

Ingress Policing in Automotive Systems

Soheil Samii, General Motors R&D

Johannes Specht, Univ. of Duisburg-Essen

Ethernet in Automotive Systems

- Automotive Ethernet will grow – Advanced Driver Assistance Systems (ADAS) is the major growth driver
 - IEEE-SA Ethernet & IP @ Automotive TechDay, October 2014
Keynote by Ian Riches, Strategy Analytics
- Fail-operational ADAS such as automated driving and active safety applications require fault-tolerance mechanisms (main driver: ISO 26262 “Road Vehicles – Functional Safety”). System-level solutions will have to include
 - P802.1CB Seamless Redundancy, and
 - Ingress Policing – *let us revisit past discussions and make progress on this topic ...*

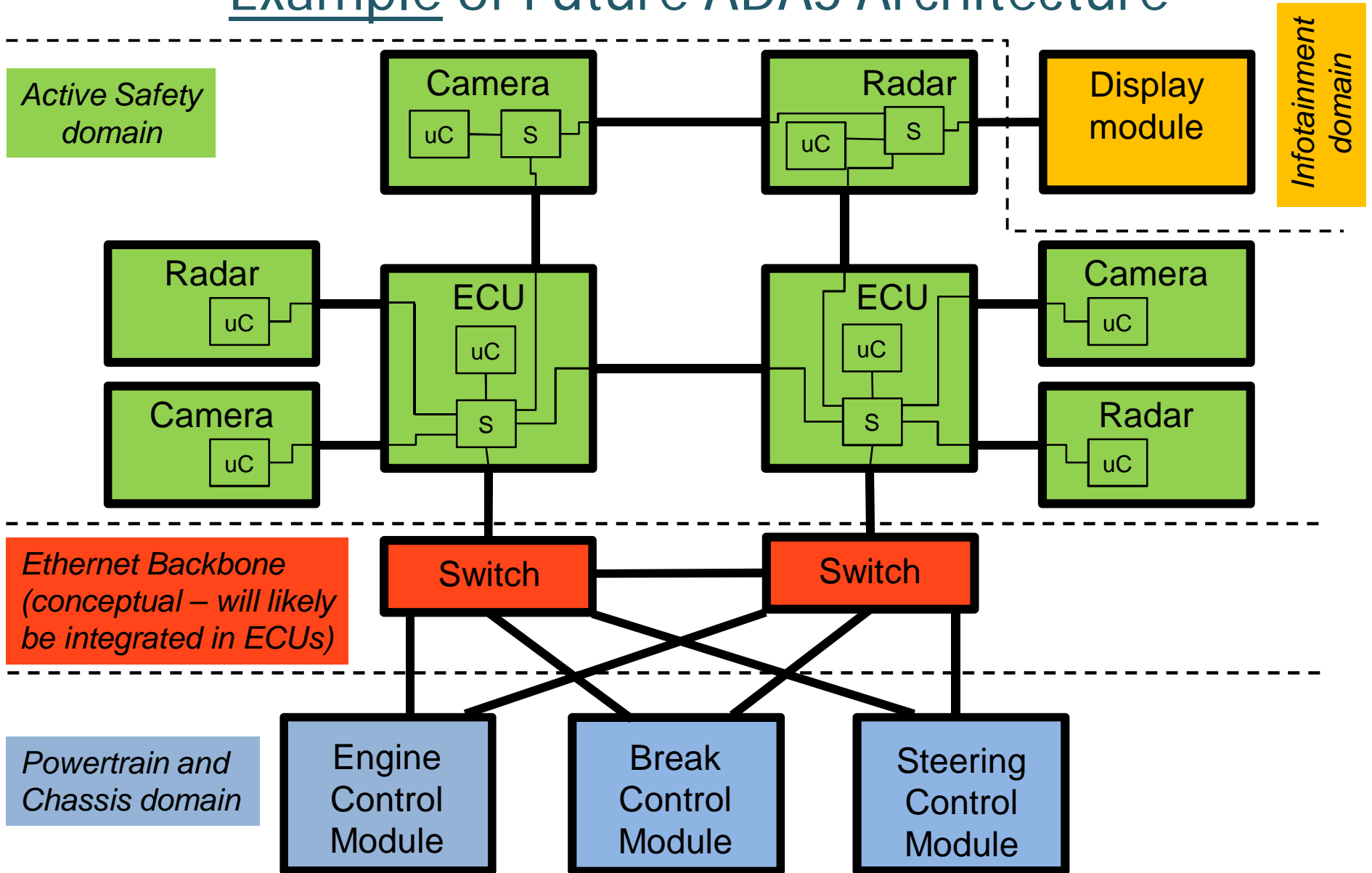


Agenda

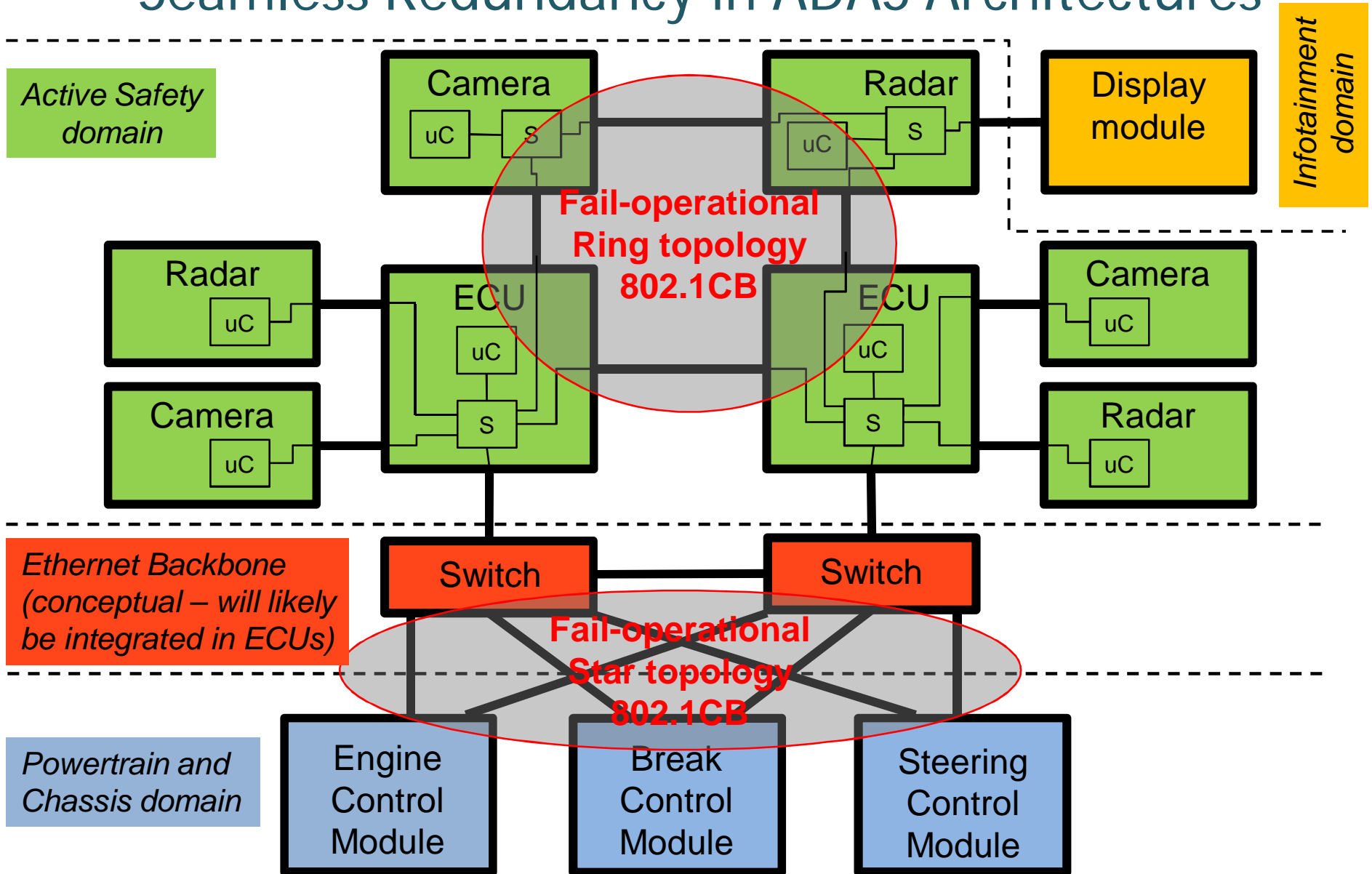
- ➔ Motivation and context for Ingress Policing
- Revisit conclusions from Ingress Policing analysis at Dallas Plenary November 2013
- Required properties and characteristics of ingress policing
- Discussion on how to proceed with ingress policing in TSN



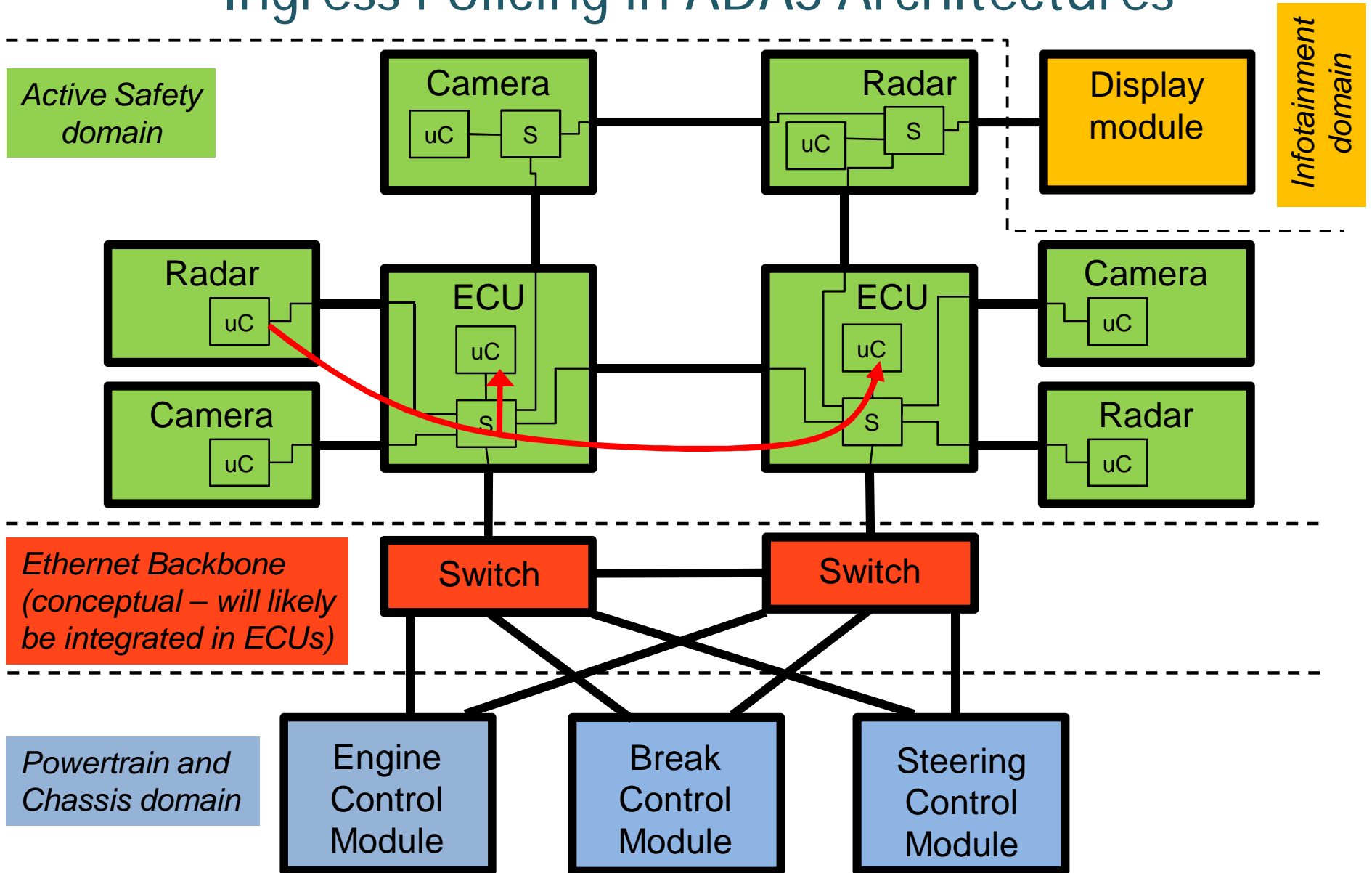
Example of Future ADAS Architecture



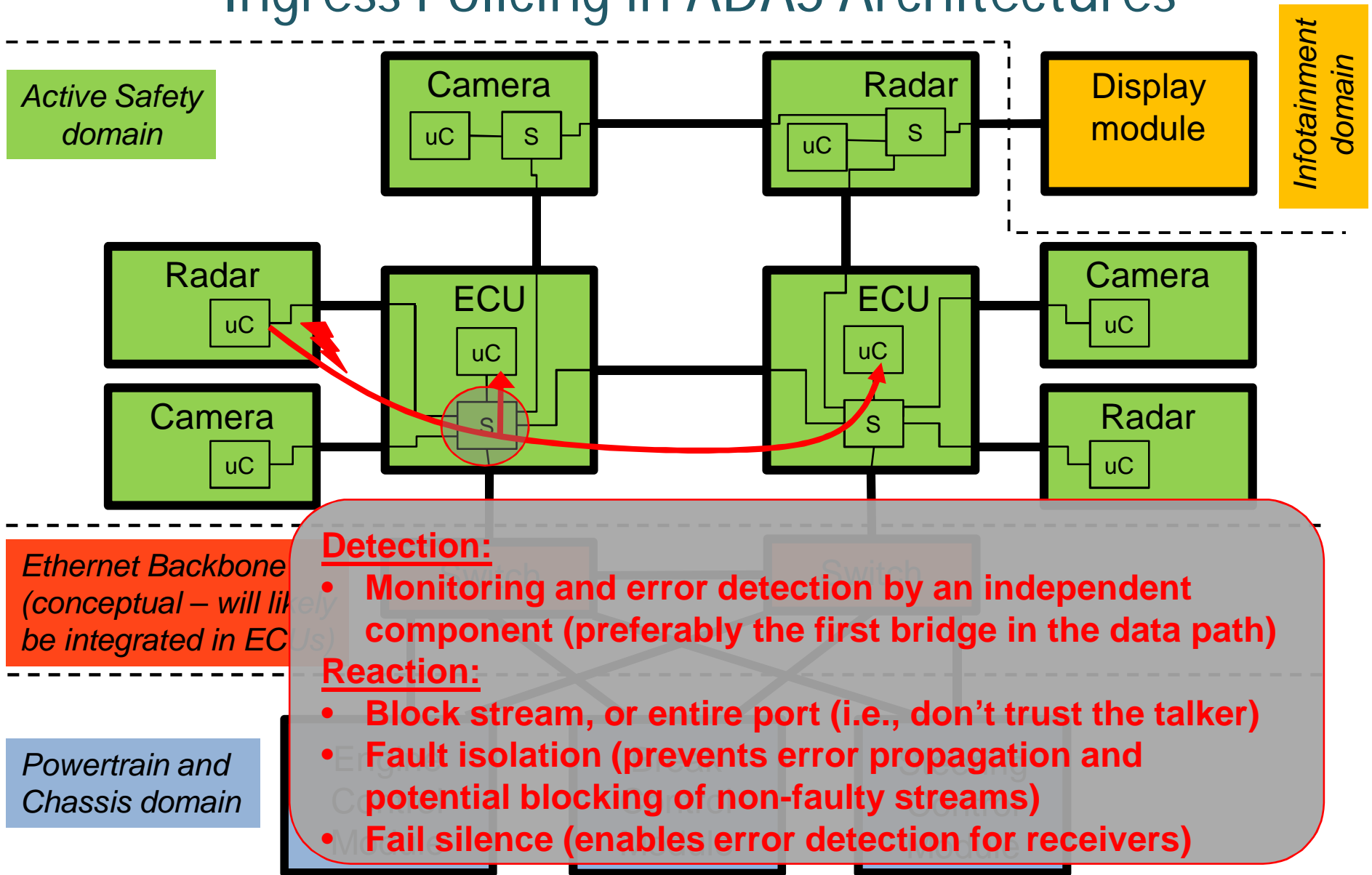
Seamless Redundancy in ADAS Architectures



Ingress Policing in ADAS Architectures



Ingress Policing in ADAS Architectures



Agenda

- Motivation and context for Ingress Policing
- ➔ Revisit conclusions from Ingress Policing analysis at Dallas Plenary November 2013
- Required properties and characteristics of ingress policing
- Discussion on how to proceed with ingress policing in TSN



Conclusions from Previous Meetings

- The following two slides are taken from Markus Jochim's presentation at IEEE 802 Plenary in Dallas, November 2013:

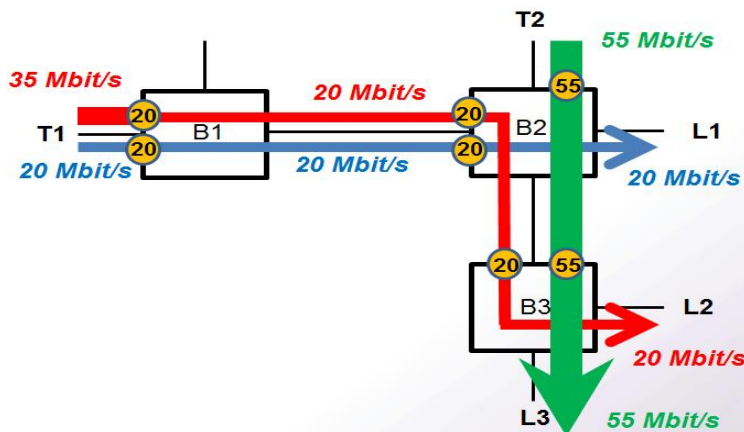
<http://www.ieee802.org/1/files/public/docs2013/tsn-jochim-ingress-policing-1113-v2.pdf>



Per Stream

(= Potentially higher number of filters per port)

Threshold Enforcing

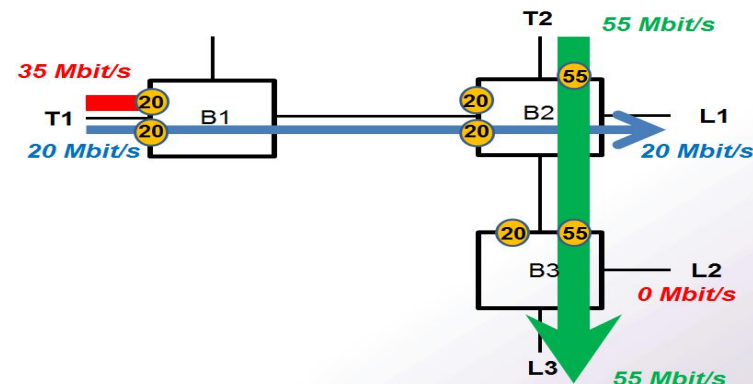


- A faulty stream sent by a faulty talker is not “silenced”.
- Other streams from faulty / fault free talkers not affected.

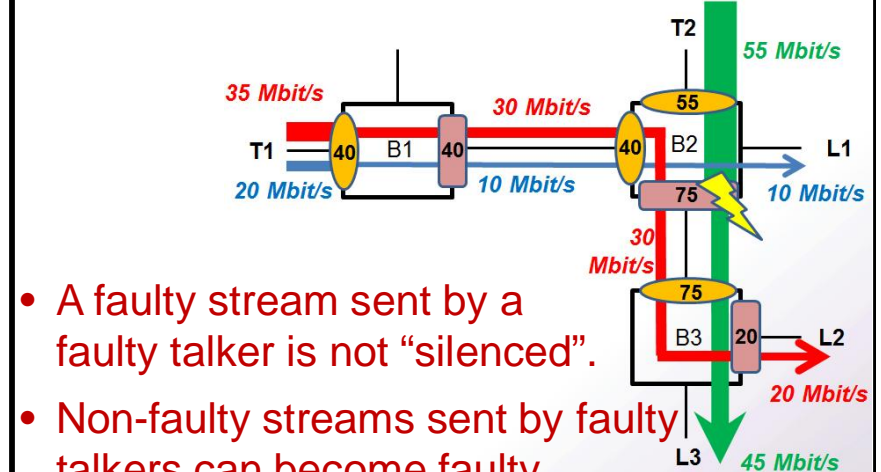
Per Class

(= Small number of filters per port)

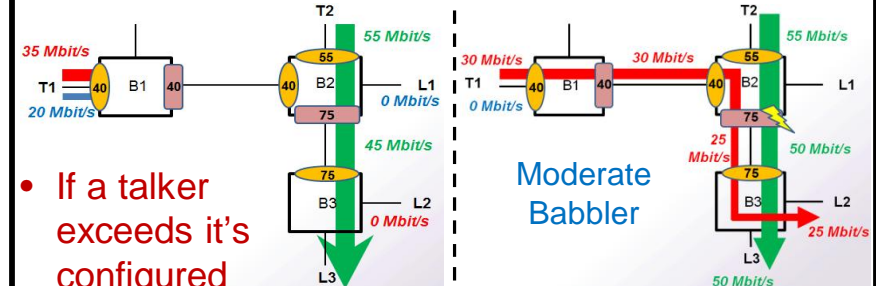
Blocking



- A faulty stream sent by faulty talker is “silenced”.
- Non-faulty streams sent by faulty talker are not necessarily silenced.



- A faulty stream sent by a faulty talker is not “silenced”.
- Non-faulty streams sent by faulty talkers can become faulty.
- A fault free stream sent by a fault free talker becomes faulty. (Fault propagation. Fault not contained)

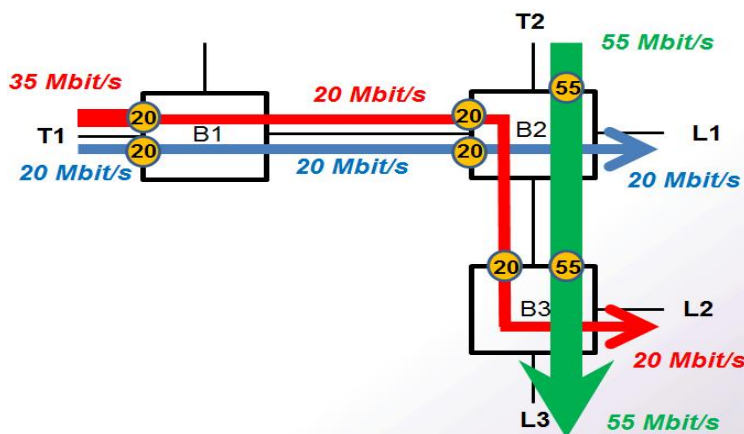


- If a talker exceeds its configured bandwidth limit, the faulty talker is “silenced”.
- In presence of a moderate babbler, a fault free stream sent by a fault free talker can become faulty. (Fault propagation. Fault not contained).
- Faulty streams sent by a faulty talker are not necessarily silenced.

Per Stream

(= Potentially higher number of filters per port)

Threshold Enforcing

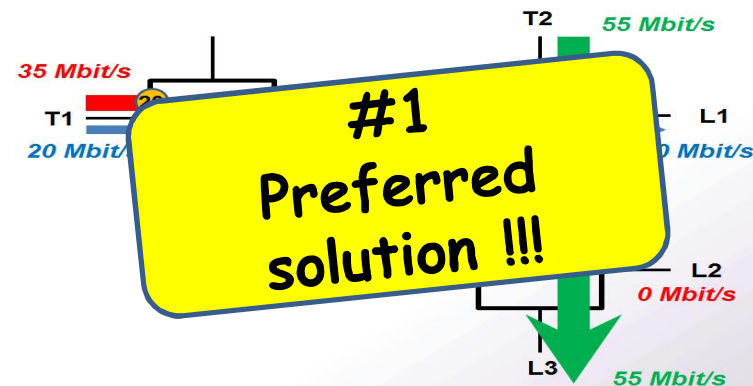


- A faulty stream sent by a faulty talker is not “silenced”.
- Other streams from faulty / fault free talkers not affected.

Per Class

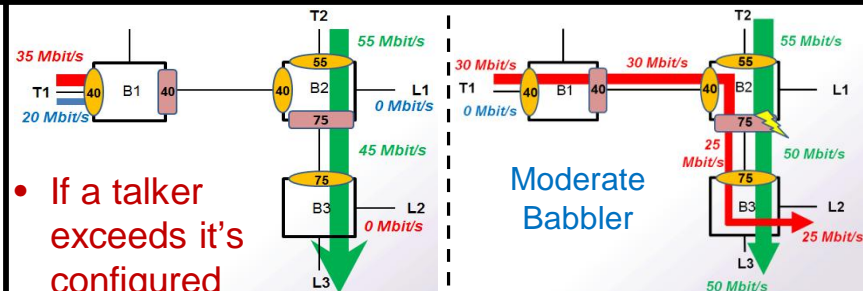
(= Small number of filters per port)

Blocking



- A faulty stream sent by faulty talker is “silenced”.
- Non-faulty streams sent by faulty talker are not necessarily silenced.

- A faulty stream sent by a faulty talker is not “silenced”.
- Non-faulty streams sent by faulty talkers can become faulty.
- A fault free stream sent by a fault free talker becomes faulty. (Fault propagation. Fault not contained)



- If a talker exceeds its configured bandwidth limit, the faulty talker is “silenced”.
- In presence of a moderate babbler, a fault free stream sent by a fault free talker can become faulty. (Fault propagation. Fault not contained).
- Faulty streams sent by a faulty talker are not necessarily silenced.

Agenda

- Motivation and context for Ingress Policing
- Revisit conclusions from Ingress Policing analysis at Dallas Plenary November 2013
- ➔ Required properties and characteristics of ingress policing
- Discussion on how to proceed with ingress policing in TSN



Ingress Policing = Error Detection + Error Handling

Error detection – desired properties [1]:

- RELIABLE: If stream is exceeding its allocated bandwidth or violating other properties as part of the traffic contract in its traffic class, then the error detection mechanism shall detect it. If the stream is staying within its allocated bandwidth and not violating the traffic contract, the error detection shall not signal an error (i.e., no false positives). (*Note: Tolerance of bandwidth monitor is tied to this property.*)
- FAST: The error detection shall with very low latency detect when streams exceed their bandwidth or violate traffic contracts
- LITTLE DISRUPTION: The error detector shall cause little disruption to the network
 - No/little influence on the normal operation of a bridge or network (e.g., CPU resources, delays in forwarding process, ...)
 - No/bounded influence of faulty streams on non-faulty streams in the network

[1] Error detection properties in distributed systems: Leners et al., “Detecting failures in distributed systems with the FALCON spy network,” Proc. of the 23rd ACM Symposium on Operating Systems Principles, 2011.



Ingress Policing = Error Detection + Error handling

Error handling (reaction):

- Configurable among the following alternatives:
 - Block stream only (e.g., isolate only the faulty sensor but still provide the remaining data of the sensor network to the application, enabling controlled transition to safe state)
 - Block entire ingress port (e.g., the faulty behavior of a sensor may make a set of sensor data obsolete, thus blocking an entire port; another argument is that for some critical sensors we cannot continue to trust the device in case one of its streams is faulty)
 - Enforce threshold for the faulty stream (there may be cases where the data is still useful to the application; there may be time intervals where blocking is not an option).



Detection and Reaction

Immediate Reactions

- Fine grained (per stream) without delays:
 - Threshold Enforcing (delaying/blocking individual frames)
- Permanently assures QoS for fault free streams in presence of faulty streams.

Detection Requirements

- Requires fast detection at least on a per packet granularity level (traffic class dependent) to assure immediate reaction

Isolation Reactions

- Coarse grained reactions:
 - Stream blocking
 - Port blocking
- Isolates faulty components (e.g., to avoid single-point failure)
- May be complemented by reconfiguration/mode changes.

Detection Requirements

- Requires unambiguous identification of faulty component/must avoid false positive isolation decisions
- Identification and detection by first bridge in the data path (i.e., at the first hop)




Precision of Stream-Based Detection

- The bandwidth threshold to be monitored shall be of similar granularity of the stream reservations
- If not: We need to make larger per-stream bandwidth reservations only due to limited error-detection capabilities – this is not an issue for functional safety but reduces significantly the available bandwidth for regular data communication.



Ingress Policing in AVB?

- We currently implement stream-based ingress policing with blocking:
 1. software on an external microcontroller
 2. proprietary non-standardized capabilities in our AVB Ethernet switches
 - This will only work for a specific use case and is not acceptable in the long run for ADAS, especially considering that the demand of fail-operational ADAS applications is growing
-  *We need standardized solutions*

Ingress Policing in AVB?

- We currently implement stream policing with...

It's the same story for Seamless Redundancy:

- We are implementing an application-specific proprietary solution on the external microcontrollers in combination with certain switch configurations
- ... while waiting for 802.1CB and TSN

... that the demand of fail-operational ADAS applications is growing

 *We need standardized solutions*



Agenda

- Motivation and context for Ingress Policing
- Revisit conclusions from Ingress Policing analysis at Dallas Plenary November 2013
- Required properties and characteristics of ingress policing
- ➔ Discussion on how to proceed with ingress policing in TSN



Conclusion

- Ingress Policing has been discussed for a long time in the TSN group and is asked for by at least two industries
- The success of Ethernet in Automated Driving, Active Safety, and broadly in ADAS depends on mechanisms like Seamless Redundancy and Ingress Policing becoming available
- Ingress policing capabilities:
 - Error detection: Per-stream monitoring and error detection is a must in future fail-operational ADAS applications. Monitoring shall be precise, same order as stream reservations.
 - Reaction: Multiple alternatives must be available: Block individual streams, block entire port, and enforce threshold.
- *Discussion: How do we proceed with standardizing appropriate ingress policing capability?*

