IEEE 802 Plenary Session
July 15-20, 2012 – San Diego, California

Support for Seamless Redundancy in AVB 2

(Version Date: July 15, 2012)

Markus Jochim, General Motors Research & Development
Oliver Kleineberg, Hirschmann Automation & Control

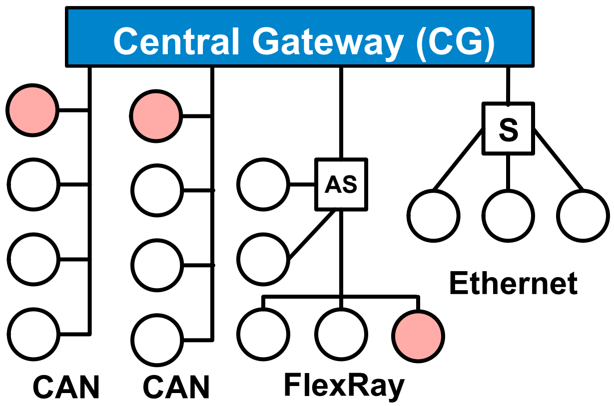
Structure of the presentation

- **PART I: Similarities in Automotive & Industrial Automation Redundancy Use Cases**
- **PART II: Proposal for a Seamless Redundancy Concept within AVB 2**

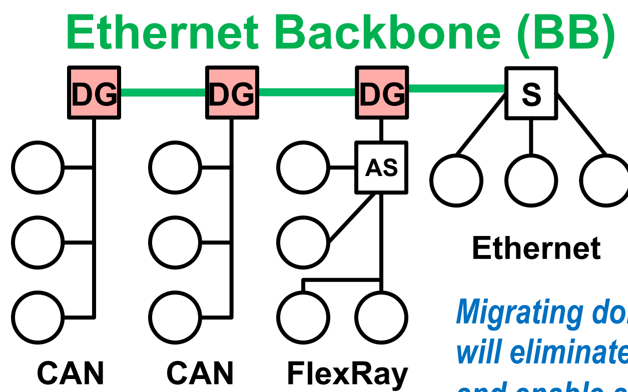
Part I:
Similarities in Automotive & Industrial Automation Redundancy Use Cases

Example 1: Architecture Level

Today:



Future:



Migrating domain busses to Ethernet will eliminate Domain Gateways (DG) and enable and Ethernet / IP based communication architecture!

- Goal: Connect automotive Domains (e.g. Powertrain, Chassis, Body, Infotainment)
- Central Gateway architecture is one of today's common solutions.
- DG = Domain Gateway with integrated Switch
  (One of the domain ECUs is selected to serve as a domain gateway that connects non-Ethernet domains to the backbone)

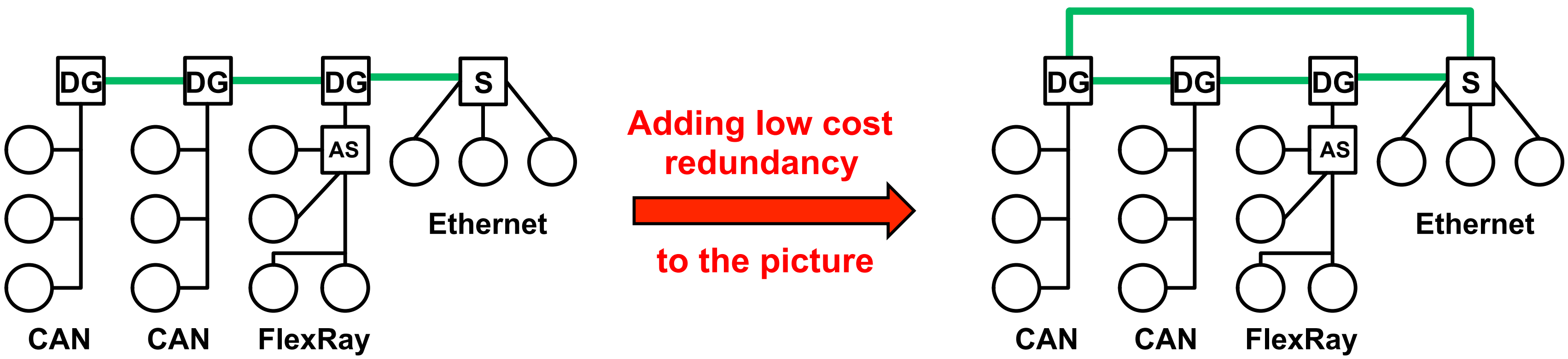
Example 1: Architecture Level

Comparison: "Central Gateway (CG)" vs. "Ethernet Backbone (BB)"

- Varying numbers of required ports (different vehicles, future extensions) will require developing / maintaining multiple variants of the CG device. (= Device proliferation)
- Flexibility / Scalability:
  - Adding / removing a domain bus: BB: Simple CG: Device proliferation
  - Future migration of domains to Ethernet: BB: Simple CG: Device proliferation
- CG: Complex customized device BB: Standard network components (Switches, NICs)
- BB:
  - More domains migrate to Ethernet => Fewer Domain Gateways required
  - IP as a "common language" in an Ethernet / IP based network
  - New IEEE 1722 formats support gating between CAN (FlexRay) domains over an AVB backbone.

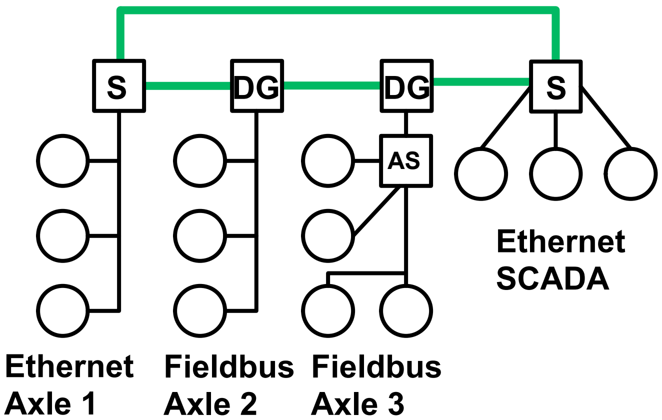
=> Once Ethernet@Automotive matures, the arguments for basing automotive communication architectures on Ethernet Backbones will be very convincing !

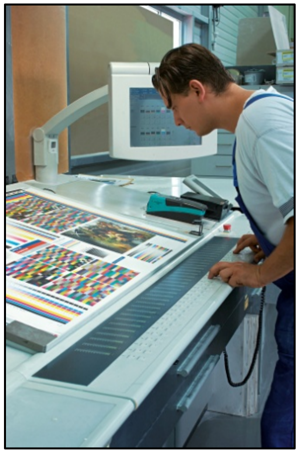
Example 1: Architecture Level



- **Redundancy** in a Backbone :
  - Network won't "fall apart" upon a single link failure **(Robustness)**
  - Supports design of **safety critical <u>interdomain</u> applications !**
- Example: Use available camera data across different domains and in multiple applications (Surround view, Semi-autonomous driving, . . .).
- Boundaries between classical automotive domains are blurring anyway !

Example 1: View from another angle (Industrial Control)



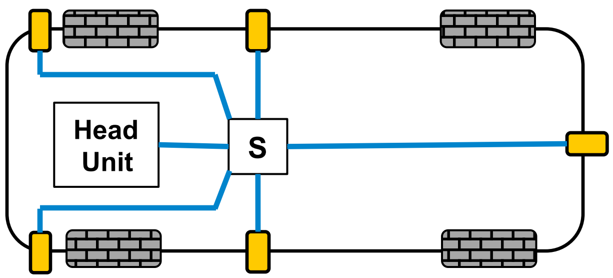


- **Redundancy** in factory automation / motion control :
  - Application of same concept to industrial automation
- Example: Synchronized axles in a large printing machine
- Individual Axle controls (either RT Ethernet or Fieldbus) are interconnected using Ethernet Backbone
- This example shows how smilar the use cases can be in ind. Automation and Automotive

Example 2: Subsystem Level

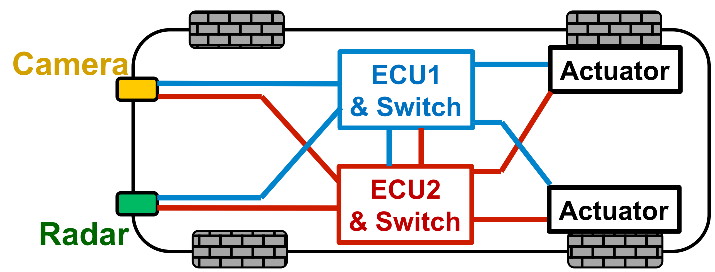
Today:

- Infotainment Systems
- Camera based Driver Assist
- . . .



Future:

- Everything on the left side
- Plus:
  Robust transmission to support Safety Critical Control Systems (Sensing and actuation !)



"Warn & Assist" ⇒ **"Warn, Assist & Actively Control"**

Trend towards "Active Control" will increase the need for robust and redundant communication (e.g. as a robust black channel for safety).
Ethernet Standard to support the design of such systems !

Example 2: View from another angle (Industrial Control)

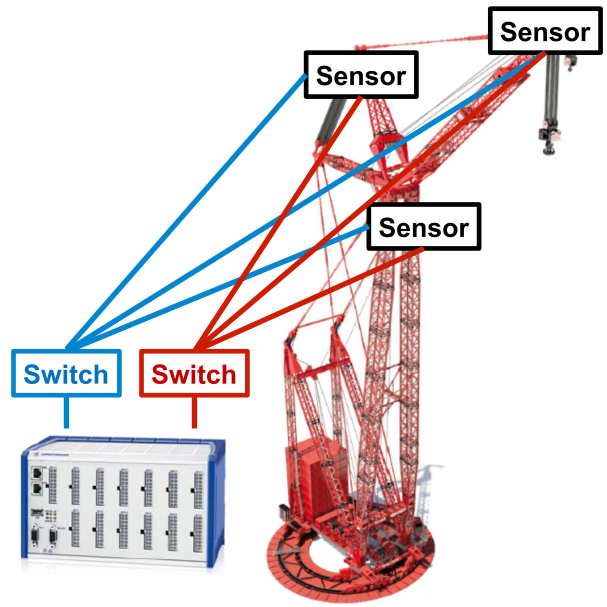
Today:

- Distributed control system of sensors and controllers in **mission-critical environments**
- Redundant communication and data handling (via Fieldbus)

Again: Different applications but very similar requirements in Industrial Control and Automotive !

Future:

- Mission-critical communication handling (fully redundancy) via Ethernet



From "Alert & Warn" to "Actively Control"

Examples (Today & Near Future)

Today: (MY 2013 Production Vehicle)

- ***Driver Alert Systems:***
  - Forward Collision Alert
  - Lane Departure Warning
  - Side Blind Zone Alert
  - Rear Cross Traffic Alert

Warn Driver

- ***Driver Assist Systems:***
  - Full-speed Adaptive Cruise Control
  - Collision Preparation
  - Front & Rear Automatic Braking (Virtual Bumper)
  - Stability Control

Active Control

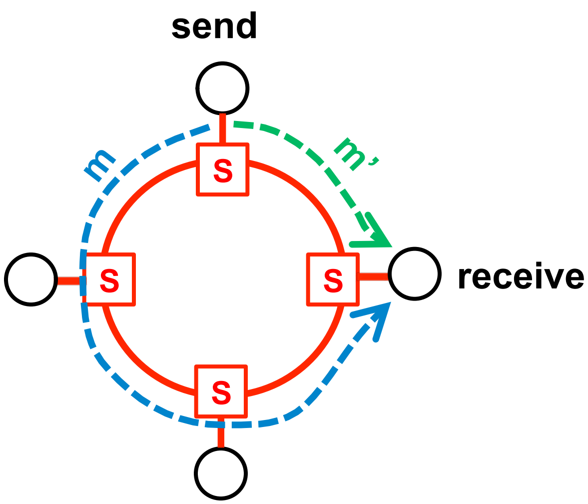
Near Future: Super Cruise Application

- Fully automatic steering, braking and lane centering in highway driving under certain optimal conditions.
- Fusion of radar, ultrasonic sensors, cameras and GPS map data
- Could be ready for production vehicles by mid-decade!

Active Control / Semi-Autonomous

Part II:
Proposal for a Seamless Redundancy Concept within AVB 2

Let’s start with some topology diagrams . . .

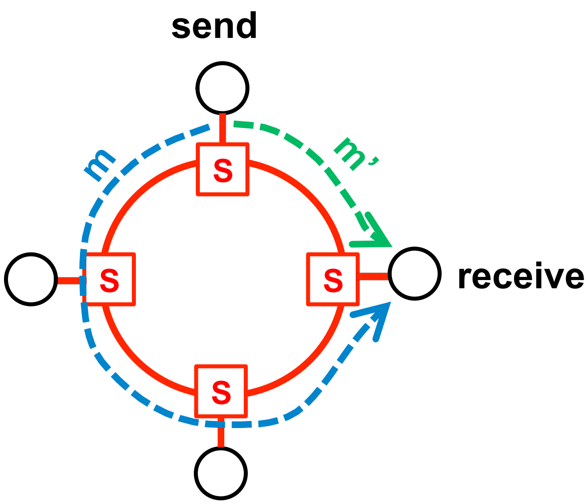


Structure: Ring

- This resembles an HSR ring. (IEC 62439-3 Clause 5).
- Sending out m and m’ simultaneously “to the right” and “to the left”.
- m = m’

Symbols: ○ a node   S a switch   m   m’   two messages

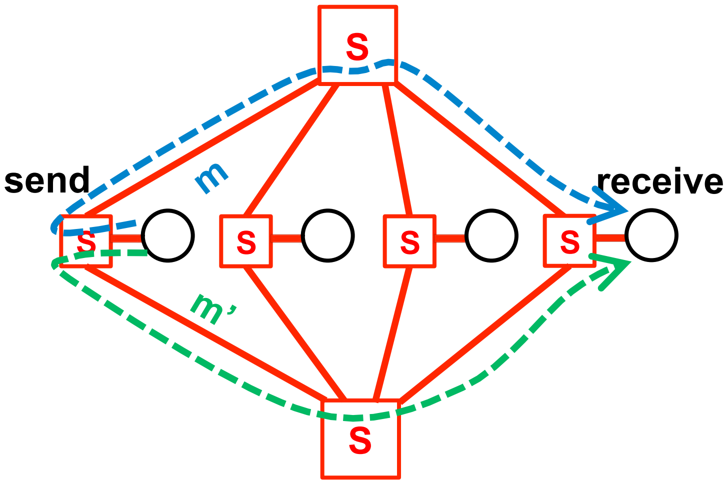
Ring: Fault tolerance / Use Case / Costs



- Tolerates one ring link failure
- Switch failure will isolate one node
- But leaves the remaining network intact.
  (Might be acceptable, since a fault tolerant design of the overall system / application may typically anticipate the possibility of a single node failure anyway.)

- Example Use Cases:
  Increased robustness for automotive backbone or automation network on the shop floor
- Moderate additional costs (1 extra link)
- Standard redundancy use case in industrial automation!

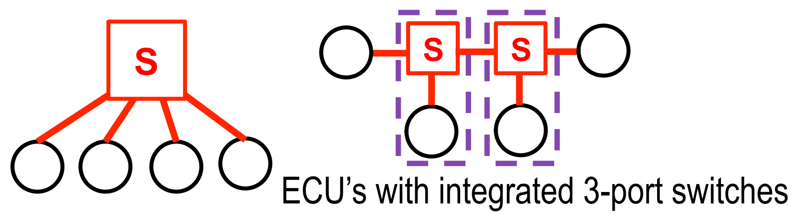
Another topology diagram . . .



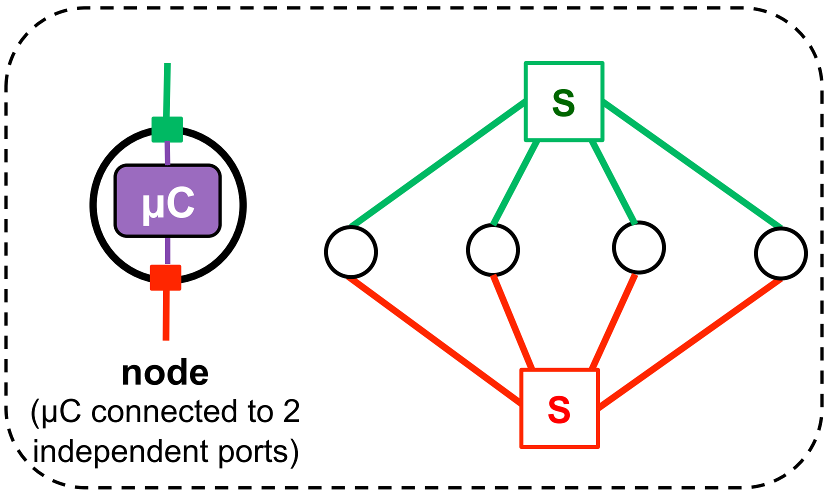
Structure: Dual channel

- This structure <u>looks</u> similar to the Active Star based dual channel FlexRay systems used for fault tolerant applications.

- Example Use Case: Safety critical domain application
- **Significant!** additional cost compared to non-redundant 4 node networks:
- But: Can be used „safety-aware“ or as very robust black channel (depends on end node implementation)

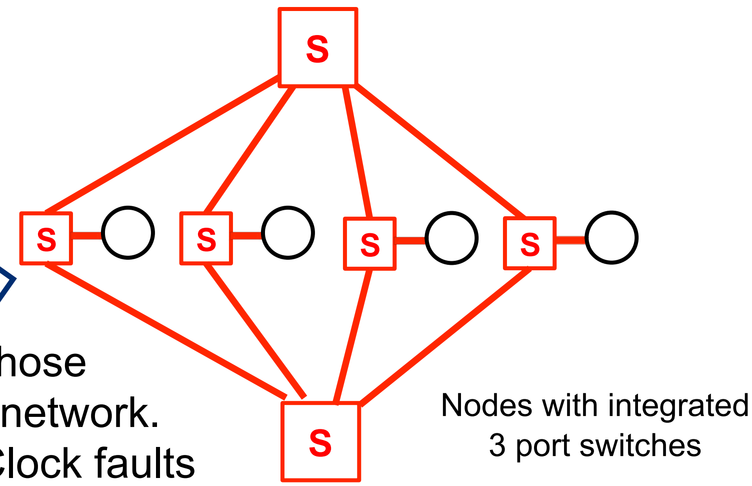


A note on fault propagation



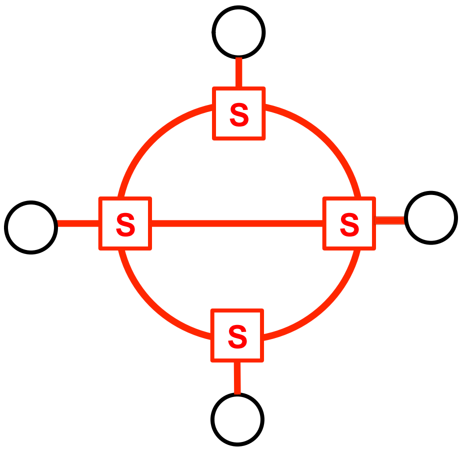
Actually
this might be closer to
the idea of 2 channels that fail
independently than that.

- Of course the µC still ties the two channels (or LANs) together.
- But otherwise they are pretty independent, which is nice from a fault propagation perspective.
- We have less independency in this topology and we should therefore even more carefully discuss those faults that have the potential to propagate through the network. (E.g. Babbling idiots may require ingress monitoring, Clock faults may requiring a fault tolerant clock sync algorithm).



Let's look at some extensions to the Ring concept. . .

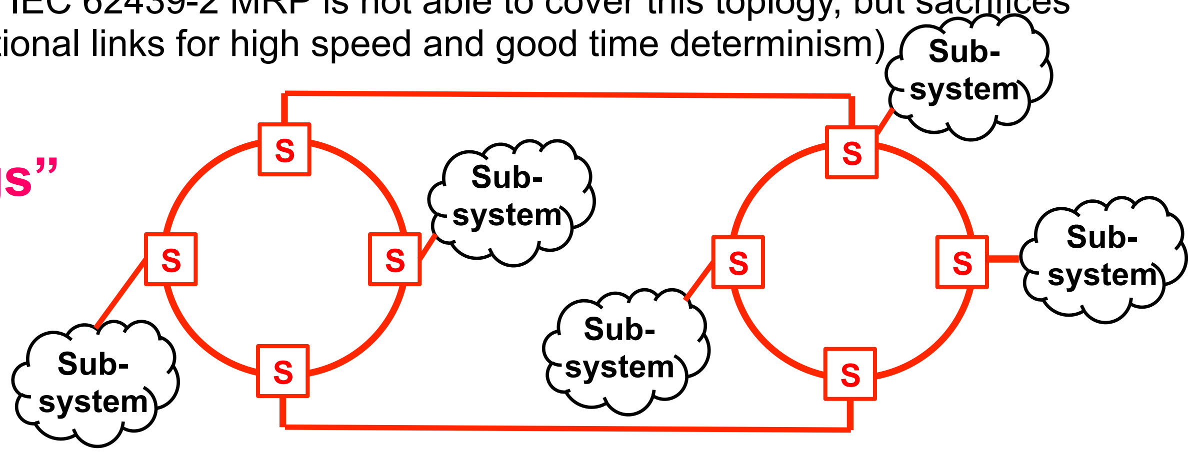
How much bandwidth & fault tolerance is required depends on the application. So let's have a look at some other topologies.



- **Structure: "Extended Ring"**
  Let's add another link to increase bandwidth & robustness.

- But: Increases requirements to redundancy control protocol functionality
  (E.g. IEC 62439-2 MRP is not able to cover this toplogy, but sacrifices additional links for high speed and good time determinism)

- **Structure: "Ring of Rings"**
  For an industrial application:



Terminology

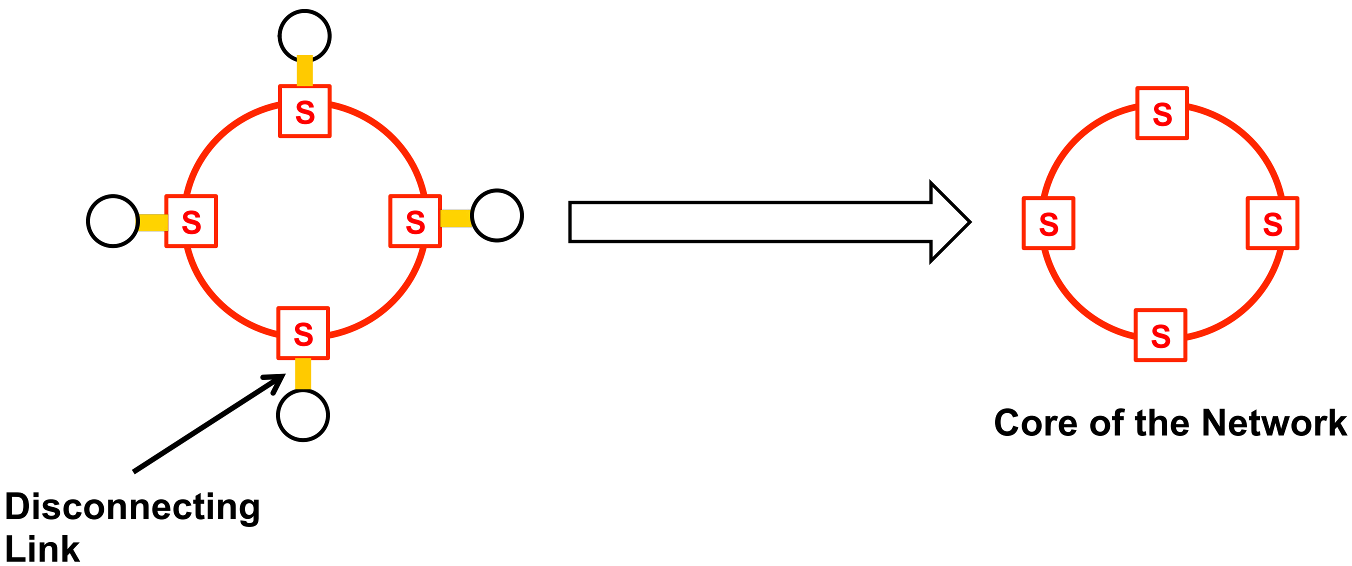
- 2-Link Connected Network: (*1)
  A network is 2-Link Connected if there are two edge-disjoint paths from every node to every other.
- Since everything that is presented can easily be extended beyond “2” (maybe for aerospace applications) one can also define K-Link Connected Networks:
  A network is k-Link Connected if it remains connected whenever fewer than k links are removed.
- This is also described by the n-1 redundancy concept, where n-1 „links“ in an n-fold network may fail/be removed without total system failure
- Disconnecting Link:
  A link in a network is a disconnecting link if the removal of that link from the network would leave a disconnected network.

(*1): Graph theory uses terms like: “k-edge-connected graph” and “bridge edge”.
Since terms like “bridge” and “edge” have a different semantic in networks, the terminology was tweaked a bit:

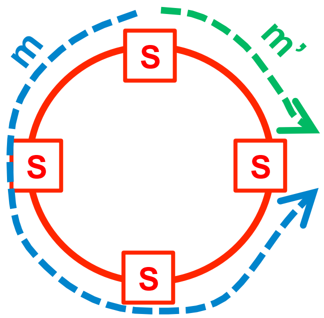
- Instead of “k-edge connected graph” we use “k-link connected network”
- Instead of “bridge edge” we say “disconnecting link”

Ignore Disconnecting Links

- On the next slide ignore all Disconnecting Links !
- And ignore all nodes or subsystems connected via Disconnecting Links !
- Focus on the “Core of the Network” instead

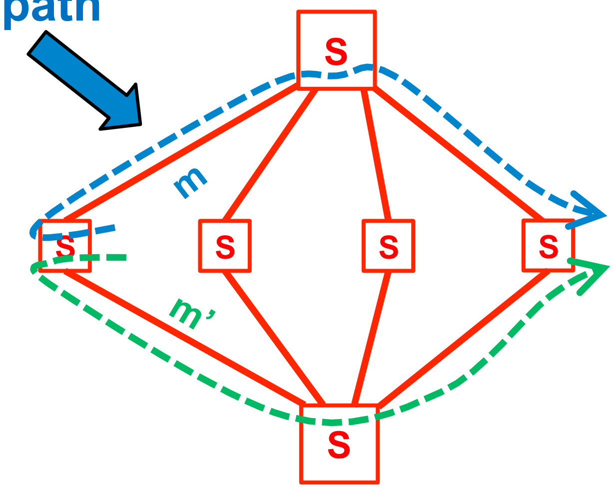


Let's now change the terminology for topologies !!!

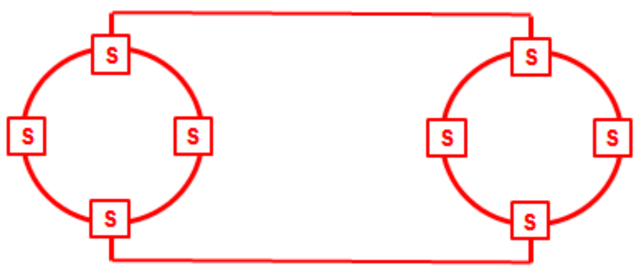


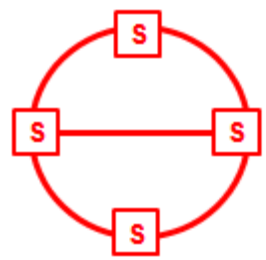
- **Do not call this a ring!**
  Call it "the simplest possible **dual path** topology".

- **Let's not talk about dual channels**
  But just about another **dual path** topology that offers two independent paths between any two switches



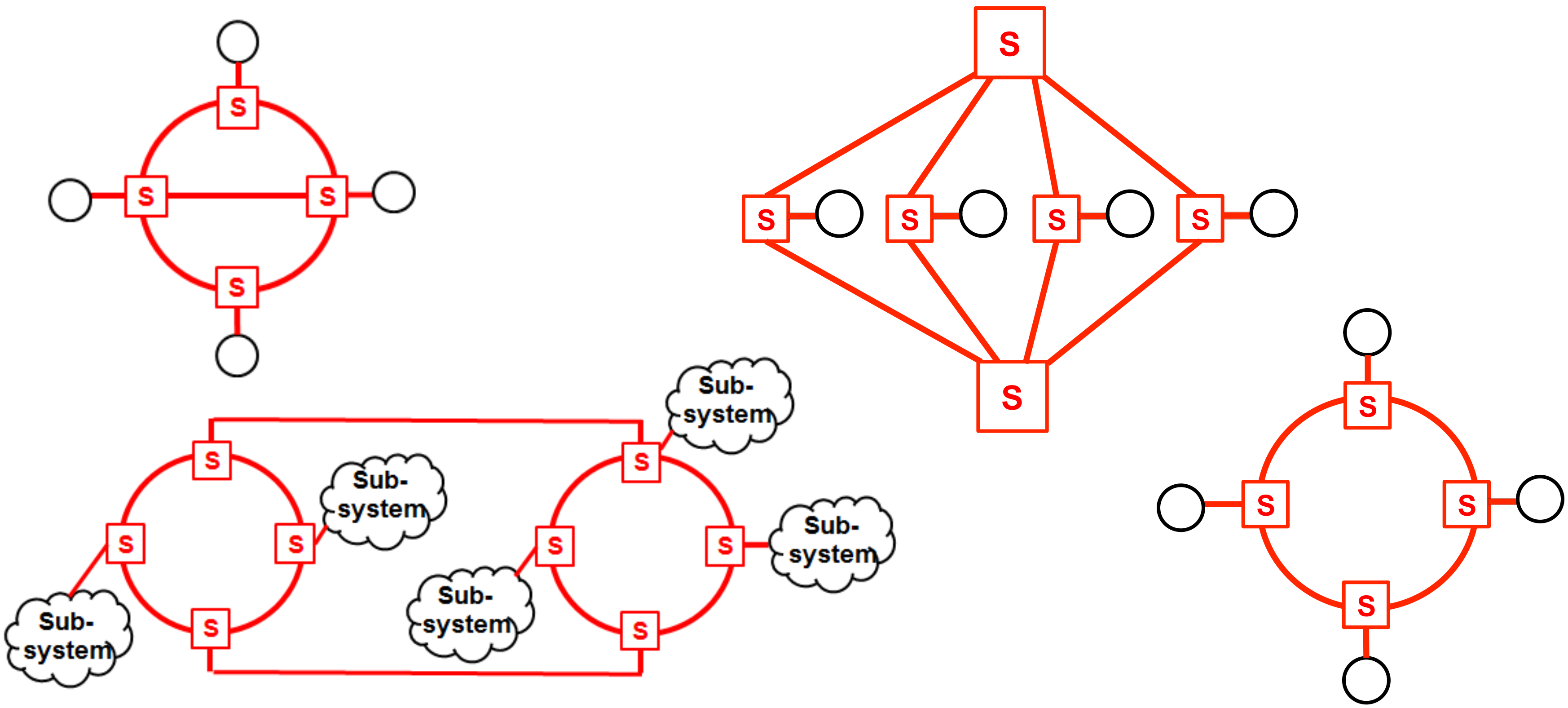
- **No "ring of rings" or "extended rings":**
  But further **dual path** examples.



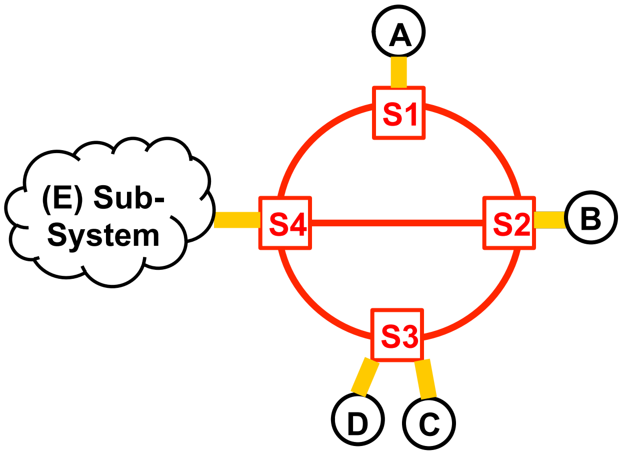


How can AVB support dual paths ?

- So what do we need to add to AVB, to support dual paths in all of these topologies (and in many more) ?



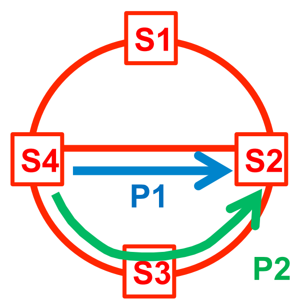
Switch "knows" which ports connect to Disconnecting Links



Red = 2-link connected core of the network
Orange = Disconnecting links

- Each switch knows (by configuration or by protocol), which port is connected to a disconnecting link.
- Within the 2-link connected core of the network, each switch "knows" two independent paths to each other switch.

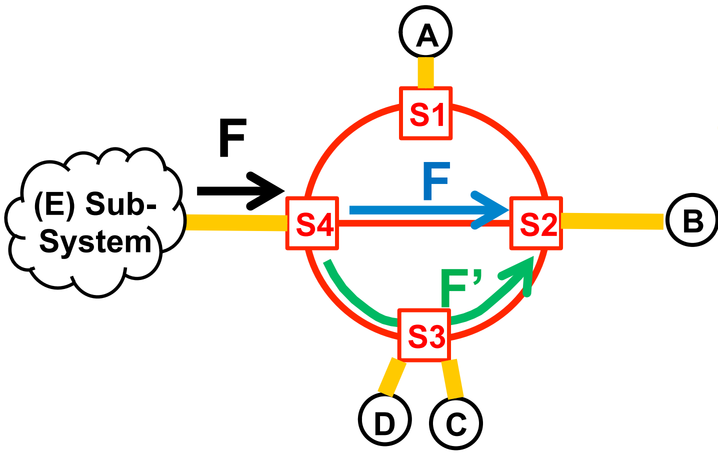
- Path selection: Through management interface, from higher layer
  - For automotive networks: 2 engineered paths
  - For industrial control: Protocol or expert or offline tool manages paths
- Example: S4 to S2:



Mission-critical & Non-Mission-critical Frames

- Each frame can be classified to be either a mission-critical frame, or a non-mission-critical frame.
- How do switches identify frames as either mission-critical or non-mission-critical?
- Traffic classes:
  - For scheduled traffic: Based on arrival time. (Or tag the frame)
  - For rate constraint traffic: Criticality as a property of the stream
  - For best effort: Always non-mission-critical

Replication of Mission-Critical Frames

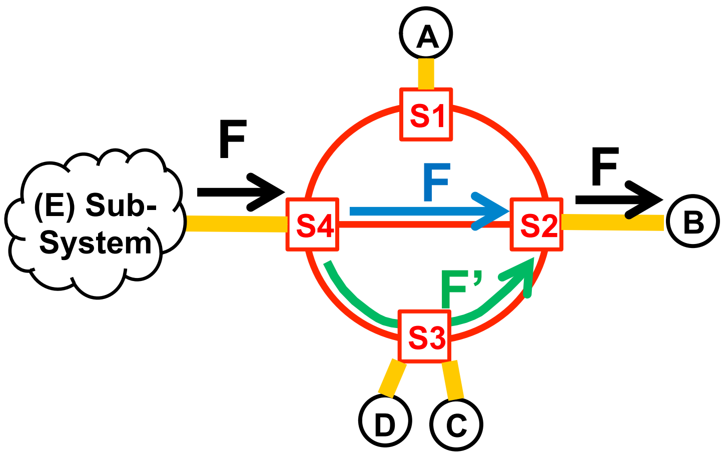


Red = **2-link connected part of the network**
Orange = **disconnecting links**

- Assumption:
  Subsystem E sends a mission-critical frame F to node B

- Switch S4 "knows":
  1) that F entered the switch via a port that connects to a disconnecting link
     AND
  2) that F is a critical frame
     (Based on criteria outlined on previous page.)
- S4 therefore replicates F and sends **F** and **F'** via the two shortest paths to S2.

Redundancy management (1/2)

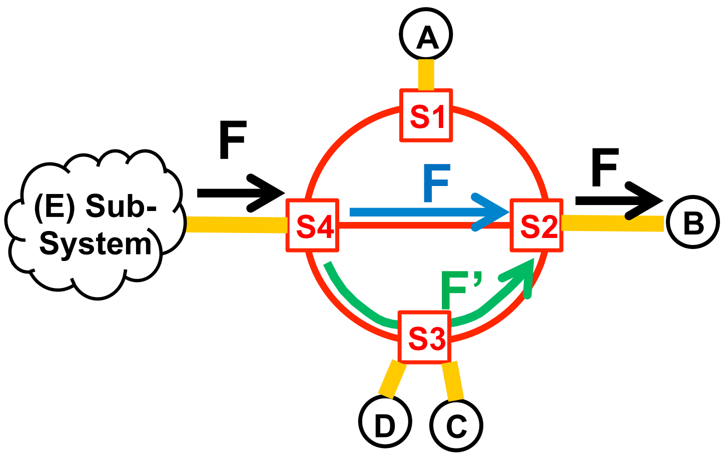


- Switch S2 "knows":
  1) that the link to B is a Disconnecting Link
     AND
  2) that F and F' are critical frames.

- S2 will therefore perform the redundancy management function.
- Example of a simple redundancy management function "Pick first":
  S2 forwards only the first frame (either F or F') that arrives at S2.

- But how does S2 know that F and F' form a pair of redundant messages?
  (See next slide!)

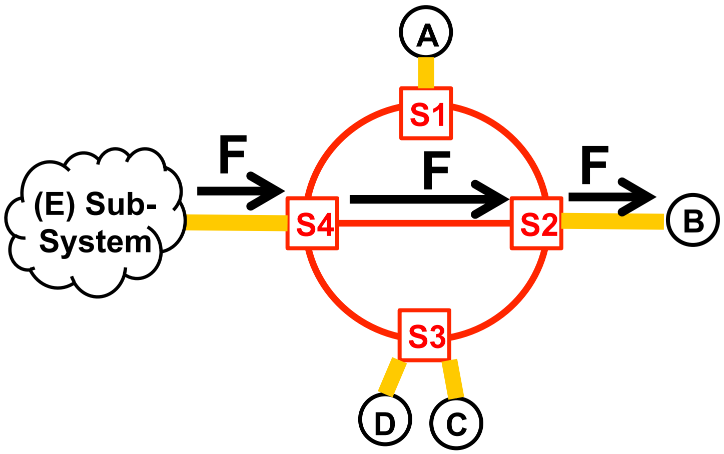
Redundancy management (2/2)



- How does S2 know that F and F' form a pair of redundant messages?
- The answer is different for different traffic classes!

- For scheduled traffic: Based on the expected arrival times for F and F'.
- For best effort traffic: Not necessary since F is non-mission-critical !
- For rate constraint traffic:
  - Think about 2 redundant streams F and F' rather than 2 individual frames F and F'
  - Declare F to be the primary stream and F' to be the secondary stream.
  - If stream F becomes unavailable switch to stream F'.

Non-Mission-critical Frames



- Assumption:
  Subsystem E sends a non-mission-critical frame F to node B

- No replication at S4, no redundancy management at S2.
- Mission-critical and Non-Mission-critical frames on the same network.

Requirement: Redundancy support for AVB Traffic Classes

- The proposed redundancy mechanism is simple and enables the combination of two concepts:

Mission-Critical Traffic

Non-Mission-Critical Traffic

with

Scheduled Traffic

Rate Constraint Traffic

Best Effort Traffic(*1)

Criticality — **Traffic Classes**

- For a **converged AVB 2 network** that enables a multitude of use cases we need to be able to bring these two concepts together !

(*1): The combination of "Best Effort" and "Mission-Critical" is not anticipated.

Objectives and proposal . . .

Objectives:

- Our main objective is to introduce the seamless redundancy mechanism that is compatible with the AVB 2 traffic classes
  (= Supports redundant rate constraint and redundant scheduled traffic)
- The redundancy concept is simple and lightweight.
  - ⇒ Can be discussed / defined / introduced within a reasonable time!
  - ⇒ Is suitable for resource constraint embedded systems!

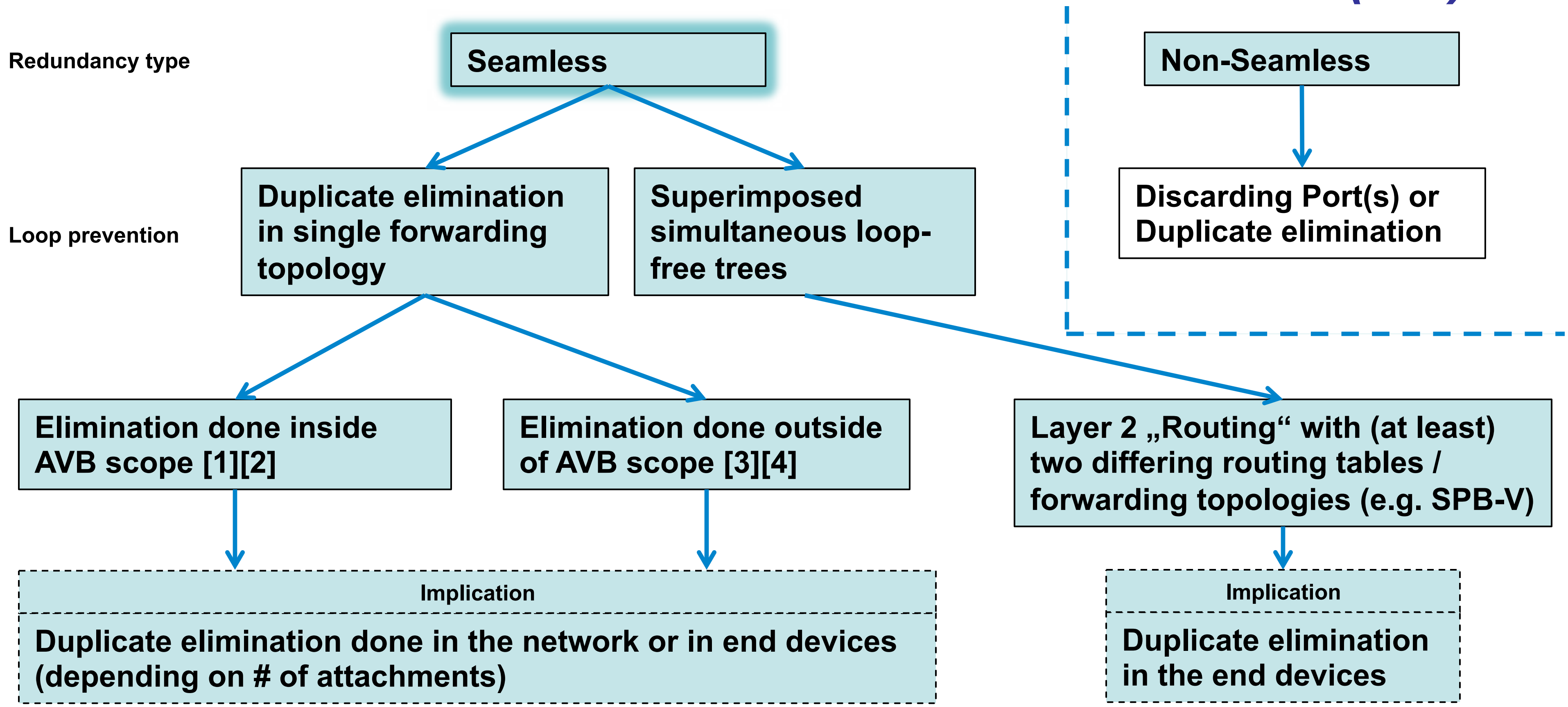
Proposal:

- We'd like to ask those who support the introduction of such a redundancy mechanism and those who have concerns to work with us during the next couple of months to refine the concept.
- This enables a timely solution / timely input to the upcoming redundancy PAR.

Options how to realize Seamless Redundancy (1/2)

- The redundancy concept proposed in this slide deck is one of several possible solutions.
- Key is that the solution we pick should have the properties outlined on the previous slide.
- The following slide gives an overview of other potential options of how to realize Seamless Redundancy.

Options how to realize Seamless Redundancy (2/2)



[1] This presentation, chapter 2
[2] http://www.ieee802.org/1/files/public/docs2011/at-kleineberg-AVB-media-redundancy-0311.pdf
[3] http://www.ieee802.org/1/files/public/docs2011/at-kleineberg-AVB-media-redundancy-1111-v02.pdf
[4] http://www.ieee802.org/1/files/public/docs2012/at-kleineberg-avb-fault-tolerance-continuation-0312-v3.pdf