NetTimeLogic GMBH

Introduction to HSR&PRP

HSR&PRP Basics

Content

- What are HSR&PRP?
- Why HSR&PRP?
- History
- How it works
- HSR vs PRP
- HSR&PRP with PTP

What are HSR&PRP?

- **High Availability Seamless Redundancy (HSR) standardized in IEC62439-3 Chapter 5**
- **Parallel Redundancy Protocol (PRP) standardized in IEC62439-3 Chapter 4**
- **Redundancy protocols for automation networks**
  - They describe how network redundancy can be achieved and how the redundancy can be monitored
- **Both standards provide zero recovery time**

Why HSR&PRP?

- **Why HSR or PRP and not another protocol**
  - Why better than (R)STP?
    - Does not provide seamless redundancy
    - Switchover times not deterministic
    - Recovery also comes with a down time
  - Why better than MRP/CRP/BRP/DRP/RRP?
    - Do not provide seamless redundancy
    - Recovery also comes with a down time

Why HSR&PRP?

- **But:**
  - HSR requires hardware assistance
  - Is designed for engineered environments, e.g. all devices shall support HSR
  - PRP requires a duplication of the Network

Why HSR&PRP?

- **Optimal protocol depends on many factors:**
  - Degree of redundancy (how many failures)
  - Switchover delay
  - Reintegration delay
  - Repair strategy
  - Supervision
  - Consequences of failure
  - Economic costs of redundancy
  - Expected failure rate

Why HSR&PRP?

NetTimeLogic GMBH

- **Where is zero recovery required?**
- **Recovery requirements:**
  - Uncritical: < 10 s (not real time)
    - Enterprise Resource Planning, Manufacturing Execution
  - Automation general: < 1 s (soft real time)
    - Human interface, SCADA, building automation
  - Benign: < 100 ms (real time)
    - process & manufacturing industry, power plants,
  - Critical: < 10 ms (hard real time)
    - synchronized drives, robot control, substations

Why HSR&PRP?

NetTimeLogic GMBH

- **In Power applications:**
  - In substations a general recovery time of <5 ms is required:
    - IEC 61850-8-1 (protection trip): < 8ms
    - IEC 61850-9-2 (bus bar protection): < 1ms
    - IEC 61850-9-2 (sample values): < 2 samples = < 500 us
  - This is not possible with the other protocols!
- **IEC61850 selected HSR&PRP as the redundancy protocols**
  - RSTP as well but only on non critical paths

History and Future

- **First version IEC62439-3:2010**
  - Approved 25.02.2010
  - Not widely deployed
- **Second version IEC62439-3:2012**
  - Approved 05.07.2012
  - Widely deployed, incompatible with first version
  - Also defines how PTP shall work with HSR&PRP
- **Third version IEC62439-3:2016**
  - Approved 31.03.2016
  - Widely deployed, compatible with second version
  - Basically only corrections

How does it work?

NetTimeLogic GMBH

- **Both protocols define very similar things in regards on how zero recovery time shall be achieved:**
  - Duplicate frames
  - Tag frames
  - Send frames over two paths in the network
  - Receive first duplicate and drop second
  - Untag frames
  - Send periodically Supervision frames
- **Both can handle only single point of failures**

How does it work?

- **A HSR&PRP node normally has three ports**
  - Port A&B:
    - The redundant port pair
  - Port C:
    - Uplink port duplicating frames and removing duplicates
    - Can be internally or external

How does it work?

- **In case of a broken path the first frame will be always received via the still working path**
- **No reconfiguration needed**
  - No path has to be explicitly chosen
  - Just take the first valid frame that arrives
  - Also during normal operation frames can switch paths all the time if the delays of the two paths are similar but varying or if frames corrupt on some of the paths

How does it work?

- **Why tagging?**
  - Uniquely identify a frame even when the payload stays exactly the same
- **Tags contain the following:**
  - A 16bit identifier to detect the tag
  - A 12bit length field representing the payload length
  - A 4bit LAN identifier which defines over which path the frame was sent
  - A 16bit sequence number to identify duplicates
- **A frame is uniquely identified by «SrcMAC» & «SeqNr» & «LanId»**
  - Except wraparound of the sequence number

How does it work?

- **Each node increments the «SeqNr» of the tag by one for every frame it duplicates and tags on sending from Port C**
  - A duplicate pair always have the same sequence number
  - The two frames only differ on the LAN identifier and CRC
  - The 16bit sequence number limits the maximum number of frames that are allowed in a network and the maximum delay on reception between the paths
  - Redundancy is transparent for the host, the HSR&PRP core shall insert the tags and do the duplication of frames

How does it work?

NetTimeLogic GMBH

- **If a node which receives a tagged frame from Port A&B which is forwarded to Port C**
  - The duplicate and the tag shall be removed on reception (for HSR only)
  - Redundancy is transparent for the host, the HSR&PRP core shall remove the tags and do the discarding of duplicate frames

- **HSR&PRP tags are not identical**
- **HSR&PRP tags are inserted in different positions of a frame**

How does it work?

- **Why supervision?**
  - Redundancy without supervision is useless
  - If no error is detected it will not be fixed and things will eventually fail
- **HSR&PRP sends periodically so called «Supervision» frames**
  - Every node sees every other nodes Supervision frames and can detect failures
  - Every node can determine the logical network topology
- **HSR&PRP «Supervision» frames are not identical**

How does it work?

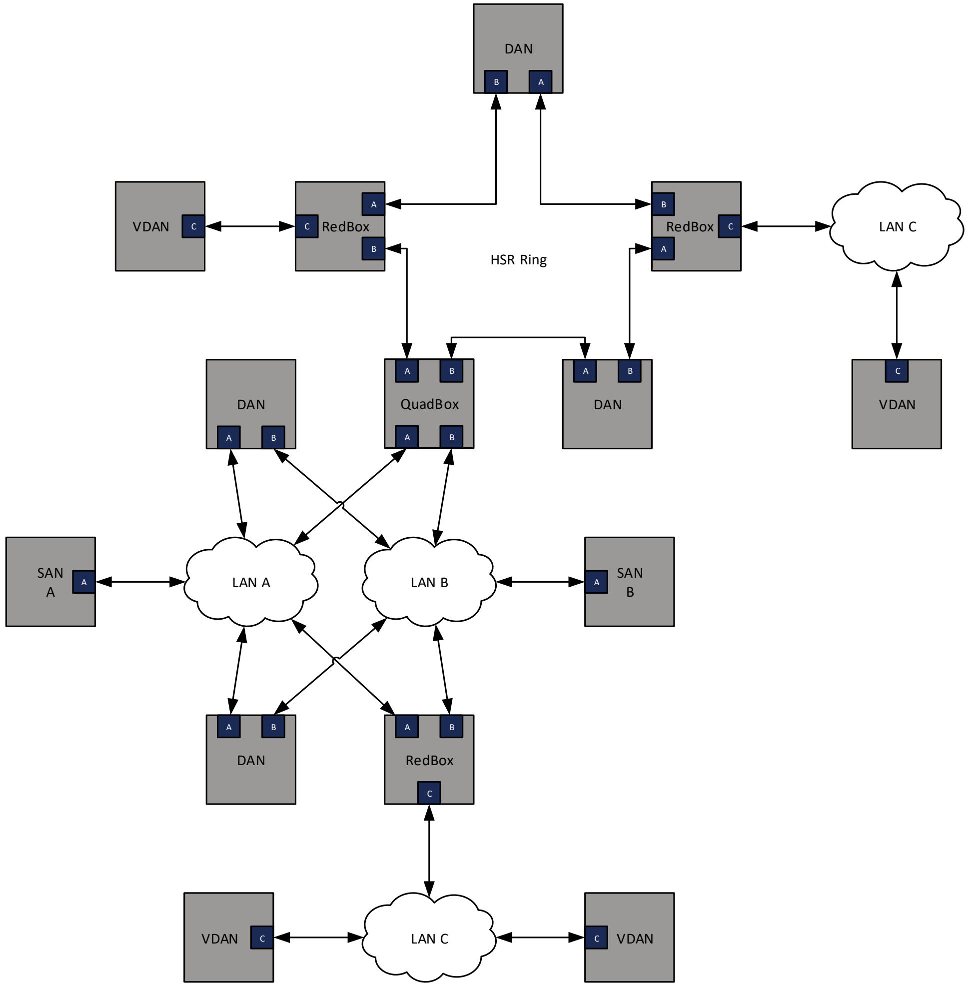
- **The standard defines how nodes communicate, where and what tags shall be inserted or removed, how duplicates can be identified and how supervision shall be done**
- **The standard says nothing about how to implement the discarding of duplicates**
  - With a hash table?
  - With a huge list of entries?
  - With node tables?
  - Etc.
- **The rule is: if in doubt, forward the duplicate!**
  - Never drop a frame by accident but allow duplicates

How does it work?
What is the difference then?

- **HSR**
  - Uses a ring topology
  - Forwarding between Ports A&B
  - Tag at the beginning of a frame
  - All nodes must support HSR
- **PRP**
  - Uses two independent parallel networks of any topology
  - No forwarding between Ports A&B
  - Tag at the end of a frame

How does it work?
Node Types

- DAN
- SAN
- RedBox
- VDAN
- QuadBox



How does it work?
Node Types

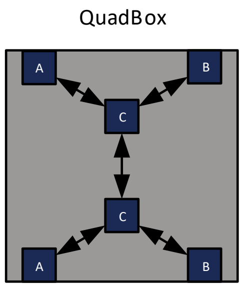
- **Double Attached Node (DAN)**
  - A node that takes part in the redundancy
  - Port C is internally and normally limited to one MAC
- **Single Attached Node (SAN)**
  - A node with only one port which is not taking part in the redundancy e.g. normal PC, Printer, etc.
  - Can be connected directly to one of the redundant networks in the case of PRP

How does it work?
Node Types

- **Redundancy Box (RedBox)**
  - A bridging node that takes part in the redundancy
  - Port C is externally and can handle more than one MAC
  - Connects nodes of non-redundant networks to the redundant network
  - Sends Supervision frames on behalf of the nodes connected on Port C
  - Filters frames out of the ring for the nodes connected on Port C in case of HSR

How does it work?
Node Types

- **Virtual Double Attached Node (VDAN)**
  - A SAN which is connected via a non-redundant network or link via a RedBox to the redundant network
- **Quad Box (QuadBox)**
  - A bridging node allowing to interconnect either multiple HSR rings or HSR rings with PRP networks
  - Basically built out of two RedBoxes in one node
  - Transfers redundancy information from one network to the other (LAN Ids and SeqNr) (not further discussed)

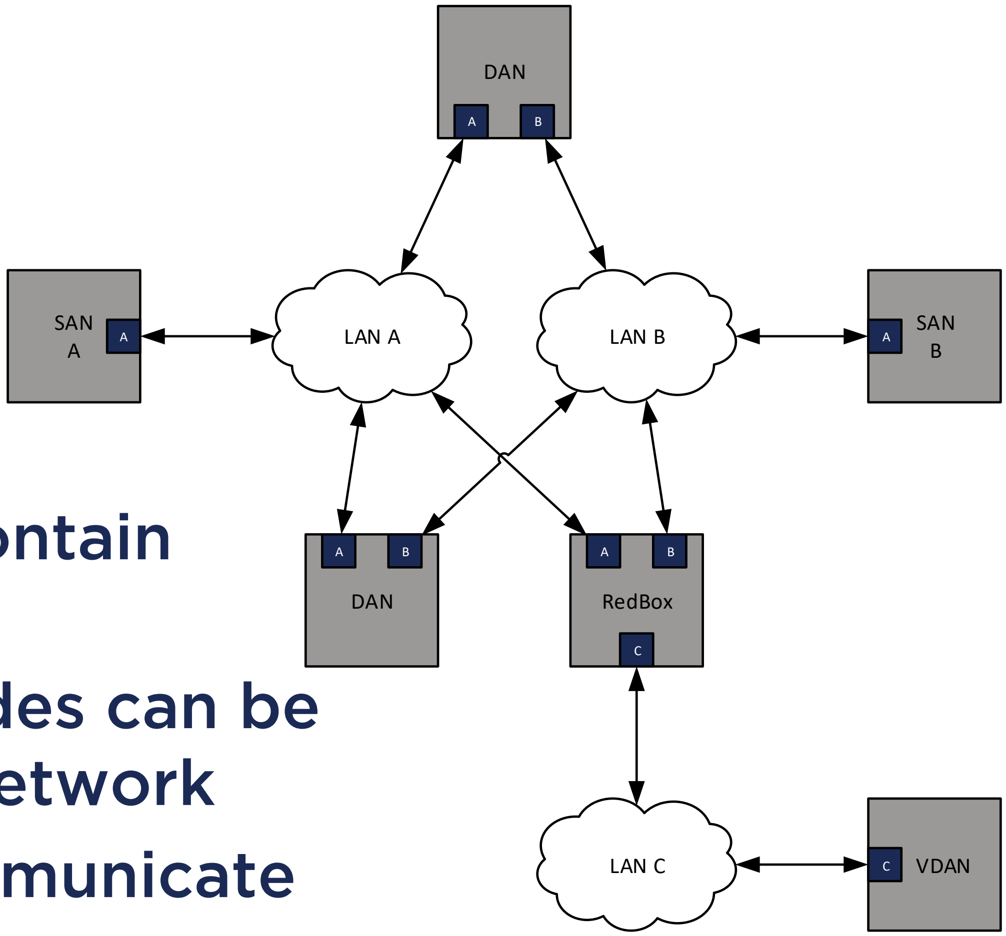


How does it work?
Single point of failure?

- **Nodes which need high availability need to be DANs**
  - RedBoxes introduce a single point of failure if either the RedBox is broken or the non-redundant link brakes
- **If two redundant networks shall be coupled two QuadBoxes need to be used**
  - Otherwise the QuadBox is the single point of failure

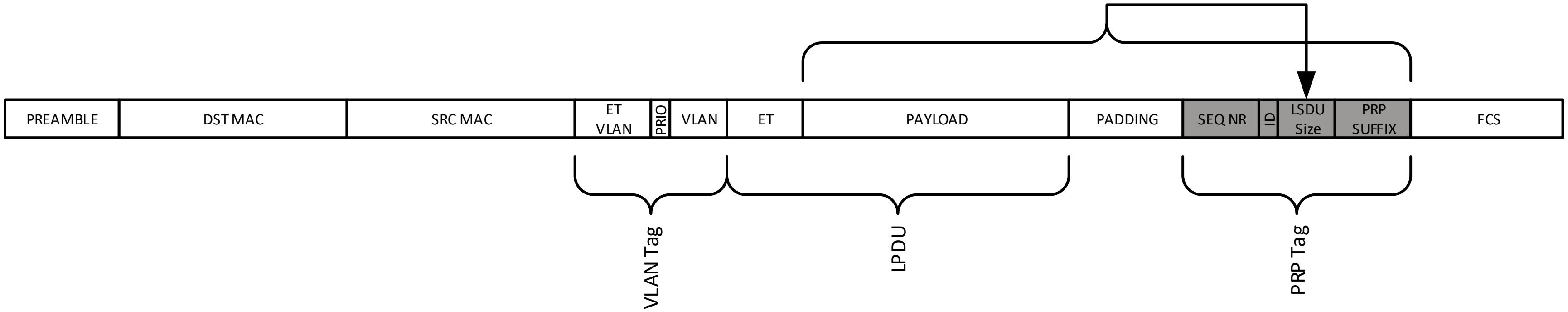
How does it work?
PRP

- Two independent networks for redundancy: LAN A and LAN B
- Non-redundant network: LAN C
- Any network can contain normal Switches
- Single attached nodes can be connected to any network
- SAN A can not communicate with SAN B



How does it work?
PRP

- **Tag is at the end of the frame (before FCS)**
- **Last 16bits are a tag identifier for PRP: 0x88FB**
- **A PRP tag can never be uniquely identified!**
  - Payload can have bytes the same way as a tag
  - Only if no SANs are present in the network the tag could be safely removed!
- **Non-redundant nodes treat the tag as padding**



How does it work?
PRP DAN

1. **Frame from Port C arrives at PRP core**
2. **Frame has to be received completely**
   - For tagging the payload size has to be know which requires that the whole frame is received
3. **Duplicate frame and increment sequence number**
   - Sequence number is passed to Port A&B
4. **Ports A&B insert tags**
   - At the end of a frame before the FCS
   - With the respective LAN identifier and the sequence number provided
5. **The two frames are sent over Port A&B**

How does it work?
PRP DAN

6. The two frames traverse through the networks
7. First frame of the duplicates arrive at other node e.g. Port B
8. Ports B checks if this frame is for this node
   - Drops it if not
9. Port B checks the tag and stores the «SrcMAC» & «SeqNr» & «LanId» of the frame
10. Port B checks if the frame is a duplicate
    - Which it is not since it was the first frame arrived
11. Ports B forwards the frame to Port C

How does it work?
PRP DAN

12. Port C optionally removes the tag
    - Normally not done since the tag can not uniquely identified
13. Port C forwards the frame to the host
14. Second frame arrives at Port A
15. Port A checks the tag and stores the «SrcMAC» & «SeqNr» & «LanId» of the frame
16. Ports A checks if this frame is for this node
17. Port A checks if the frame is a duplicate
    - Which it is since it is the second that arrived
18. Ports A drops the frame

How does it work?
PRP DAN

- If duplicate does not arrive remove the entry from the duplicate detection after a defined time
- Untagged frames are always forwarded to Port C
- Frames have to be always completely received before taking a forwarding decision

How does it work?
PRP RedBox

1. **Frame from Port C (network) arrives at PRP core**
2. **Frame has to be received completely**
   - For tagging the payload size has to be know which requires that the whole frame is received
3. **Make an entry or update entry in the so called «ProxyNodeTable» for this «SrcMAC»**
4. **Duplicate frame and increment sequence number for this «SrcMAC» entry in the «ProxyNodeTable»**
   - Sequence number is passed to Port A&B

How does it work?
PRP RedBox

5. **Ports A&B insert tags**
   - At the end of a frame before the FCS
   - With the respective LAN identifier and the sequence number provided
6. **The two frames are sent over Port A&B**
7. **The two frames traverse through the networks**
8. **First frame of the duplicates arrive at other RedBox e.g. Port B**
9. **Ports B checks if this frame is for the RedBox**
   - Drops it if so

How does it work?
PRP RedBox

10. **Port B checks the tag and stores the «SrcMAC» & «SeqNr» & «LanId» of the frame**
11. **Port B checks if the frame is a duplicate**
    - Which it is not since it was the first frame arrived
12. **Ports B forwards the frame to Port C**
    - No destination check since the RedBox doesn't know which nodes are connected on network C which have not yet sent a frame or are aged out
13. **Port C optionally removes the tag**
    - Normally not done since the tag can not uniquely identified
14. **Port C forwards the frame to network C**

How does it work?

PRP RedBox

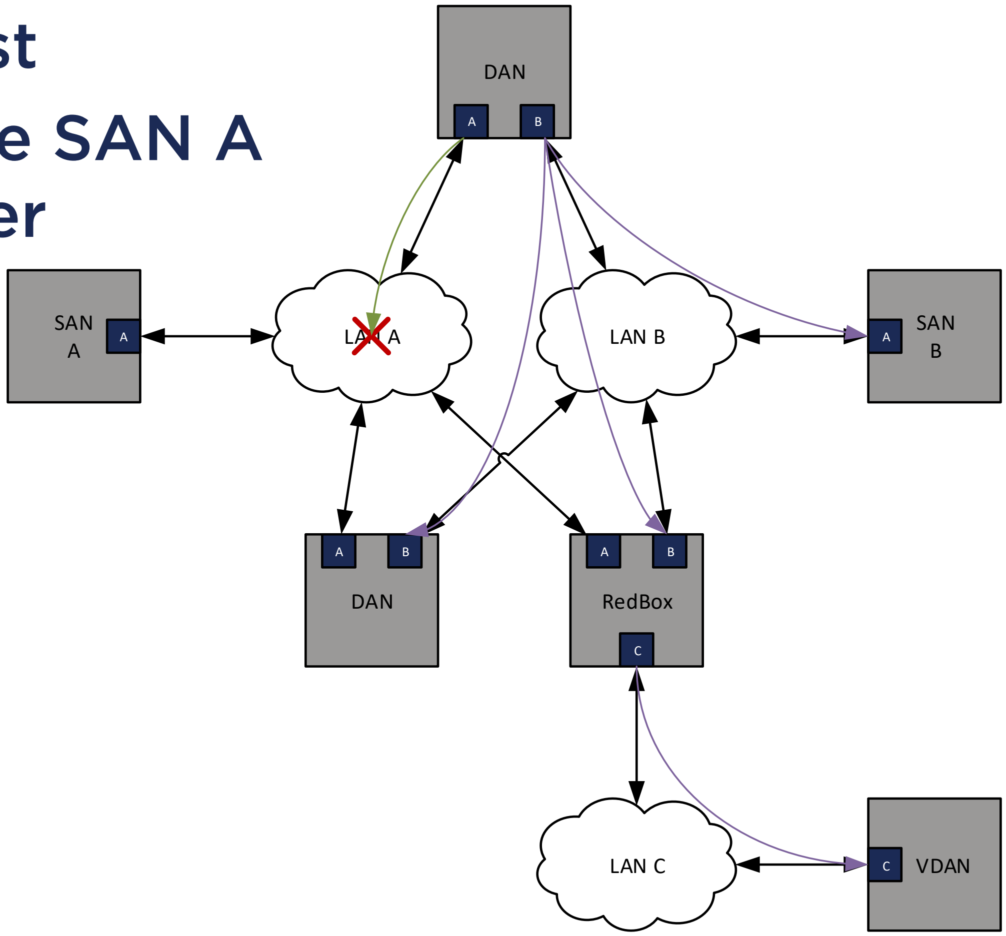
15. Second frame arrives at Port A of the RedBox
16. Port A checks the tag and stores the «SrcMAC» & «SeqNr» & «LanId» of the frame
17. Ports A checks if this frame is for the RedBox
18. Port A checks if the frame is a duplicate
    - Which it is since it is the second that arrived
19. Ports A drops the frame

How does it work?
PRP RedBox

- If duplicate does not arrive remove the entry from the duplicate detection after a defined time
- Untagged frames are always forwarded to Port C
- Frames have to be always completely received before taking a forwarding decision
- Entries in the «ProxyNodeTable» age out if no frames arrive anymore from this «SrcMAC» on Port C after a defined time
- RedBox sends in Supervision frames on behalf of each «SrcMAC» in the «ProxyNodeTable» in addition to its own Supervision frames

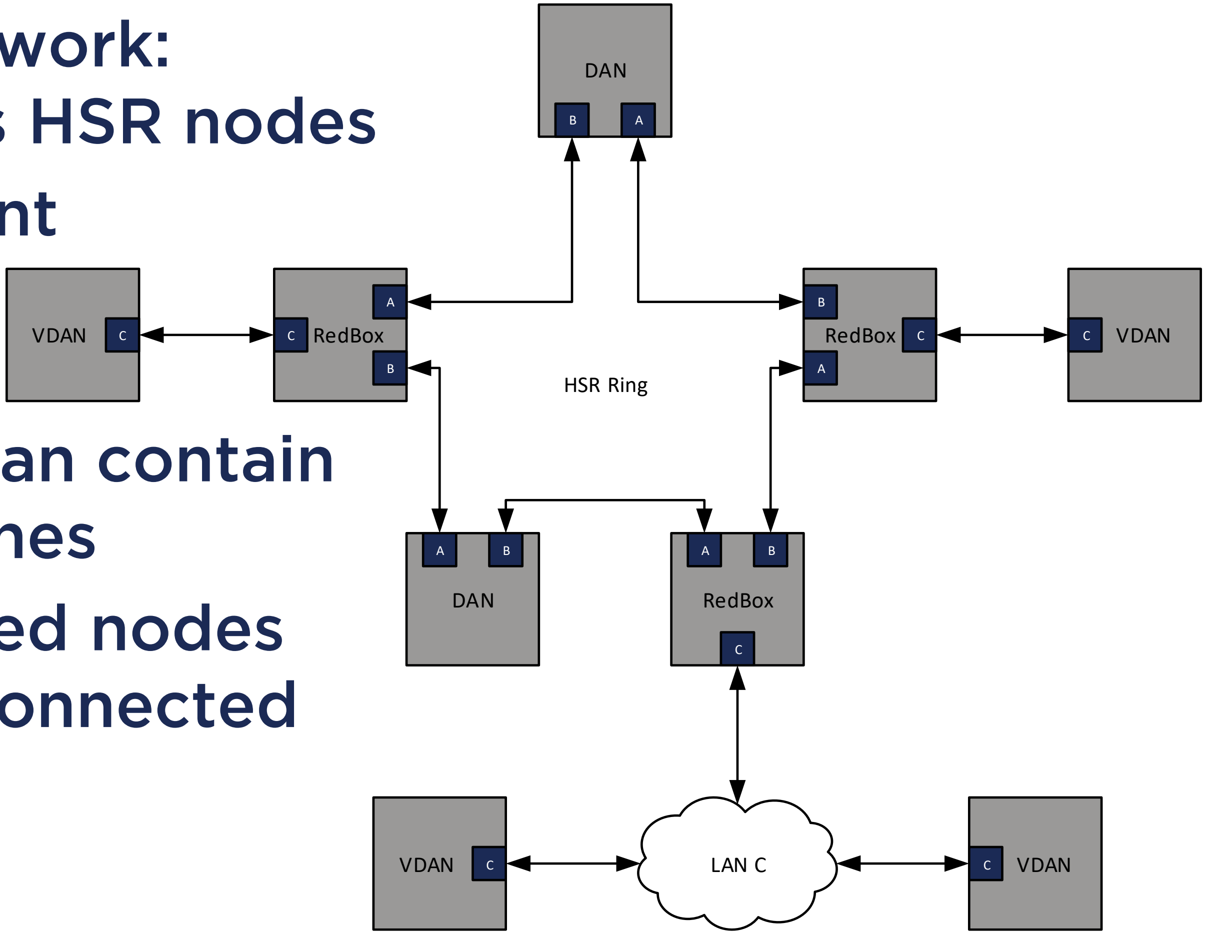
How does it work?
PRP

- DAN sends multicast
- All nodes except the SAN A are still reached over LAN B



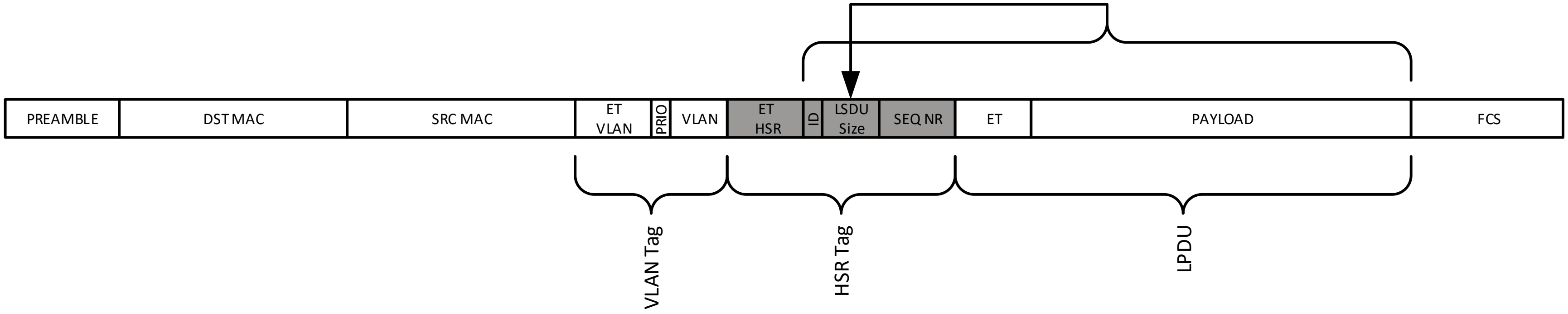
How does it work?
HSR

- HSR Ring network:
  Only contains HSR nodes
- Non-redundant network:
  LAN C
- Only LAN C can contain normal Switches
- Single attached nodes can only be connected to LAN C



How does it work?
HSR

- Tag is at the beginning of the frame (before Ethertype)
- First 16bits of the tag are the Ethertype of HSR: 0x982F
- A HSR tag can always be uniquely identified
- Non-redundant nodes treat the tag as a different protocol!



How does it work?
HSR DAN

1. **Frame from Port C arrives at HSR core**
2. **Frame has to be received completely**
   - For tagging the payload size has to be know which requires that the whole frame is received
3. **Duplicate frame and increment sequence number**
   - Sequence number is passed to Port A&B
4. **Ports A&B insert tags**
   - At the beginning of a frame before the Ethertype
   - With the respective LAN identifier and the sequence number provided
5. **The two frames are sent over Port A&B**

How does it work?
HSR DAN

6. **The two frames traverse through the ring**
7. **First frame of the duplicates arrive at other node e.g. Port B**
8. **Ports B checks if this frame is for this node**
   - Drops the frame to Port C if not
   - Drops the frame to Port A if it is
   - Forwards the frame to Port A if not (or multicast)
9. **Port B checks the tag and stores the «SrcMAC» & «SeqNr» & «LanId» of the frame**
10. **Port B checks if the frame is a duplicate**
    - Which it is not since it was the first frame arrived

How does it work?
HSR DAN

11. Ports B forwards the frame to Port C
12. Port C removes the tag
    - Has to be done always
13. Port C forwards the frame to the host
14. Second frame arrives at Port A
17. Ports A checks if this frame is for this node
    - Drops the frame to Port C if not
    - Drops the frame to Port B if it is
    - Forwards the frame to Port B if not (or multicast)

How does it work?
HSR DAN

18. Port A checks the tag and stores the «SrcMAC» & «SeqNr» & «LanId» of the frame
19. Port A checks if the frame is a duplicate
    - Which it is since it is the second that arrived
20. Ports A drops the frame

How does it work?
HSR DAN

- If duplicate does not arrive remove the entry from the duplicate detection after a defined time
- Untagged frames are always forwarded to Port C but not forwarded between Port A&B
- Link Local Traffic is not forwarded between Port A&B
- Forwarding decision can be taken after tag was received
- Frames are removed from the ring when the node is either a unicast destination or when a frame is received where a node was the source (multicast or non existing node)

How does it work?
HSR RedBox

1. **Frame from Port C (network) arrives at HSR core**
2. **Frame has to be received completely**
   - For tagging the payload size has to be know which requires that the whole frame is received
3. **Make an entry or update entry in the so called «ProxyNodeTable» for this «SrcMAC»**
4. **Duplicate frame and increment sequence number for this «SrcMAC» entry in the «ProxyNodeTable»**
   - Sequence number is passed to Port A&B

How does it work?
HSR RedBox

5. **Ports A&B insert tags**
    - At the beginning of a frame before the Ethertype
    - With the respective LAN identifier and the sequence number provided
6. **The two frames are sent over Port A&B**
7. **The two frames traverse through the ring**
8. **First frame of the duplicates arrive at other node e.g. Port B**
    - Which it is not since it was the first frame arrived

How does it work?
HSR RedBox

9. **Ports B checks if this frame is for the RedBox or a node in the «ProxyNodeTable»**
   - Drops the frame to Port C if for the RedBox
   - Drops the frame to Port A if it is either for the RedBox or a node in the ProxyNodeTable
   - Forwards the frame to Port A if not (or multicast)
10. **Port B checks the tag and stores the «SrcMAC» & «SeqNr» & «LanId» of the frame**
11. **Port B checks if the frame is a duplicate**

How does it work?
HSR RedBox

12. Ports B forwards the frame to Port C
    - No destination check since the RedBox doesn't know which nodes are connected on network C which have not yet sent a frame or are aged out
13. Port C removes the tag
    - Has to be done always
14. Port C forwards the frame to the host
15. Second frame arrives at Port A

How does it work?
HSR RedBox

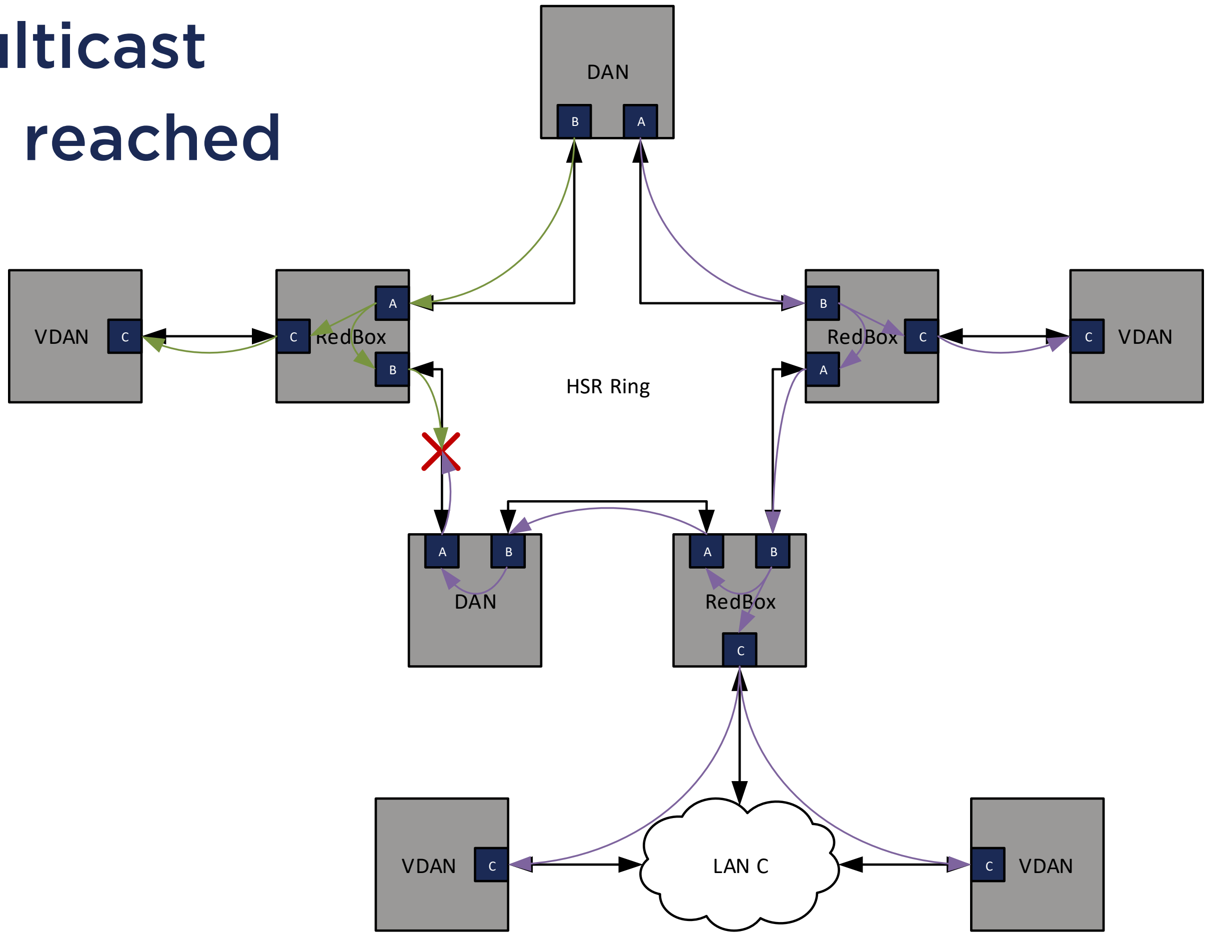
16. **Ports A checks if this frame is for the RedBox or a node in the «ProxyNodeTable»**
    - Drops the frame to Port C if for the RedBox
    - Drops the frame to Port B if it is either for the RedBox or a node in the ProxyNodeTable
    - Forwards the frame to Port B if not (or multicast)
12. **Port A checks the tag and stores the «SrcMAC» & «SeqNr» & «LanId» of the frame**
17. **Port A checks if the frame is a duplicate**
    - Which it is since it is the second that arrived
18. **Ports A drops the frame**

How does it work?
HSR RedBox

- If duplicate does not arrive remove the entry from the duplicate detection after a defined time
- Untagged frames are always forwarded to Port C but not forwarded between Port A&B
- Link Local Traffic is not forwarded between Port A&B
- Forwarding decision can be taken after tag was received
- Frames are removed from the ring when the node is either a unicast destination or when a frame is received where a node was the source (multicast or non existing node) in the «ProxyNodeTable»

How does it work?
HSR

- DAN send multicast
- All nodes still reached



How does it work?
Supervision

- **Each HSR&PRP node sends periodically (every 2s) «Supervision» frames**
  - Multicast MAC: 01-15-4E-00-01-XX
  - Ethertype: 0x88FB
  - TLVs for fields
  - Contains the protocol version and type, MAC (which might not be the same as the SrcMac of the frame, RedBox) and a monotonic increasing sequence number
  - If a RedBox, contains also the MAC of the RedBox
  - Supervision frames are tagged and duplicated the same way as other frames

How does it work?
Supervision

- **A RedBox sends Supervision frames on behalf of the nodes in the «ProxyNodeTable»**
  - Same format as for a DAN
  - MAC in the Supervision frame is the one from the node in the ProxyNodeTable
  - SrcMac of the frame is the one from the RedBox
  - Also inserts the MAC of the RedBox in a TLV
  - Sends for each node in the ProxyNodeTable a separate Supervision frame
  - Supervision frames are tagged and duplicated the same way as other frames coming from the RedBox itself

How does it work?
Supervision

- **With the information from the Supervision frames a HSR/PRP node can create a so called «NodeTable» with each entry containing:**
  - The MAC of the node seen
  - A time when the last frame was received on Port A and B
  - Flags for single attached node SAN A&B
  - Frame counters how many frames have been received on Port A&B
  - Error counters and rates for Port A&B
- **A HSR/PRP core normally drops Supervision frames on Port C**

How does it work?
Supervision

- **A «NodeTable» is optional**
- **It is sufficient to have only one node per redundant network which provides a «NodeTable»**
  - Since every node in the network is seen
  - A QuadBox or RedBox might drop Supervision frames on forwarding which would not allow supervision frame detection behind them

HSR vs. PRP

- **HSR**

  + Cost effective because of ring topology (no additional switches needed)

  + Rings of rings possible

  - Single attached nodes can only be connected via a RedBox

  - Every node in a HSR ring must participate in HSR, so no normal Switches can be used

  - Needs hardware support for fast forwarding

  - Higher complexity

HSR vs. PRP

- **PRP**
  - \+ Single attached nodes can directly be connected to either of the LANs
  - \+ Can be implemented in Software (on the cost of heavy CPU load though), simple to implement
  - \+ Normal Switches can be used in the Networks
  - \+ Network topology in the two Networks flexible and can be in principle also redundant networks itself
  - \- Doubles the cost of network infrastructure
  - \- SANs in LAN A can not communicate with SANs in LAN B

HSR&PRP with PTP
What is PTP?

NetTimeLogic GMBH

- **PTP**
  - Precision Time Protocol standardized in IEEE1588
  - Allows synchronization in the sub-microsecond range
  - Measures path delays
  - Builds a Master-Slave topology based on messages and the so called Best Master Clock Algorithm (BMCA) which evaluates many different quality values and uses Port Identities as a tiebreaker to determine the best clock

  - More information about PTP: PTP Basics.pdf

HSR&PRP with PTP Challenges

- **PTP needs to know the path it synchronizes to and HSR&PRP hide from which path the frame was received**
- **PTP traffic has to be threated differently from other traffic to allow PTP to know over which path it synchronizes**
  - No seamless redundancy for PTP, on failure a switchover happens for the synchronization
  - Switchover delay heavily depends on PTP frame rates and timeouts
  - Switchover delay in a couple of seconds is acceptable since synchronization will still be ok

HSR&PRP with PTP Solutions

NetTimeLogic GMBH

- **A PTP profile defines the handling of PTP with HSR&PRP: Utility Profile**
- **Redundant Clocks (OC, BC or TC)**
  - A node which is redundantly connected and determines which path it will synchronize to based on the BMCA (and/or other status information) and only forwards timing information from one path
- **Stateless Clocks (TC)**
  - Modifies PTP frames (tiebreaker: PortId bit 13:12) on reception on Port A&B with a LAN tag and forwards all (also duplicates) PTP traffic to Port C, letting the nodes behind choose the path to synchronize to based on the BMCA

Questions

Thank you!

Questions?

NetTimeLogic GmbH
www.nettimelogic.com
contact@nettimelogic.com