports across which data frames were not allowed are blocking (blocked ports in an active topology) may now be required to move to the forwarding state (root and designated ports allow data frames across and are said to be in forwarding state). Similarly, ports which were previously in the forwarding state may now be required to move to a blocking state. Since the STP converges after some iterations, it is possible that temporary loops and temporary disconnections will appear, even though the new final topology is guaranteed to be loop-free. Even if they are temporary, loops cause frame proliferation and, therefore, it is desirable to avoid temporary loops in a changing topology. The STP used in bridges ensures that loops are avoided by forcing ports in the blocking state to wait for some time before switching to the forwarding state. This wait period is set to be large enough to allow the topology change information to propagate through the network. Obviously, a port that needs to move to the blocking state in the new

topology from the forwarding state in the old topology is allowed to switch state immediately. This may cause temporary disconnections resulting in frame drops, which is an acceptable LAN behavior as reliable delivery of data is left to the higher layer protocols.

When a port is being moved from the blocking state to the forwarding state, the IEEE 802.1D standard defines two intermediate states during the wait period. In the *listening* intermediate state, the bridge listens but does not learn station addresses through that particular port, while in the *learning* intermediate state, address learning is enabled. In both intermediate states, bridges do not forward any data frames. A state diagram reproduced from the IEEE 802.1D standard depicting state transitions from blocking state to forwarding state is shown in Figure 10.

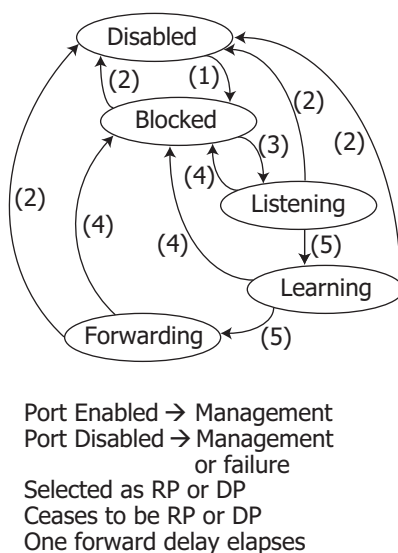


Figure 10: Illustrating the Transition from Blocking to Forwarding State.

## 4.8 BPDU Format

The configuration BPDUs exchanged between bridges for the operation of the STP have a generic format given in Figure 11. The TC flag in the bridge message format indicates a topology change. When a bridge notices that a port has moved into or out of the blocked state, it transmits a topology change notification on the segment connected to its root port (by setting the TC flag). The bridge designated on that segment (which receives the topology change notification on the designated port), sets the topology-change acknowledgement (TCA) flag in the next BPDU transmitted onto that segment so that the bridge which has launched the topology-change notification is informed that action is being taken about the reported topology change. In addition to sending an acknowledgement (by means of

Number of octets				
2	Protocol Identifier			
1	Version			
1	Message Type			
1	TCA	Reserved	TC	Flags
8	Root ID			
4	Cost of Path to Root			
8	Bridge ID			
2	Port ID			
2	Message Age			
2	Max Age			
2	Hello Time			
2	Forward Delay			

Figure 11: Format of Message Exchanged Between Bridges.

TCA flag) to the bridge indicating the topology change, this bridge also relays the topology change notification on its own root port towards the root bridge. In this manner, the topology change notification reaches the root bridge which sets the TC flag in its BPDUs for a time period equal to the sum of MaxAge and *forward delay*. The forward delay time is a network wide parameter that indicates the longest distance between any two bridges in the extended LAN, and may be set by the network manager to ensure that the topology change notification is propagated to all the bridges in the extended LAN before a port is actually moved from the blocking state to the forwarding state.

## 5 Bridges and LAN Multicast

When a source intends to send a frame to multiple receivers, a naive solution is to broadcast the frame throughout the extended LAN such that the frame is transmitted on every segment and received by all the hosts. Hosts interested in this frame pass it to their higher layers while hosts uninterested in this frame drop it at the link layer. This solution has some inherent problems: first, broadcasting multicast traffic over slower links might cause those links to saturate and second, transmitting a message to a large number of stations in an extended LAN when it is only intended for a few stations does not seem to be a reasonable option, even if all the links in the LAN may have high capacity.

A solution to disseminate multicast traffic only through those portions of the LAN where it is actually required is to allow the end stations (or hosts) to explicitly indicate their interest in receiving traffic on given multicast addresses. Entries for multicast addresses are entered into the same filtering database (FDB) that is used to forward or filter unicast addresses. Such multicast entries may either be added statically by a network manager

or could be entered dynamically by using a protocol called GARP Multicast Registration Protocol (GMRP). If a bridge only allows static multicast entries (that may be manually entered), the bridge is said to be providing *Basic Filtering Services*. In contrast, provision of *Extended Filtering Services* requires that the bridge additionally allows dynamic multicast forwarding entries be entered into the FDB. In either case, a bridge must allow static unicast entries (entered manually) as well as dynamic unicast entries (through the address learning process). A bridge conforming to the IEEE 802.1D standard must provide basic filtering services and may optionally provide extended filtering services.

As with unicast FDB entries, multicast forwarding entries in a bridge's FDB enlist the set of ports on which a frame should be forwarded indicating the presence of at least one end station, approachable through a listed port, interested in receiving that particular multicast frame. Thus, each port of a bridge keeps track of the interest of end stations that may be accessible through that port. Note that if multiple end stations, accessible through a port of a bridge, are interested in receiving a multicast frame, it is only necessary for the bridge to transmit just one copy of the frame on that port.

End stations indicate their interest in receiving a frame destined for a specific multicast address using GMRP, which uses the services of Group Attribute Registration Protocol (GARP). While the details encompassed by GMRP are too many to describe here (and the reader is referred to the IEEE 802.1D standard), briefly, GMRP includes mechanisms that allow end stations to register (i.e., indicate interest in a multicast group address) or deregister membership information with the bridges attached to the same LAN segment. GMRP also allows bridges to disseminate multicast group registration information throughout the extended LAN. For each bridge in the extended LAN, a port is marked to forward a multicast frame on a port if at least one member of that multicast group is accessible through that port. Similarly, forwarding of multicast traffic on a port is retracted when all members of the multicast group which are accessible through that port have deregistered.

As a result of GMRP operation, frames sent to a particular multicast group address can be received on all LAN segments to which at least one end station interested in receiving frames sent to that multicast group address is attached. A multicast frame is also transmitted on those LAN segments which are on the way, along the active topology, from the source of multicast frame to any intended recipient of that frame. It is important to note that the source of a multicast frame may reside anywhere in the topology and may not need to explicitly register for that multicast address, unless it also wishes to receive frames sent to that multicast address. Thus, any station that wishes to send frames to a particular group can do so from any point of attachment in the extended LAN. This property is sometimes referred to as *Open host group concept*.

## 6 Generic Attribute Registration Protocol (GARP)

As described in Section 5, dissemination and distribution of interest in a particular multicast group is a function of GMRP for which it uses GARP. In fact, GARP may be used to register for and disseminate information about *any* generic attribute in a bridged LAN. A multicast group address is just one example of a generic attribute. Thus, GMRP may be considered as an application which uses GARP service. Another application which uses the services of GARP, in much the same way as GMRP does, is GARP VLAN Registration Protocol (GVRP). GVRP uses the services of GARP to disseminate information about Virtual LAN IDs (VLAN ID) which is another generic attribute. The attributes registered, deregistered, or propagated through GARP are opaque to GARP itself in the sense that it is up to the application (such as GMRP or GVRP) to interpret a generic attribute.

### 6.1 GARP Operation

The two principal control messages used in GARP to register (or deregister) an attribute are:

**Join Message** This message is initiated by an end station (and relayed by the bridges) to register an attribute.

**Leave Message** This message is initiated by an end station (and relayed by the bridges) to deregister an attribute.

There are other messages derived from these two principal messages, details of which can be found in the IEEE 802.1D standard. The Join and Leave messages are initiated by end stations to register or deregister for a particular attribute (a VLAN ID or a multicast group address).

As previously mentioned, if multiple end stations, all accessible through a port of a bridge, are interested in receiving a multicast frame, it is only necessary for the bridge to transmit just one copy of the frame on that port. Thus, if there are multiple end stations on a single segment, all interested in a given attribute, only one of them needs to register for the said attribute. This indicates to the bridges connected to that segment that there is at least one end station on this segment which is interested in receiving frames corresponding to the given attribute, and thus the bridges will forward the frame on their ports attached to this segment. In contrast, a bridge will not forward a frame with given attributes on a given port if no one has indicated an interest in that attribute from that port, indicating that none of the interested recipients is accessible through this port.

For each bridge that implements an application that uses GARP, a GARP *participant* exists at each port for each application (such as GMRP or GVRP). Each GARP participant further consists of a GARP *application* component and a GARP *Information Declaration* (GID) component. While the application component has more to do with the application (attribute type and values, how to send frames to the application, etc.), the GID component is more relevant with the actual declaration (register and deregister) and dissemination of an attribute. The GID component further consists of two state machines, an *Applicant* state machine which keeps track of the interest level of end stations in a given attribute and a *Registrar* state machine which allows the bridge to selectively forward (or filter) a frame for a given attribute. When someone registers for an attribute, the associated registrar at the port, after listening the registration message, should be moved to a state (IN) for that attribute. While in the IN state for an attribute at a given port, the bridge should forward the frames on that port for that attribute. A Leave message (or a timeout) for a given attribute on a given port should cause the registrar move to MT state indicating that there is no need for the bridge to forward a frame with that attribute on that port.

## 6.2 Virtual LANs and GVRP

An extended LAN may cover a large campus covering a large subnetwork with many end stations. It is likely that people in one department are not interested in receiving frames that really belong to another different department. This calls for the formation of Virtual LANs (VLANs) on the active topology such that the unicast, multicast or broadcast traffic remains limited to a portion of the active topology (the VLAN). Thus, a VLAN is a collection of LAN segments and end stations connected to them within an extended LAN that has exactly the same properties of an independent LAN. In an extended LAN comprising several VLANs, traffic belonging to a VLAN is restricted from reaching users in other VLANs.

End stations indicate their interest in one or more VLANs by sending register message for those VLANs on the segments to which they are attached. The register message includes the generic attribute which, in this case, is the VLAN ID. Differentiation among traffic belonging to different VLANs is accomplished by the addition of VLAN tags (VLAN ID or VID) to frames; bridges use VIDs to appropriately filter the frames.

The VID is attached with a frame as an additional data called a *Tag*. A Tag consists of 16 bits, 12 of which are used for the VID. It is expected that VLAN deployment in a bridged LAN is incremental such that some station will still be VLAN-unaware. Such legacy station are not able to handle the VLAN Tags. Thus, VLAN-aware bridges must ensure that tagged frames (frames which include VLAN Tags) are untagged before they are forwarded



to a legacy station. Thus, each bridge maintains a set of ports called *Untagged Set* which consists of ports through which frames that are transmitted shall be sent untagged. In other words, a bridge removes the tags (if already present) from the frames before forwarding them on any port that is included in the untagged set.

The actual realization of a VLAN is made possible by means of another set called the *Member Set*, which consists of those ports which are included in a given VLAN. Each bridge maintains a member set corresponding to each VLAN. Since a VLAN is a subset of the spanning tree, none of the blocked ports may be included in any member set of any bridge. In an extended LAN where all bridges are VLAN-aware, traffic belonging to a VLAN is restricted to that VLAN and is not allowed to leave the VLAN.

A VLAN is also represented by a tree (a VLAN is a subset of spanning tree and different VLANs may overlap), which is confined to some (or all) of the nodes (bridges) in the extended LAN. Forwarding and filtering of frames on a VLAN is very similar to how it is done if VLANs are not implemented. However, learning is done on each VLAN independently. That is, for each VLAN, the port through which particular end stations can be reached is determined and entered into the FDB for that VLAN. Sometimes, shared learning on multiple VLANs is also desirable and it can be configured using another parameter (FID) of bridges. Many commercial bridges available these days are VLAN-aware and their VLAN capabilities are frequently used.

## 7 Summary and Recent Trends

Bridges are an essential part of today's local area networks. They allow a LAN to scale from a small building to a large campus, and from a few stations to a large number of stations. Bridges perform learning to avoid unnecessary forwarding of data traffic onto LAN segments. This selective forwarding can also be used to register specific attributes with the bridges such that traffic corresponding to a multicast group or a virtual LAN is forwarded to the appropriate interfaces.

A spanning tree protocol (STP) is used by the bridges to determine the active topology over which data frames will flow in an extended LAN. Recent versions of STP include rapid spanning tree protocol (RSTP) which results in faster convergence to a final spanning tree after a topology change. This is possible by allowing ports to move from blocked state to forwarding state a bit sooner. Another variation of STP is multiple spanning tree protocol (MSTP) which uses overlapping multiple spanning trees, each with its own root bridge, for efficiently utilizing the LAN segments in an extended LAN.

Traditionally, bridges are considered as the devices that just extend the scope of a local

area network in a given campus, while different campuses are connected to each other using routers. However, advances and cost-effectiveness of ethernet technology has led to the use of ethernet standard over a larger, metropolitan, area. In this case, subscribers or branch offices of a larger business may be connected over a metropolitan area using bridges, thus forming a LAN that covers an entire metropolitan area.

## 8 Glossary

**Bridge** A layer 2 packet switch that receives a frame from any of its interfaces and appropriately forwards that frame to some of its other interfaces.

**Packet** A formatted block of data carried by a computer network from one node to another.

**Packet Switch** A network device that receives packets from any of its incoming interfaces and appropriately forwards it to some of its outgoing interfaces based on the information contained in the packet header.

**Router** A layer 3 packet switch that forwards layer 3 packets based on the information in the packet header.

**Frame** A formatted block of data sent from one node to an adjacent node in a computer network. A frame is an entity that is dependent upon the link layer technology.

**LAN** Local area network, which is used to signify either a LAN segment or an extended LAN.

**Extended LAN** A conglomerate of LAN segments joined by bridges.

**Hub** A LAN device that connects two or more LAN segments, usually on separate collision domains, but does not run the spanning tree algorithm or performs any learning on these segments. A hub broadcasts a frame received on a port to all the other ports.

**Repeater** A LAN device that connects two or more LAN segments, usually on the same collision domain, but does not run the spanning tree algorithm or performs any learning on these segments. A repeater broadcasts a frame received on a port to all the other ports.

**Transparent Bridge** Bridges are transparent in the sense that two stations on two different segments of an extended LAN, formed by using transparent bridges, communicate as if they are on the same LAN segment.

**IEEE** The Institute of Electrical and Electronics Engineers, Inc.

**Collision domain** The extent of a physical medium over which two or more frames can not coexist without interfering.

**CSMA/CD** A multiaccess protocol in which each transmitter senses the carrier prior to transmission and also employs a mechanism to detect a collision.

**Broadcast domain** The extent of an extended LAN. All devices in the same broadcast domain can be reached by sending a frame to the data link layer broadcast address.

**MAC** Media Access Control (MAC) is a layer in a layered network architecture that provides an addressing and channel access control mechanism to many nodes connected to the same physical media.

**LLC** Logical link control (LLC) is the upper sublayer of the data link layer. It is independent of the underlying physical layer and acts as an interface between the MAC layer and the network layer.

**FDB** Forwarding database (FDB) of a bridge is a table that lists the forwarding information, mapping a destination MAC address to a set of outgoing ports.

**Spanning tree** A loop-free active topology that traverses all the bridges in a bridged LAN comprising a set of LAN segments connected by the bridges. A spanning tree is obtained by using the spanning tree protocol.

**Active topology** A topology comprising portions of a bridged LAN over which transmission of data is allowed. Control traffic is usually used to obtain an active topology, and may be transmitted over portions of the local area network that are not part of the active topology.

**STA** The algorithm used by the spanning tree protocol to find a spanning tree that can be used as active topology in a bridges local area network.

**STP** Spanning tree protocol is used to find a spanning tree in a bridged local area network.

**BPDU** A frame that is used to exchange control traffic between bridges that run the spanning tree protocol.

**Root bridge** The bridge with lowest identification number.

**Designated bridge** Designated bridge of a LAN segment is the bridge responsible for relaying data frames on a given LAN segment. A given bridge may be designated for multiple LAN segments.

**GVRP** GARP VLAN registration protocol is used to register and deregister virtual local area networks (VLANs) in a bridged LAN.

**VLAN** Virtual local area network is a subset of a bigger local area network. Data traffic in a VLAN is confined within the subset that represents the VLAN.

**GARP** Generic attribute registration protocol is used to register and deregister attributes whose corresponding data traffic is of interest to a bridge or an end station.

**GMRP** GARP multicast registration protocol is used to register and deregister multicast group addresses in a bridged LAN, for which an end station or a bridge is interested in receiving data traffic.

## References

- [1] R. Perlman, *Interconnections: Bridges and Routers*, 2000.
- [2] IEEE 802 Standards Committee, *802.1D-2004: 802.1D MAC Bridges*, 2003-2004.
- [3] W. Stallings, *Data and Computer Communications, 8th Edition*, 2006.
- [4] A. S. Tanenbaum, *Computer Networks, 4th Edition*, 2002.
- [5] N. Olifer and V. Olifer, *Computer Networks: Principles, Technologies and Protocols for Network Design*, 2006.
- [6] R. Seifert, *The Switch Book: The Complete Guide to LAN Switching Technology*, 2000.
- [7] J. Matthews, *Computer Networks: Internet Protocols in Action*, 2005.
- [8] G. Keiser, *Local Area Networks*, 2001.
- [9] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet (3rd Edition)*, 2004.
- [10] B. A. Forouzan, *Data Communications and Networking, 3rd Edition*, 2003.
- [11] B. A. Forouzan, *Local Area Networks* 2002.
- [12] [http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/sw\\_ntman/cwsimain/cwsi2/cwsiug2/vlan2/stpapp.htm](http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/sw_ntman/cwsimain/cwsi2/cwsiug2/vlan2/stpapp.htm) (STP description)
- [13] <http://www.cisco.com/warp/public/473/146.html> (RSTP description)
- [14] <http://www.cisco.com/warp/public/473/147.html> (MSTP description)
- [15] <http://www.firewall.cx>
- [16] <http://www.vpls.org>
- [17] <http://www.metroethernetforum.org>
- [18] <http://linux-net.osdl.org/index.php/Bridge>
- [19] [http://en.wikipedia.org/wiki/Network\\_bridge](http://en.wikipedia.org/wiki/Network_bridge)