802.1AS Fast Master Clock Selection

Moving 802.1AS closer to RSTP

Version 2

Norman Finn

Cisco Systems

Introduction

Introduction

- IEEE 1588 networks that contain transparent clocks and the current draft of P802.1AS both have Announce Messages that:
  - Elect the clock that will drive the network's timing;
  - Propagate over an underlying data transport network, a spanning tree in the case of Layer 2; and
  - Introduce timeouts to let the master election settle.
- The result is that the convergence of the master clock election and clock distribution tree formation:
  - Depends on a pre-existing data forwarding topology;
  - Is slower than the alternative presented in this slide deck; and
  - Is conceptually more complex than the alternative presented, here.

Rapid Spanning Tree Protocol Basics

Rapid Spanning Tree Protocol (RSTP) Basics

- Networks consist of Bridges that have Ports attaching them to LANs.

  A LAN can be attached to two Ports (point-to-point medium) or more than two Ports (shared medium).

- Each Bridge can transmit Bridge Protocol Data Units (BPDUs) on Ports. A BPDU says, “This is the state of RSTP on this Port.”

  That state is different on each Port.

- Every Bridge considers itself either the Root Bridge or not the Root Bridge.

Applying RSTP to 802.1AS

- The most tricky bits of RSTP are concerned with ensuring that the data plane, which can operate independently of the control plane, never forwards frames in a closed loop, barring malfunctions that cripple the algorithm.

  The tricky bits also make RSTP “rapid” compared to the old STP.

  Those tricky bits are what cause RSTP to fail catastrophically when algorithm malfunctions do occur.

- But, 802.1AS has no such independent data plane, so those tricky bits are not needed for clock distribution.
- Therefore, these slides present only the (simpler) bits that are needed by 802.1AS.

RSTP Port Roles

- Every Port on a Bridge takes one of four roles:

  The **Root Port**;

  An **Alternate Port**;

  A **Designated Port**; or

  A **Backup Port**.

Port Roles: Root Port

- The one Port closest to the Root Bridge. This Bridge:

  Expects to receive a regular stream of BPDUs on this Port from the Bridge closer to the Root Bridge.

  Will modify and propagate the information received in these BPDUs messages to the rest of the network through this Bridge's Designated Ports.

  The Root Bridge has no Root Port. Each Non-Root Bridge has exactly one Root Port.

Port Roles: Alternate Port

- Any Port that is connected to a Bridge that is closer to the Root than this Bridge, but is not the Root Port. This Bridge:

  Expects to receive a regular stream of BPDUs on this Alternate Port from the LAN's Designated Port.

  Will **not** propagate the information received in the BPDUs.

  Can instantly transform the best of the Alternate Ports into the Root Port, if the Root Port fails.

Port Roles: Designated Port

- A Port that has the best claim, via BPDUs, to being the closest Port to the Root Bridge among all of the Ports connected to the same LAN as the Designated Port.

  A Port that is not connected to another Bridge is always a Designated Port.

  All of the other Bridges' Ports connected to that same LAN are either Root Ports or Alternate Ports.

  Every Port on the Root Bridge is either a Designated Port or a Backup Port.

- On a Designated Port, this Bridge will:

  Transmit a regular stream of BPDUs that propagate timing information to the rest of the network.

Port Roles: Backup Port

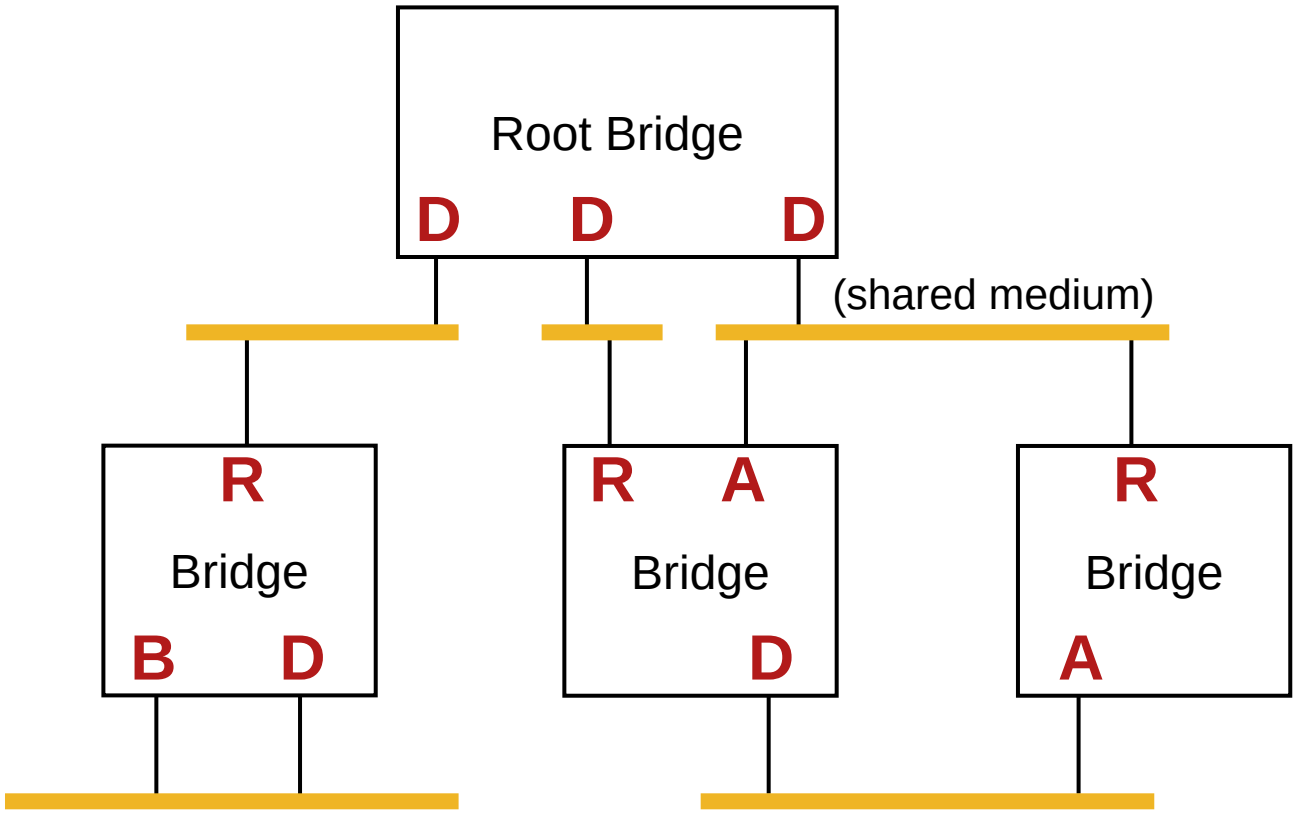
- A Port on a Bridge that is connected to a LAN that is connected to a Designated Port on the same Bridge. This Bridge:

  Expects to receive a regular stream of BPDUs on this Port from the Designated Bridge.

  Will **not** propagate the information received in the BPDUs.

- Every Port on the Root Bridge is either a Designated Port or a Backup Port.

Port Roles



- **R**oot Port, **D**esignated Port, **A**lternate Port, **B**ackup Port

RSTP BPDU: Comparing BPDUs

- **On each LAN, there is (eventually) exactly one Designated Port.**

  The Bridge to which that Port belongs is the Designated Bridge for this LAN.

  Only the LAN's Designated Bridge transmits BPDUs; the other Bridge(s) just listen.

  When a LAN comes up, every Bridge attached to it assumes that it is the Designated Bridge.

  The first Bridge to transmit a BPDU either really is the Designated Bridge, or the other Bridge(s) will respond with their BPDUs.

  Very quickly, all agree on which is the Designated Bridge.

  (The algorithm works even if agreement is delayed.)

RSTP BPDUs: What does "best" mean?

- To compare claims, the fields in a BPDU (or variables in memory) are concatenated into a "priority vector" that is treated as a very long binary number.

  The smallest numerical value wins.

  Since all priority vectors are the same length and are unsigned, there is no difference between lexical and numerical comparisons.

- Different combinations of fields, and thus different vectors, are used for different computations.

RSTP BPDUs: The four main fields

- **Root ID:** The globally unique ID of the Bridge that this Bridge thinks is the Root Bridge.
- **Root Path Cost:** The total cost from this Bridge to the Root Bridge, where the cost of each hop is inversely proportional to the link speed.
- **Bridge ID:** The globally unique ID of this Bridge. (Same as Root ID in a BPDU sent by the Root Bridge.)
- **Port ID:** Uniquely identifies the Port on which the BPDU was sent among all Ports with the same Bridge ID.

RSTP: When I receive a BPDU ...

- **Step 1: New Root Bridge?**

Compare received:	To my:
Root ID	Root ID from Root Port (my Bridge ID if I think I'm Root)

- If received Root ID is better, replace my information with his.

  This is now the Root Port.

  Update all of the Ports' information.

RSTP: When I receive a BPDU ...

- **Step 2: Who is the Designated Bridge on this LAN?**

Signifi-cance	Compare received:	To my:
most	Root ID	Root ID from Root Port (my Bridge ID if I think I'm Root)
...	Root Path Cost	Root Port's Root Path Cost (0 if I think I'm Root)
...	Bridge ID	My Bridge ID
least	Port ID	Port ID of this Port

- If I'm Designated, ignore his information.

RSTP: When I receive a BPDU ...

- **Designated Bridge determination:**

Root ID	Whose information is derived from the best Root? If equal ...
Root Path Cost	Who is closest to that Root? If equal ...
Bridge ID	Who is configured best (high bytes of ID) or has the lowest address? If equal ...
Port ID	(I'm listening to myself.) Which Port has the lowest priority or lowest address?

- If I win, this is a Designated Port.

RSTP: When I receive a BPDU ...

- **Step 3: Which is the Root Port? Compare all non-Designated Ports':**

BPDU Root ID	Which Port's information is derived from the best Root? If equal ...
BPDU Root Path Cost + this Port's Cost	Which is closest to that Root, including the receiving Costs? If equal ...
BPDU Bridge ID	Which Designated Bridge is configured best (high bytes of ID) or has the lowest address? If equal ...
BPDU Port ID	(Two Ports are listening to the same other Bridge.) Which Port has the lowest priority or lowest address?

What RSTP fields matter?

Protocol Identifier (2)	Bridge Identifier (8)
Protocol Version Identifier (1)	Port Identifier (2)
BPDU Type (1)	Message Age (2)
Flags (1)	Max Age (2)
Root Identifier (8)	Hello Time (2)
Root Path Cost (4)	Forward Delay (2)

- Ignore the dull bits and data plane interlock bits.
- We've seen the green fields.
- What are these fields?

RSTP BPDU: What fields are left?

- **Message Age:** How many hops has this information made since leaving the Root Bridge?
- **Max Age:** After how many hops should this information be discarded?
- **Hello Time:** How often does the Designated Port send BPDUs?

RSTP BPDU: Message Age and Max Age

- The Root Bridge's configured Max Age is spread throughout the network via the BPDUs, and determines how many hops the information can travel. Message Age is incremented at each hop.
- The **good news** is that this is a simple technique that ensures the algorithm will converge. Unless ...
- The **bad news** is that, if the network is larger than Max Age hops, the network will not converge.
- When this happens, the outlying areas may not be connected to each other, and each island will use the best Root it can find.

RSTP BPDU: Hello Time

- Each Designated Bridge announces the rate at which it intends to transmit BPDUs.
- If the receiving Bridge misses three BPDUs, it figures that the Designated Bridge is dead, and proceeds as if it never received a BPDU on that Port.

  If not the Root Port, that's easy.

  If it's the Root Port, then another Root Port must be selected from among the Alternate Ports. If there are no Alternate Ports, then this Bridge becomes the Root Bridge (until corrected).

RSTP: Getting Started

- A Bridge simply considers itself to be the Root Bridge, and starts running.
- If it’s wrong, its neighbors will quickly inform it.

Restating RSTP using 802.1AS terminology

802.1AS (Fast Convergence) Basics

- Networks consist of Bridges that have Ports attaching them to LANs.

  A LAN can be attached to two Ports (point-to-point medium) or more than two Ports (shared medium).

- Every Bridge has the capability of being a Grandmaster Clock.
- Every Bridge can transmit Announce Messages on Ports. An Announce Message says, “This is the state of 802.1AS on this Port.”

  That state is different on each Port.

- Every Bridge considers itself either the Grandmaster Clock or not the Grandmaster Clock.

802.1AS Port Roles

- Every Port on a Bridge takes one of four roles:

The **Slave Port**;	(Root Port in RSTP)
A **Passive Port**;	(Alternate Port in RSTP)
A **Master Port**;	(Designated Port in RSTP)
A **Master (Idle) Port**; or	(Designated Port in RSTP)
A **Backup Port**.	(Backup Port in RSTP)

Port Roles: Slave Port

- The one Port closest to the Grandmaster Clock. This Bridge:

  Expects to receive a regular stream of Sync, Followup and Announce Messages on this Port from the Bridge closer to the Grandmaster Clock.

  Expects to exchange a regular stream of Pdelay Messages with that Bridge, and to respond to them.

  Will modify and propagate the information received in the Sync, Followup and Announce Messages to the rest of the network through this Bridge's Master Ports.

- The Grandmaster Clock has no Slave Port. Each Non-Grandmaster Clock has exactly one Slave Port.

Port Roles: Passive Port

- Any Port that is connected to a Bridge that is closer to the Root than this Bridge, but is not the Root Port. This Bridge:

  Expects to receive a regular stream of Sync, Followup, and Announce Messages on this Passive Port from the LAN's Master Port.

  Expects to exchange a regular stream of Pdelay Messages with that Bridge, and to respond to them.

  Will **not** propagate the information received in the Sync, Followup, and Announce Messages.

  Can instantly transform the best of the Passive Ports into the Slave Port, if the Slave Port fails.

Port Roles: Master Port

- A Port that has the best claim, via Announce Messages, to being the closest Port to the Grandmaster Clock among all of the Ports connected to the same LAN as the Master Port.

  All of the other Bridges' Ports connected to that same LAN are either Slave Ports or Passive Ports.

  Every Port on the Root Bridge is either a Master Port or a Backup Port.

- On a Master Port, this Bridge will:

  Transmit a regular stream of Sync, Followup, and Announce Messages that propagate synchronization information to the rest of the network.

  Exchange a regular stream of Pdelay Messages with the other Bridge, and to respond to them.

Port Roles: Master (Idle) Port

- A Master Port becomes a Master (Idle) Port if no responses are received from its Pdelay Messages.
- On a Master (Idle) Port, a Bridge:

  Sends regular Announce and Pdelay Messages.

  Does not sent Sync or Followup messages.

Port Roles: Backup Port

- A Port on a Bridge that is connected to a LAN that is connected to a Master Port on the same Bridge. This Bridge:

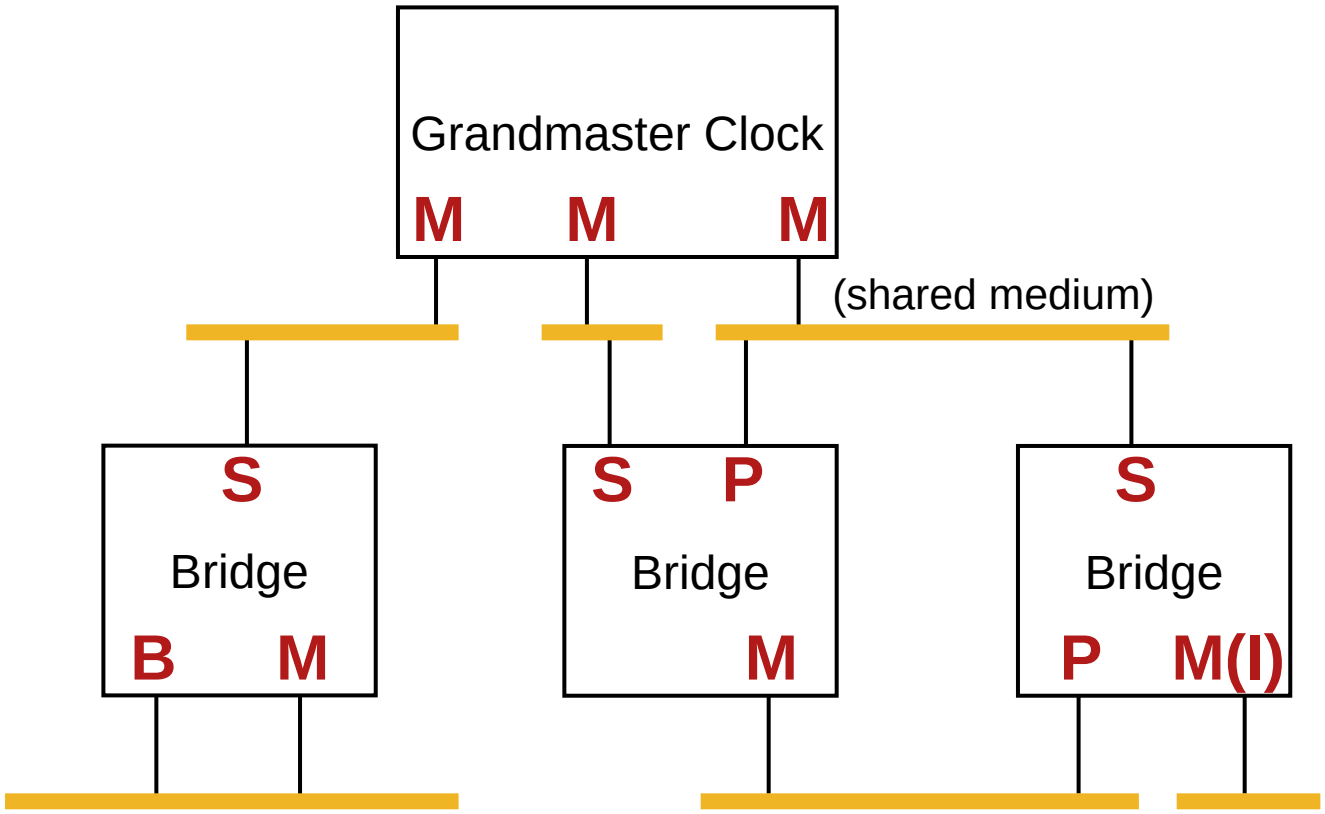
  Expects to receive a regular stream of Announce Messages on this Port from the Master Bridge, but not Syncs or Followups.

  May or may not (TBD) receive Sync, Followup and/or Pdelay Messages from the Master Bridge.

  Will **not** propagate the information received in the Announce Messages.

- Every Port on the Grandmaster Clock is either a Master Port or a Backup Port.

Port Roles



- **S**lave Port, **M**aster (**I**dle) Port, **P**assive Port, **B**ackup Port.

802.1AS Announce: Comparing claims

- **On each LAN, there is (eventually) exactly one Master Port.**

  The Bridge to which that Port belongs is the Master Bridge for this LAN.

  Only the LAN's Master Bridge transmits Sync, Followup, or Announce Messages; the other Bridge(s) just listen.

  When a LAN comes up, every Bridge attached to it assumes that it is the Master Bridge.

  The first Bridge to transmit an Announce Message either really is the Master Bridge, or the other Bridge(s) will respond with their Announce Messages.

  Very quickly, all agree on which is the Master Bridge.

  (The algorithm works even if agreement is delayed.)

802.1AS Announce Messages: What does "best" mean?

- To compare claims, the fields in an Announce Message (or variables in memory) are concatenated into a "priority vector" that is treated as a very long binary number.

  The smallest numerical value wins.

  Since all priority vectors are the same length and are unsigned, there is no difference between lexical and numerical comparisons.

- Different combinations of fields, and thus different vectors, are used for different computations.

802.1AS Announce Messages: The four main fields

- **grandmasterIdX:** The globally unique ID of the Bridge that this Bridge thinks is the Grandmaster Clock.
- **GrandmasterPathCost:** The total cost from this Bridge to the Grandmaster Clock, where the cost of each hop is proportional to the accumulated propagation error.
- **ClockIdX:** The globally unique ID of this Bridge. (Same as grandmasterIdX in an Announce Message sent by the Grandmaster Clock.)
- **PortID:** Uniquely identifies the Port on which the Announce Message was sent among all Ports with the same ClockIdX.

Announce Fields

- **grandmasterIdX:** Same function as RSTP Root Identifier. Consists of:

  1 byte (or less) of priority1: Just like the high bytes of an RSTP Bridge ID. This supports absolutely forcing the Grandmaster Clock selection.

  1 bytes (or less) of clock quality: In the absence of priority1 configuration, this causes the best clock to be selected as Grandmaster Clock.

  1 byte (or less) of priority2: This supports forcing a Grandmaster Clock selection among equal quality clocks.

  2 bytes (or more) of virtual entity number: Just like the high bytes of an RSTP Bridge ID. This provides 4k or more IDs for virtual devices from a single physical address. (This can be thought of as extra priority2 bits, but selected by the implementation, not administered.)

  8 bytes of MAC address: 1588 supports EUI-64 addresses, so this protocol should, also.

Announce Fields

- **GrandmasterPathCost:** Same function as RSTP Root Path Cost.

  Allows comparison of two paths to the Grandmaster Clock, so that the path that introduces the least inaccuracies can be chosen.

  In RSTP, the configured Port Cost is added to the Root/Alternate Port's Root Path Cost, and that value is distributed to all other Bridges. Thus, only the BPDU receiver adds Port Cost.

  For 802.1AS, each end of the link contributes some inaccuracy, perhaps due to implementation choices. Therefore, the Master Port adds the Root-to-Designated-across-the-Bridge Cost and the Transmit Cost, and the Slave/Passive Port adds the Receive Cost.

  If we believe that the probable error after n hops is the sum of the n hops' errors, that's fine. We might believe instead that the probable error is the root-mean-square of the hop errors, so the Cost parameter would be a sum of squares of the individual Costs. (There is no need to take the square root to compare relative Costs.)

Announce Fields

- **ClockIdX:** Same function as RSTP Bridge Identifier.

  Same format as grandmasterIdX.

  If I'm the Grandmaster Clock, this becomes the grandmasterIdX.

- **PortID:** 1588 portIdentity is a subset of RSTP Port Identifier, works like RSTP Port Identifier

  1588 provides a 2-byte PortID.

  RSTP provides a 4-bit configurable priority field, so that the Backup vs. Designated choice can be configured, followed by a 12-bit Port ID.

  The RSTP format is probably more useful, but this should be discussed.

Announce Fields

- **Message Age:** Same function as RSTP Message Age.

  It is now a simple hop count.

- **Max Age:** Same function as RSTP Max Age.

- **Clock ID List:** Possible improvement.

  The Message Age and Max Age could be augmented by including a list of ClockIdXs traversed along the path taken by this Grandmaster Clock information.

  The whole ClockIdX does not need to be stored; only the virtual entity number and the MAC address are required.

- Including all three of these fields allows absolute minimum convergence time, while bounding the frame size.

802.1AS:
When I receive an Announce Message...

- **Step 1: New Grandmaster Clock?**

Compare received:	To my:
grandmasterIdX	grandmasterIdX from Slave Port (my ClockIdX if I think I'm Grandmaster Clock)

- If received grandmasterIdX is better, replace my information with his.

  This is now the Slave Port.

  Update all of the Ports' information.

802.1AS:
When I receive an Announce Message...

- **Step 2: Who is the Master Bridge on this LAN?**

Signifi-cance	Compare received:	To my:
most	grandmasterIdX	grandmasterIdX from Slave Port (my Bridge ID if I think I'm MC)
...	GrandmasterPathCost	Slave Port's MC Path Cost (0 if I I'm Grandmaster Clock)
...	ClockIdX	My ClockIdX
least	PortID	PortID of this Port

- If I'm Master, ignore his information.

802.1AS:
When I receive an Announce Message...

- **Master Bridge determination:**

Grandmaster Clock Root ID	Whose information is derived from the best Grandmaster Clock? If equal ...
GrandmasterPathCost	Who is closest to that Grandmaster? If equal ...
ClockIdX	Who is configured best or has the lowest address? If equal ...
PortID	(I'm listening to myself.) Which Port has the lowest priority or lowest address?

- If I win, this is a Master Port.

802.1AS:
When I receive an Announce Message...

- **Step 3: Which is the Slave Port? Compare all non-Master Ports':**

Announce grandmasterIdX	Which Port's information is derived from the best Grandmaster Clock? If equal ...
Announce MC Path Cost + this Port's Cost	Which is closest to Grandmaster Clock, including the receiving Costs? If equal ...
Announce ClockIdX	Which Master Bridge is configured best (high bytes of ID) or has the lowest address? If equal ...
Announce PortID	(Two Ports are listening to the same other Bridge.) Which Port has the lowest priority or lowest address?

802.1AS: What fields are left?

- **domainNumber:** Identifies the reach of this protocol.
- **Message Age:** How many hops has this information made since leaving the Grandmaster Clock?
- **Max Age:** After how many hops should this information be discarded?
- **logMeanMessageInterval:** How often does the Master Port send Announce Messages?

802.1AS BPDU: Message Age and Max Age

- The Grandmaster Clock's configured Max Age is spread throughout the network via the Announce Messages, and determines how many hops the information can travel. Message Age is incremented at each hop.
- The **good news** is that this is a simple technique that ensures the algorithm will converge. Unless ...
- The **bad news** is that, if the network is larger than Max Age hops, the network will not converge.
- When this happens, the outlying areas may not be connected to each other, and each island will use the best Grandmaster Clock it can find.

802.1AS: MMI and Domain No.

- **logMeanMessageInterval:** Each Master Bridge announces the rate at which it intends to transmit Announce Messages.
- If the receiving Bridge misses three Announce Messages, it figures that the Master Bridge is dead, and proceeds as if it never received an Announce Message on that Port.

  If not the Slave Port, that's easy.

  If it's the Slave Port, then another Slave Port must be selected from among the Passive Ports. If there are no Passive Ports, then this Bridge becomes the Grandmaster Clock (until corrected).

- **domainNumber:** Identifies the reach of this protocol. Not in RSTP.

  Bridge ignores Announce Messages from a different domain.

  This could be expanded to a name + version number, like MSTP.

802.1AS: An alternative to hop count

- A Bridge could also record a chain of Bridge IDs from the Grandmaster Clock, with each Bridge adding its own ClockIdX to the list, and discarding any Announce Message containing its own ClockIdX.
- The **good news** is that this is a simple technique that ensures the algorithm will converge. Unless ...
- The **bad news** is that the frame grows and grows, and if it fills up, the algorithm will not fail.
- When this happens, the outlying areas may not be connected to each other, and each will use the best Grandmaster Clock it can find.

802.1AS: Getting Started

- A Bridge simply considers itself to be the Grandmaster Clock, and starts running.
- If it's wrong, its neighbors will quickly inform it.
- Every Bridge has a clock and can be the Grandmaster Clock!

  It might be a lousy clock – an integer count of times through the scheduler – but, it's got a clock.
- There is no such thing as a Station!

  There is only a Bridge that happens to have just one Port.

  That one Port is either a Master Port (I'm the Grandmaster Clock) or it's the Slave Port (I'm not the Grandmaster Clock).

802.1AS: Time until convergence?

- **This scheme reacts as fast as the information can propagate. There are no timers!**
- Every link has a Master Bridge, and that Bridge runs the Sync and followup on that link, whether the Grandmaster Clock information is propagated (the other end of the link is a Slave Port) or not (the other end is a Passive Port).

802.1AS: Time until convergence?

- **This scheme reacts as fast as the information can propagate. There are no timers!**
- Having said that, we may introduce a safety timer that limits Announce messages, just in case a bad implementation prevents convergence.

Combining message types.

Keep the Announce Message separate?

- We could keep the Announce Message separate from the other messages, as they exist in the current 802.1AS draft.

  Grandmaster Clock selection and propagation would still be more responsive than 1588.

  One could more easily argue that 802.1AS is derived properly from 1588, than if the Announce is combined with another message type, i.e., Followup.

  Announce Messages can be lost, so a Bridge must be ready to ignore other messages that are not expected, and we must ensure that the fields are in the messages to allow us to do that.

Combine the Announce Message with sync?

- We could combine the Announce Message with any or all Sync Messages from a Designated Port.

  This is one step further removed from 1588.

  This results in the receiver learning about the best Grandmaster Clock at the same moment it gets a Sync based on that clock.

  But, we know that the Syncs will often be handled by hardware. We don't want to take a chance that someone cannot handle the additional information in a Sync.

Combine the Announce Message with Pdelay?

- We could combine the Announce Message with the Pdelay_Req and/or other Pdelay Messages.

  (Not Pdelay_Resp – the Announce isn't sent in that direction.)

  But both sides of the link send Pdelay_Req, and both respond, so this may result in extra Announce Messages being processed.

  This is, again, one step further removed from 1588 than a separate Announce message.

Combine the Announce Message with Followup?

- We could combine the Announce Message with the Followup Messages.

  Followup Messages are sent by Master Ports, just like we want the Announce Messages to be sent.

  This is, again, one step further removed from 1588 than a separate Announce message.

  But, one can argue that we are not eliminating the Announce Message – we are simply packing the Announce Message and the Followup Message into a single frame, and that said packing allows us to use a single header for both.

Do Followups happen too often?

- If necessary, we can reduce the amount of computation required for the spanning tree even further if we:

  Attach an “InfoRevision” field to each Announce message.

  InfoRevision is incremented each time the information sent in a given Port’s Announce message changes.

  The “spanning tree” process is awakened only when a received InfoRevision doesn’t match the last-receive value.

  InfoRevision is initialized to a random value each time a device reboots. (Not the same value each time!)

  InfoRevision is incremented, whether the information changed or not, every one or a few seconds, just in case the receiver rebooted, or in case the random value collided with the value in use before the reboot of the sender.

Do Followups happen too often?

- We can also look at adding the Announce Message to only some of the Followup Messages.
- Further study is needed to decide whether we can eliminate the separate frame carrying just the Announce Message.

  There are tradeoffs both ways.

Summary

Summary

- Protocol simplification: Every node that participates in the protocol is an n-Port Clock. n=1 for a station.
- Making the Announce message work like RSTP BPDUs makes both Grandmaster Clock selection and recovery from topology changes faster, because there are no more timeouts, and there is no underlying data transport topology that must converge, first.
- Tying a GrandmasterPathCost algorithm to clock propagation accuracy improves the clocks' accuracy.
- Combining the new Announce message with another message, probably Followup, may further simplify and speed up the protocol.
- We want the result to still fit under the 1588 umbrella.