Multiple Spanning Tree Protocol

The **Multiple Spanning Tree Protocol** (**MSTP**) and algorithm, provides both simple and full connectivity assigned to any given Virtual LAN (VLAN) throughout a Bridged Local Area Network. MSTP uses BPDUs to exchange information between spanning-tree compatible devices, to prevent loops in each MSTI (Multiple Spanning Tree Instances) and in the CIST (Common and Internal Spanning Tree), by selecting active and blocked paths. This is done as well as in STP without the need of manually enabling backup links and getting rid of bridge loops danger.

Moreover, MSTP allows frames/packets assigned to different VLANs to follow separate paths, each based on an independent MSTI, within MST Regions composed of LANs and or MST Bridges. These Regions and the other Bridges and LANs are connected into a single Common Spanning Tree (CST).

Contents

History and motivation

It was originally defined in IEEE 802.1s as an amendment to 802.1Q, 1998 edition and later merged into IEEE 802.1Q-2005 Standard, clearly defines an extension or an evolution of Radia Perlman's Spanning Tree Protocol (STP) and the Rapid Spanning Tree Protocol (RSTP). It has some similarities with Cisco Systems' Multiple Instances Spanning Tree Protocol (MISTP), but there are some differences.

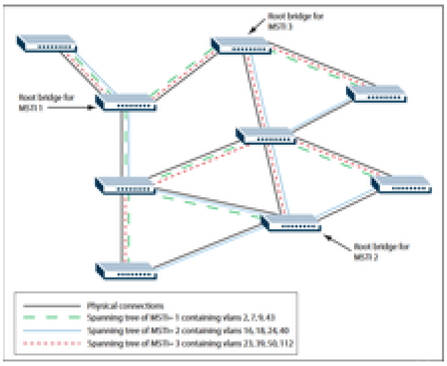
The original STP and RSTP work on the physical link level, preventing bridge loops when redundant paths are present. However, when a LAN is virtualized using VLAN trunking, each physical link represents multiple logical connections. Blocking a physical link blocks all its logical links and forces all traffic through the remaining physical links within the spanning tree. Redundant links cannot be utilized at all. Moreover, without careful network design, seemingly redundant links on the physical level may be used to connect *different* VLANs and blocking any of them may disconnect one or more VLANs, causing *bad paths*.

Instead, MSTP provides a potentially better utilization of alternate paths by enabling the use of alternate spanning trees for different VLANs or groups of VLANs.

Main Entities

Multiple Spanning Tree Instances (MSTI)

As MSTP enables grouping and mapping VLANs into different spanning tree instances, there's an urge of determining a group or set of VLANs, which are all using the same spanning tree, this is what we come to know as a MSTI.
Each instance defines a single forwarding topology for an exclusive set of VLANs, by contrast, STP or RSTP networks contains only a single spanning tree instance for the entire network, which contains all the VLANs. A region can include:[1]


Root bridge for MSTI 3
Root bridge for MSTI 1
Root bridge for MSTI 2
Physical connections
Spanning tree of MSTI- 1 containing vlans 2, 7, 9, 43
Spanning tree of MSTI- 2 containing vlans 16, 18, 24, 40
Spanning tree of MSTI- 3 containing vlans 23, 39, 50, 112

Different Spanning trees created by different MSTIs on the same physical layout.

- **Internal Spanning-Tree Instance (IST)**: Default spanning tree instance in any MST region. All VLANs in this IST instance conform a **single** spanning tree topology, allowing only one forwarding path between any two nodes. It also provides the root switch for any VLAN configured switches which are not specifically assigned to a MSTI.
- **Multiple Spanning Tree Instance (MSTI)**: Unlike IST, this kind of instance comprises all static VLANs specifically assigned to it and at least, must include one VLAN.

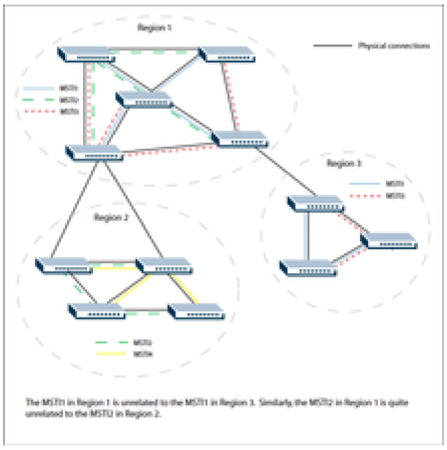
While each MSTI can have multiple VLANs, each VLAN can be associated with only one MSTI.

MSTP Regions

A set of interconnected switches that must have configured the same VLANs and MSTIs, also have the same following parameters:

- **MST Configuration Name**
- **Revision Level**
- **Configuration Digest:** Mapping of which VLAN are mapped to which MST instances.

An MSTI is unable to span across MST regions because of its inherent locality to a single MST region. This is done by an identifying number for each MSTI. For achieving the task of assigning each bridge to a region, each switch/bridge must compare their **MST Configuration Identifiers (Format Selector, Region Name, Revision Level and Configuration Digest)**, either of them represents VLAN to MSTIs mapping for each bridge.

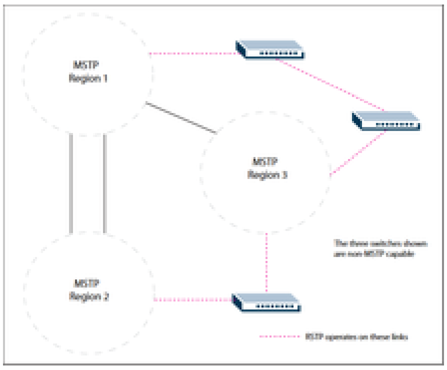

Region 1
Physical connections
MSTI1
MSTI2
MSTI3
Region 3
MSTI1
MSTI3
Region 2
MSTI2
MSTI4
The MSTI1 in Region 1 is unrelated to the MSTI1 in Region 3. Similarly, the MSTI2 in Region 1 is quite unrelated to the MSTI2 in Region 2.

MSTIs in different regions.

Common and Internal Spanning Tree (CST/CIST)

We can differentiate two kinds of conformated Spanning Trees into the different networks created by MSTP, these are:

- **Common Spanning Tree (CST):** Administers the connectivity among MST regions, STP LANs and RSTP LANs in a bridged network.
- **Common Internal Spanning Tree (CIST):** Identifies regions in a network and administers the CIST root bridge for the network, for each region and for each spanning tree instance in each region. It's also the default spanning tree instance of MSTP so that any VLAN which isn't a member of a particular MSTI, will be a member of the CIST. Furthermore, works as well as the spanning tree that runs between regions and between MST regions and Single Spanning Tree (SST) entities.


MSTP Region 1
MSTP Region 3
MSTP Region 2
The three switches shown are non-MSTP capable
RSTP operates on these links

CIST operates links between regions and to SST devices.

The role of the Common Spanning Tree (CST) in a network, and the Common and Internal Spanning Tree (CIST) configured on each device, is to prevent loops within a wider network that may span more than one MSTP Region and parts of the network running in legacy STP or RSTP mode.

MSTP Bridge Protocol Data Units (BPDU)

Its main function is enabling MSTP to select its root bridges for the proper CIST and each MSTI. MSTP includes all its spanning tree information in a single BPDU format. Not only does reduce the number of BPDUs required on a LANs to communicate spanning tree information for each VLAN, but it also ensures backward compatibility with RSTP (and in effect, classic STP too).

BPDUs' general format comprises a common generic portion *-octets 1 to 36-* that are based on those defined in IEEE Standard 802.1D,2004,[2] followed by components that are specific to CIST *-octets 37 to 102*. Components specific to each MSTI are added to this BPDUs data block.

BPDU table info (https://www.alliedtelesis.com/sites/default/files/stp_feature_config_guide.pdf) and STP BPDUs **show a deeper resume of the MSTP BPDU format** and, besides, some additional information about how was this object structured in older or different versions of this protocol as STP and RSTP, maintaining its compatibility.

MSTP Configuration Identification

In case there is an allocation of VIDs (VLAN IDs) into a MST Region which differs within the different bridges that compound it, **frames for some VIDs might be duplicated or even not delivered to some LANs at all**. To avoid this, MST Bridges check that they are allocating VIDs to the same spanning trees as their neighboring MST Bridges in the same Region by transmitting and receiving MST Configuration Identifiers along with the spanning tree information. These MST Configuration Identifiers, while compact, **are designed so that two matching identifiers have a very high probability of denoting the same configuration even in the absence of any supporting management practice for identifier allocation.** Either one of this "objects" contains the following:

- **Configuration Identifier Format Selector:** Indicates the use which is going to be given to the following components.
- **Configuration Name**[3][4][5]
- **Revision Level and the Configuration Digest:**[6][7] A 16B signature HMAC-MD5 Algorithms created from the MST Configuration Table.

This object is specific and unique of MSTP, neither STP or RSTP use it.

Protocol Operation

MSTP configures for every VLAN a single spanning tree active topology in a manner that there's at least one data route between any two end stations, eliminating data loops. It specifies various "objects" allowing out the algorithm to operate in a proper way. The different bridges in the various VLANs start advertising their own configuration to other bridges using the MST Configuration Identifier in order to allocate frames with given VIDs (VLAN ID) to any of the different MSTI. A priority vector is utilized to construct the CIST, it connects all the bridges and LANs in a Bridged LAN and ensures that paths within each region are always preferred to paths outside the Region. Besides, there is a MSTI priority vector, this one compromises the necessary information to build up a deterministic and independently manageable active topology for any given MSTI within each region.

Additionally, comparisons and calculations done by each bridge select a CIST priority vector for each Port (based on priority vectors, MST Configuration Identifiers and on an incremental Path Cost associated to each receiving port). This leads to one bridge been selected as the CIST Root of the Bridged LAN; then, a minimum cost path to the root is shifted out for each Bridge and LANs (thus preventing loops and ensuring full connectivity between VLANs). Subsequently, in each region, the bridge whose minimum cost path to the root doesn't pass through another bridge with the same MST Conf.ID will be identified as its Region's CIST Regional Root. Conversely, each Bridge whose minimum cost path to the Root is through a Bridge using the same MST Configuration Identifier is identified as being in the same MST Region as that Bridge.

In summary, MSTP encodes some additional information in its BPDU regarding region information and configuration, each of these messages conveys the spanning tree information for each instance. Each instance can be assigned several configured VLANs, frames (packets) assigned to these VLANs operate in this spanning tree instance whenever they are inside the MST region. To avoid conveying their entire VLAN to spanning tree mapping in each BPDU, bridges encode an MD5 digest of their VLAN to instance table in the MSTP BPDU. This digest is then used by other MSTP bridges, along with other administratively configured values, to determine if the neighboring bridge is in the same MST region as itself.

Port Roles

Common Internal Spanning Tree Ports

- **Root:** Provides the minimum cost path from the Bridge to the CIST Root through the Regional Root.
- **Designated:** Provides the least cost path from the attached LAN through the Bridge to the CIST Root.
- **Alternate or Backup:** Provides connectivity if other Bridges, Bridges Ports or LANs fail or are erased.

Multiple Spanning Tree Instance Ports

- **Root:** Provides the minimum cost path from the Bridge to the MSTI Regional Root.
- **Designated:** Provides the least cost path from the attached LANs through the Bridge to the Regional Root.
- **Master:** Provides connectivity from the Region to a CIST Root that lies outside the Region. The Bridge Port that is the CIST Root port for the CIST Regional Root is the Master port for all MSTI.
- **Alternate or Backup:** Provides connectivity if other Bridges, Bridges ports or LANs fail or are erased.

RSTP compatibility

MSTP is designed to be STP and RSTP compatible and interoperable without additional operational management practice, this is due to a set of measurements based on RSTP (Clause 17 of IEEE Std 802.1D, 2004 Edition) intending to provide the capability for frames assigned to different VLANs, to be transmitted along different paths within MST Regions.

Both protocols have in common various issues such as: the selection of the CIST Root Bridge (it uses the same fundamental algorithm, 17.3.1 of IEEE Std 802.1D, 2004 Edition, but with extended priority vector components within MST Regions), the selection of the MSTI Root Bridge and computation of port roles for each MSTI, the port roles used by the CIST are the same as those of STP and RSTP (with the exception of the Master Port), and the state variables associated with each port.

Into the bargain, they also share some problems as, for instance: MSTP can't protect against temporary loops caused by the inter-connection of two LANs segments by devices other than the Bridges that operate invisibly with respect to support of the Bridges' MAC Internal Sublayer Service.

For all the above, it can be concluded that MSTP is fully compatible with RSTP bridges, an MSTP BPDU can be interpreted by an RSTP bridge as an RSTP BPDU. This not only allows compatibility with RSTP bridges without configuration changes, but also causes any RSTP bridges outside of an MSTP Region to see the region as a single RSTP bridge, regardless of the number of MSTP bridges inside the region itself.

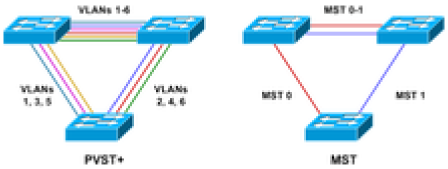
Protocol configuration

This section is mainly oriented to provide any user a proper manner of configuring a MSTP network over Cisco devices.

Before configuring MSTP

Be sure of having configured VLANs and having associated them with switch ports, afterwards determine: MSTP Regions, revision level and instances; which VLANs and switch ports will belong to which MSTIs and, finally, which devices do you want to be root bridges for each MSTI.

Configuration guidelines for MSTP



Simple network topology for MSTP trials.

1. Switches must have the same MST configuration identification elements (region name, revision level and VLAN to MSTI mapping) to be in the same MST region. When configuring multiple MST regions for MSTP, MSTIs are locally significant within an MST region. MSTIs will not span from one region to another region.
2. Common and Internal Spanning Tree (CIST) is the default spanning tree instance for MSTP. This means that all VLANs that are not explicitly configured into another MSTI are members of the CIST.
3. The software supports a single instance of the MSTP Algorithm consisting of the CIST and up to 15 MSTIs.

A VLAN can only be mapped to one MSTI or to the CIST. One VLAN mapped to multiple spanning trees is not allowed. All the VLANs are mapped to the CIST by default. Once a VLAN is mapped to a specified MSTI, it is removed from the CIST.To avoid unnecessary STP processing, a port that is attached to a LAN with no other bridges/switches attached, can be configured as an edge port.

An example of how to configure a simple, three switch MSTP topology wherein a layer-two access switch carries four VLANs and has two uplinks to two distribution switches, can be found here: MSTP Configuration Guide (http://packetlife.net/blog/2010/apr/26/multiple-spanning-tree-mst/)
A good configuration view, from the above-mentioned example shall be:

```
S3# show spanning-tree mst
```

```
##### MST0    vlans mapped:   1-19,21-39,41-4094
Bridge        address 000e.8316.f500  priority      32768 (32768 sysid 0)
Root          address 0013.c412.0f00  priority      0     (0 sysid 0)
             port    Fa0/13          path cost     0
Regional Root address 0013.c412.0f00  priority      0     (0 sysid 0)
                                     internal cost 200000    rem hops 19
Operational   hello time 2, forward delay 15, max age 20, txholdcount 6
Configured    hello time 2, forward delay 15, max age 20, max hops    20
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/13           Root FWD 200000    128.13   P2p
Fa0/16           Altn BLK 200000    128.16   P2p
##### MST1    vlans mapped:   20,40
Bridge        address 000e.8316.f500  priority      32769 (32768 sysid 1)
Root          address 000f.345f.1680  priority      1     (0 sysid 1)
             port    Fa0/16          cost          200000    rem hops 19
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/13           Altn BLK 200000    128.13   P2p
Fa0/16           Root FWD 200000    128.16   P2p
```

Extensions

Alternative Multiple Spanning Tree Protocol (AMSTP)

The first skel of this protocol was proposed in.[8] AMSTP is a simplified one tree instance rooted at each edge bridge in the core to forward frames.

Protocol operation

To set up these trees, AMSTP relies in one basic tree which will be used to obtain instances (named Alternate Multiple Spanning Tree Instances – AMSTI), until one of them is built per switch for the network. The process applied to build up the main/basic tree is the same as in RSTP. In summary, firstly a bridge must be elected as the Root Bridge (this is done by the emission of BPDUs from each switch on the network periodically, every "Hello Time", and selecting the lowest Bridge ID). Then, every switch will compute and calculate its cost to the Root Bridge and, afterwards, the root ports must be elected by selecting the one which receives the best BPDU, this is, the one that announces minimum path cost to root bridge.

BPDUs

AMSTP BPDUs use the same local multicast protocol addresses than STP and have a structure that resembles MSTP BPDUs since both are comprised essentially of a basic BPDU and several AM-Records, allowing full-backwards compatibility with RSTP and STP standard protocols. Each of the AM-Records contains the data used to negotiate a specific tree instance (AMSTI). Every ABridge, except for the elected root bridge, creates an AM-Record for its own spanning tree instances. They are used by connected ports of neighboring switches to negotiate the transitions of each tree instance with a proposal/agreement mechanism.

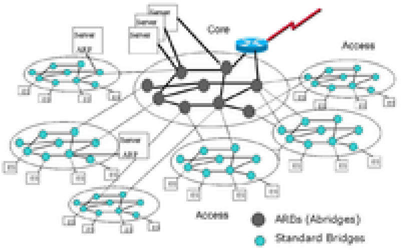
ABRIDGES

This protocol, developed in[9] emphasizes in the terms of efficiency in network usage and path length. That's the main cause why it uses AMSTP, a simplified and self-configuring version of MSTP protocol.

Abridges can be described as 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). As it has been mentioned, it is focused in terms of efficiency, this is due to the ability of AMSTP to provide optimum paths in the core mesh and the usage of RSTP to aggregate efficiently the traffic at islands networks. Its convergence speed is as fast as RSTP and MSTP.

Architecture

With the objective of enhancing the properties of Abridges protocol, a two-level hierarchical link layer infrastructure in which segmentation is performed at link layer is proposed. The core will be composed, primarily, by Abridges (Bridges using an implementation of AMSTP) and will oversee connecting the leaf access networks that are referred to as "access layer". Besides, each of this access networks, also called islands, will be a layer-two sub-network using STP connected to one or more Abridges.



Two-layer network proposal for ABridges.

Protocol operation

Inside every island or access network a bridge is automatically elected to behave as the Root Bridge, this one bridge will behave as a gateway, allowing the forwarding of frames from the core to an island and conversely. Just one Abridge is going to perform these gateway functions, although many could be connected. 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 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 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.

ABridges functionality

ABridges is composed by three basic functional modules, which could be resumed in:

- **STD Bridge:** Performs standard bridging functions with the nodes of its island. The access functionality resides on the access ports of this module, which has an equivalent behavior to a standard bridge acting as a root bridge.
- **AMSTP Routing:** Routes frames between Abridges and the Gateway. It has core ports, either of them interconnect ABridges, which learn root bridge IDs from the AMSTP BPDUs received and store this information in a database, known as "Forwarding Database".
- **GateWay:** Interconnects the above-mentioned modules.

Abridges will configure each of their ports to be part either of the core or of an island, this port self-configuration is done with very simple stipulations: if a port is not connected to another Abridge using a point-to-point link, it will turn itself an access port; on the other hand, ports directly connected to another Abridge are configured as core ports. This auto-configuration mechanism is pretty like the one used in RSTP.

ARP and ABridge resolution

As any layer-two based protocol, ABridges uses ARP broadcasts to obtain the link layer address associated to an IP address at the same LAN or VLAN. That is the main cause why avoiding overflooding is a matter of paramount priority; to limit this broadcast traffic, is recommended the use of distributed load ARP servers, although its use is not compulsory.

See also

- Bridge Protocol Data Unit
- Distributed minimum spanning tree
- EtherChannel
- Ethernet Automatic Protection Switching
- Flex links
- Media Redundancy Protocol
- Minimum spanning tree
- TRILL (Transparent Interconnection of Lots of Links)
- Unidirectional Link Detection
- Virtual Link Trunking

References

1. packard, Hewlett (2006). *Multiple Instance Spanning-Tree Operation* (ftp://ftp.hp.com/pub/networking/software/2900-AdvTrafficMgmt-Aug2006-59916197-Chap04.pdf) (PDF).
2. IEEE, Standard (2004). *IEEE Standard for Local and metropolitan area networks, Media Access Control (MAC) Bridges* (http://www.ccna-powertraining.de/wp-content/uploads/2014/10/802.1D-2004.pdf) (PDF). IEEE Computer Society.
3. IETF, RFC (1998). *RFC 2271 SnmpAdminString object* (https://tools.ietf.org/html/rfc2271). IETF, D. Harrington.
4. IETF, RFC (1999). *RFC 2571 SnmpAdminString object* (https://tools.ietf.org/html/rfc2571). IETF, D. Harrington.
5. IETF, RFC (2002). *RFC 3411 SnmpAdminString object* (https://tools.ietf.org/html/rfc3411). IETF, D. Harrington.
6. IETF, RFC (1997). *HMAC: Keyed-Hashing for Message Authentication* (https://tools.ietf.org/html/rfc2104). IETF, H. Krawczyk.
7. IETF, RFC (2011). *Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms* (https://tools.ietf.org/html/rfc2104). IETF, S. Turner.
8. Ibáñez, García, Azcorra, Guillermo, Alberto, Arturo (2002). *Alternative Multiple Spanning Tree Protocol (AMSTP) for Optical Ethernet Backbones* (http://e-archivo.uc3m.es/bitstream/handle/10016/2791/amstp-2004.pdf?sequence=1) (PDF). IEEE Computer Society.
9. Ibáñez, García, Azcorra, Soto, Guillermo, Alberto, Arturo, Ignacio (2007). *Alternative Multiple Spanning Tree Protocol (AMSTP) for Optical Ethernet Backbones* (https://e-archivo.uc3m.es/bitstream/handle/10016/2954/COMPNW_3675_08.pdf?sequence=2&isAllowed=y) (PDF). Departamento de Ingeniería Telemática, Universidad Carlos III, Madrid, Spain, CAPITAL MEC Project.

External links

- IEEE "Home Page" for 802.1 (https://1.ieee802.org/) (Related Standards of the 802.1 family)
- MSTP Tutorial (http://blog.ine.com/2008/07/27/mstp-tutorial-part-i-inside-a-region/) (Brief Tutorial for the comprehension of MSTP)
- RBridge (http://www.postel.org/pipermail/rbridge/)
- Cisco Implementations
  - [1] (https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/24248-147.html) (Cisco Implementation and brief tutorial about MSTP)
  - Cisco home page for the Spanning-Tree protocol family (http://www.cisco.com/en/US/tech/tk389/tk621/tsd_technology_support_protocol_home.html) (discusses CST, MISTP, PVST, PVST+, RSTP, STP)
  - Educational explanation of STP (http://www.cisco.com/image/gif/paws/10556/spanning_tree1.swf) www.cisco.com
- Perlman, Radia. "Algorhyme" (https://web.archive.org/web/20110719212324/http://www.csua.berkeley.edu/~ranga/humor/algorhyme.txt). University of California at Berkeley. Archived from the original (http://www.csua.berkeley.edu/~ranga/humor/algorhyme.txt) on 2011-07-19. Retrieved 2011-09-01.
- IEEE Standards
  - ANSI/IEEE 802.1D-2004 standard (http://standards.ieee.org/getieee802/download/802.1D-2004.pdf), section 17 discusses RSTP (Regular STP is no longer a part of this standard. This is pointed out in section 8.)

  - ANSI/IEEE 802.1Q-2005 standard (http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf), section 13 discusses MSTP
- RFCs
  - RFC 2271-1998, - An Architecture for Describing SNMP Management Frameworks
  - RFC 2571-1999, - An Architecture for Describing SNMP Management Frameworks
  - RFC 2674-1999,- Proposed standard, Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions
  - RFC 1525-1993, - SBRIDGEMIB, proposed standard, Definitions of Managed Objects for Source Routing Bridges
  - RFC 1493-1993 - BRIDGEMIB, draft standard, Definitions of Managed Objects for Bridges
  - ABridge Standard (https://tools.ietf.org/html/draft-gibanez-trill-abridge-01)