### AVB for low latency / industrial networks:

# Redundancy for fault tolerance and AVB – possible future of a „lightweight“ SRP


**Oliver Kleineberg, Hirschmann Automation & Control**

(oliver.kleineberg@belden.com)


**IEEE 802 Plenary Meeting, November 2011, Atlanta, USA**

# Agenda

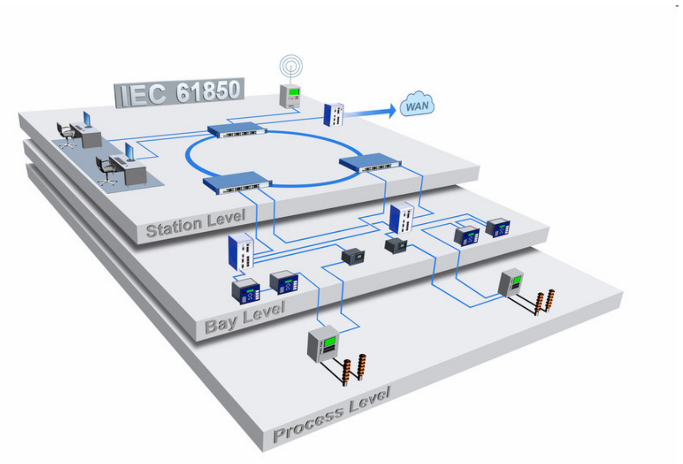
## Agenda:

- Ethernet and new application fields: lessons learned from past years
- How to prepare „SRP Generation 2" to cope with requirements to come?
- Future-proof SRP: „Layering" and interfaces to other technologies

# Ethernet and new application fields: lessons learned from past years

# Ethernet and new application fields: lessons learned from past years

- In the past 10+ years, Ethernet has migrated into several new application fields. Examples:



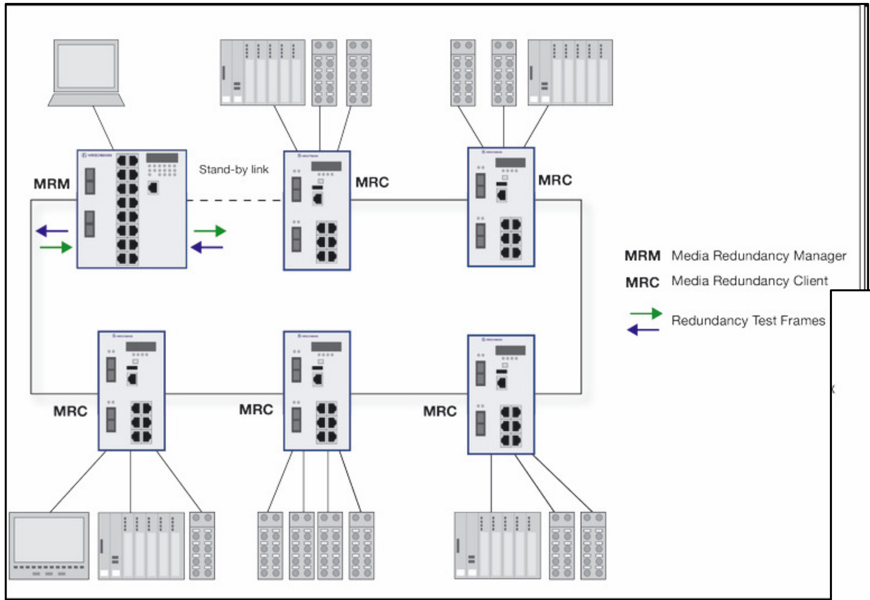

Industrial Automation & Control

Power Utility Automation

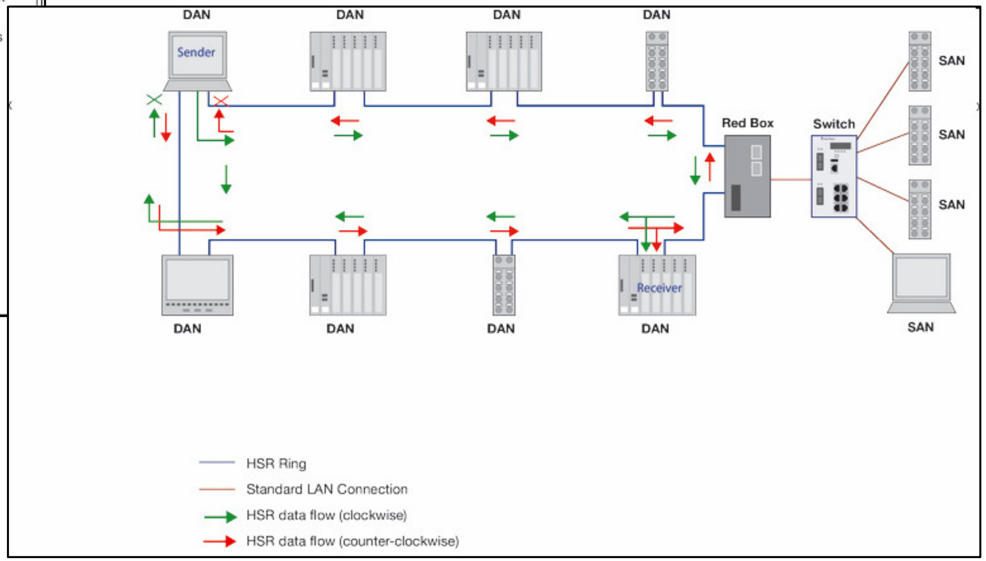
Traffic Control Systems

# Ethernet and new application fields: lessons learned from past years

- For most new application fields, a specific “enabler” technology needed to be developed. Examples:




Industrial Automation & Control: Deterministic redundancy recovery and VLANs/CoS




Power Utility Automation: Seamless Redundancy Concepts

# Ethernet and new application fields: lessons learned from past years

- But: The speed of Ethernet migration into new application fields is increasing in the last years at a very high level. New possible application fields include*:
  - Medical applications
  - Military, aeronautic and astronautic control
  - Safety-critical applications
  - Vehicle/car networks
  - Building automation
  - … (insert arbitrary mission-critical application here)
- AVB (Gen.2) mechanisms will be instrumental as "enablers" for many new application fields… but only if they are flexible enough to allow other "enablers" to work as well!

*for some applications, there are already niche technologies installed, e.g. AFDX in aeronautic control

# Ethernet and new application fields: lessons learned from past years

Assumptions:

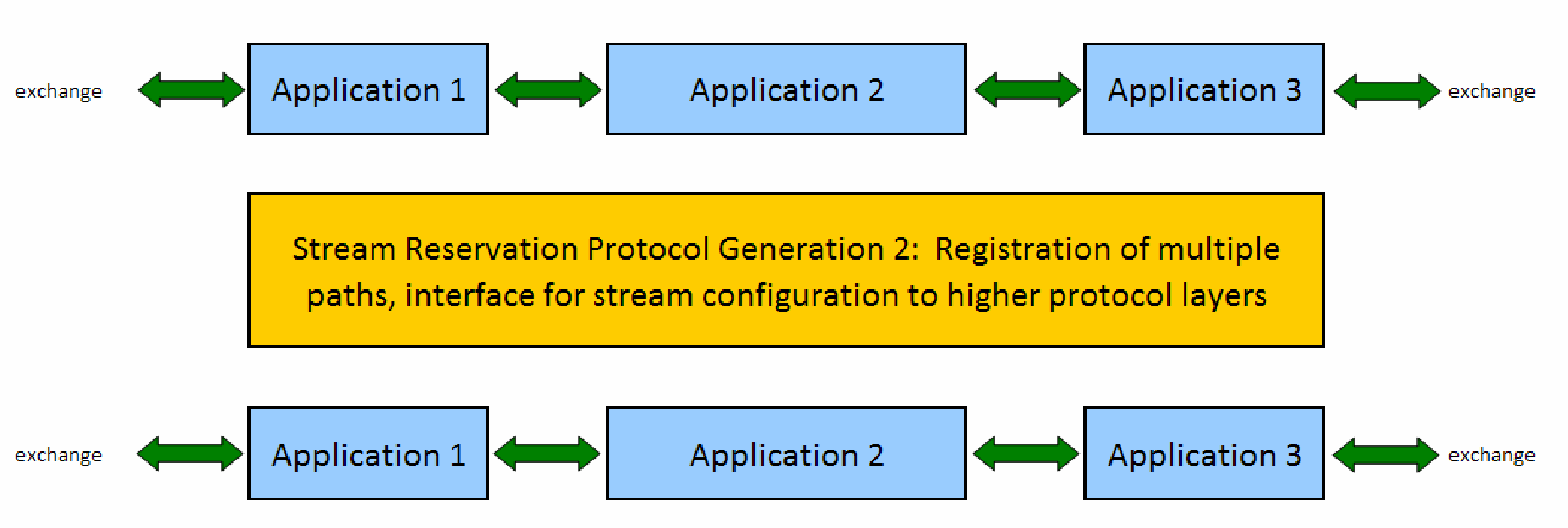
- Standardization of Ethernet technology is still important (probably now more than ever), but new and very specific application demands will still outpace standardization or will never be truly standardized (in the near future)
- Especially with mission-critical applications, fault-tolerant design is paramount, it will be a basic requirement of most future Ethernet-based communication systems
- SRP needs to be able to adapt flexibly to new application requirements because it is instrumental in enabling fault-tolerant low-latency network design and needs to be able to work with today's and future Ethernet systems
- As applications come (and go), requirements and technologies on “higher” and “lower” layers of the overall architecture change. SRP must allow for those changes.

# How to prepare „SRP Generation 2“ to cope with requirements to come?

**How to prepare „SRP Generation 2“ to cope with requirements to come?**

# How to prepare „SRP Generation 2“ to cope with requirements to come?

- Layering: To allow SPR to work with new (probably application-specific) technologies, the most important aspect of SRP Gen.2 is to allow those technologies to “attach” themselves loosely “on top” or “bottom up”




- SRP main extensions:
  - Registration of streams through all available paths
  - Service interface to higher layers to allow control of stream registration and transmission, (worst case) latency surveillance, etc…

# Future-proof SRP: „Layering“ and interfaces to other technologies

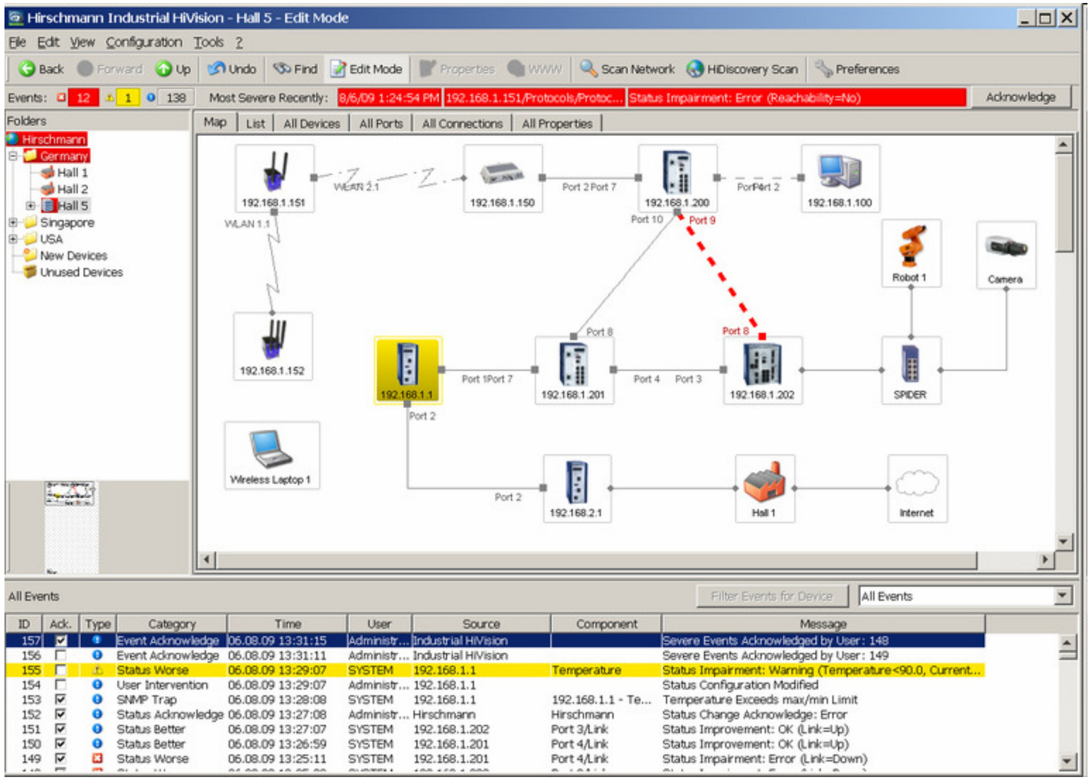
**Future-proof SRP: „Layering“ and interfaces to other technologies**

# Future-proof SRP: „Layering“ and interfaces to other technologies

- SRP is in a challenging situation: It needs to gain knowledge of all available paths and needs to be able to register streams over all paths
- Yet, it needs to offer the possibility of influencing stream registration and stream flow if needed.

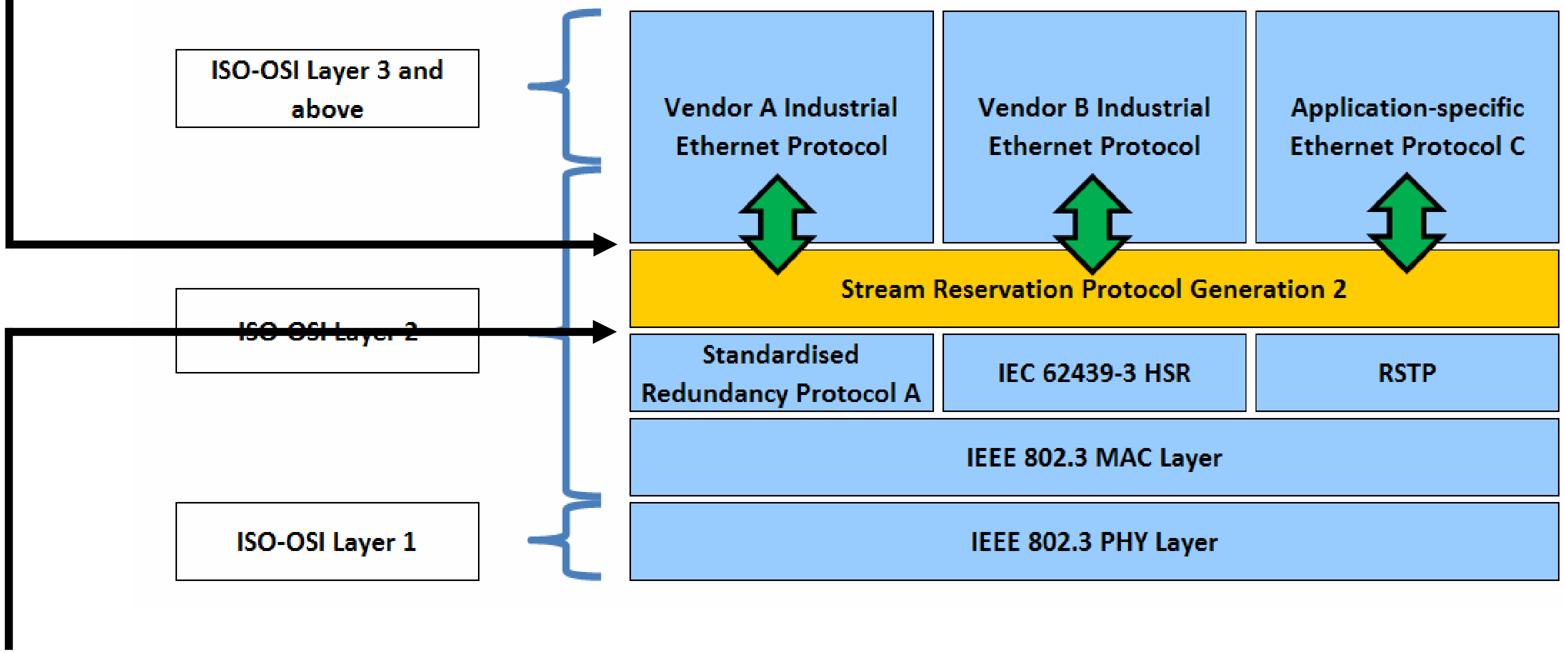
look ahead:

Probably the most important “single” piece of information: topology knowledge



# Future-proof SRP: „Layering“ and interfaces to other technologies

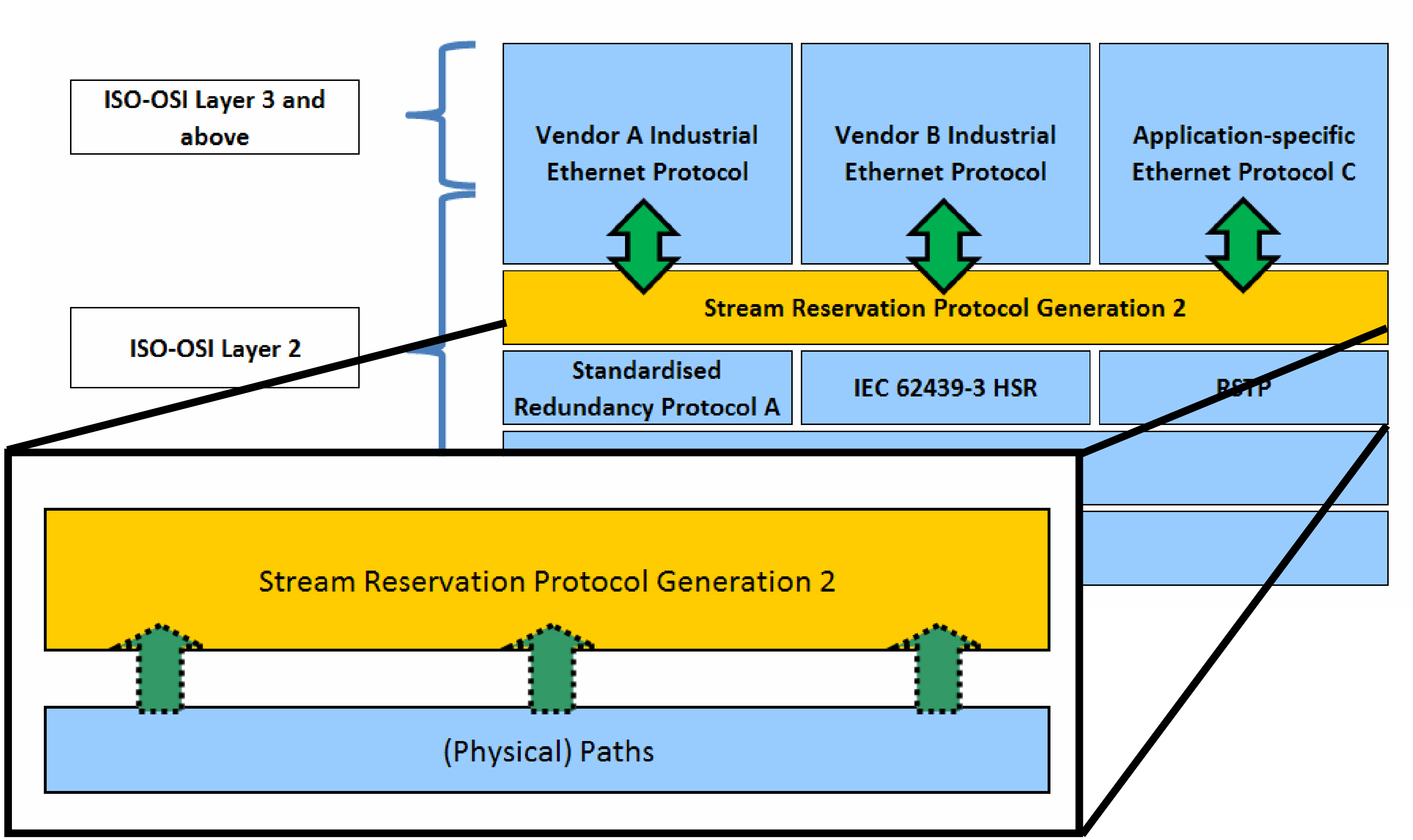
- SRP needs to offer an interface to higher layers to enable stream arbitration and control. → Interfacing to the application




- SRP needs to be able to operate with arbitrary (physical) topologies. These topologies are dependant on the redundancy control protocol, e.g. RSTP

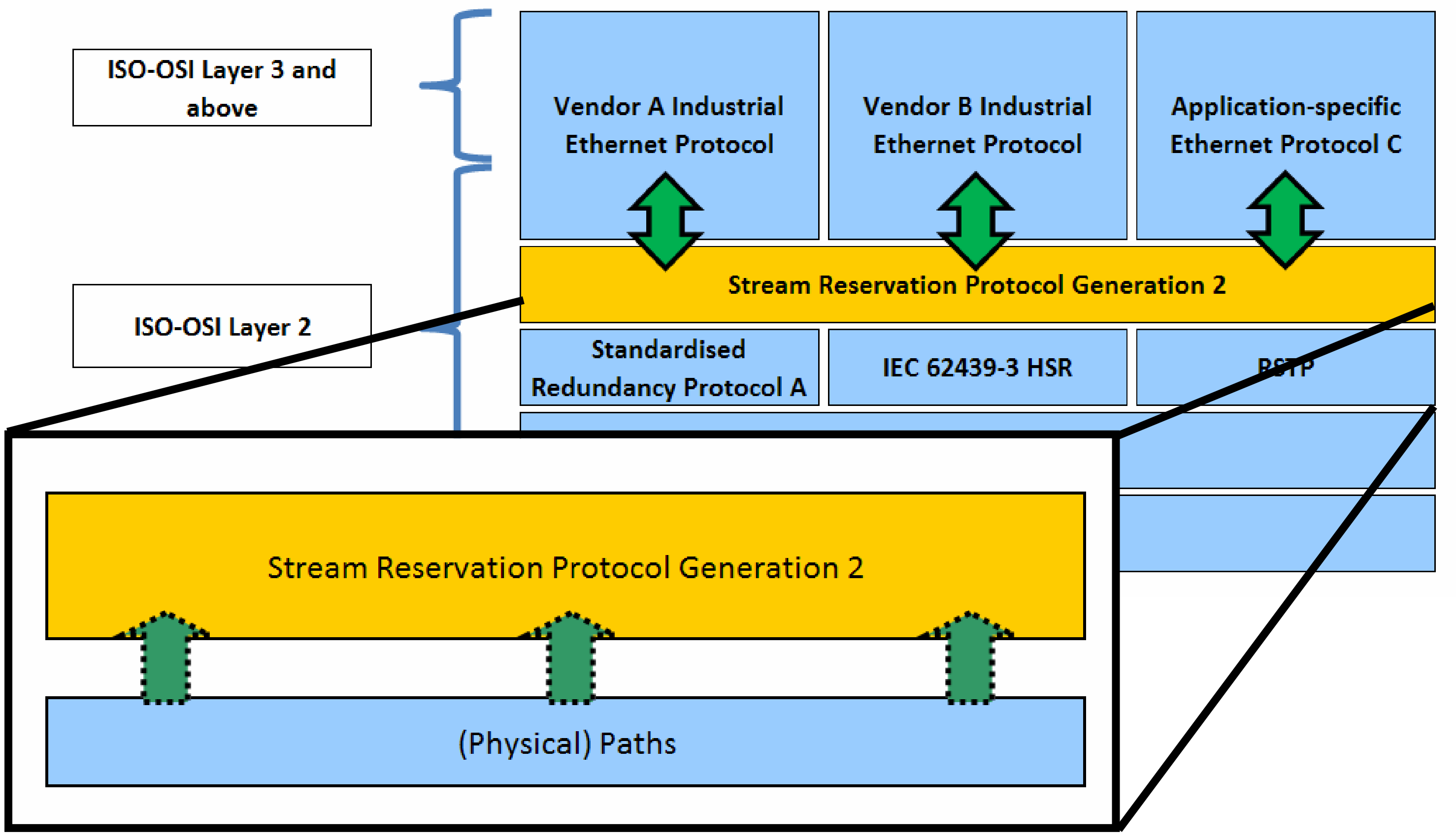
  → Abstraction from the redundancy control protocol

# Future-proof SRP: „Layering“ and interfaces to other technologies




- Physical (and Logical) Topology are imposed on SRP
- SRP Gen.1 still follows the RSTP logical topology
- SRP Gen.2 observes and registers streams on all available paths, ignoring discarding ports for stream registration

# Future-proof SRP: „Layering“ and interfaces to other technologies

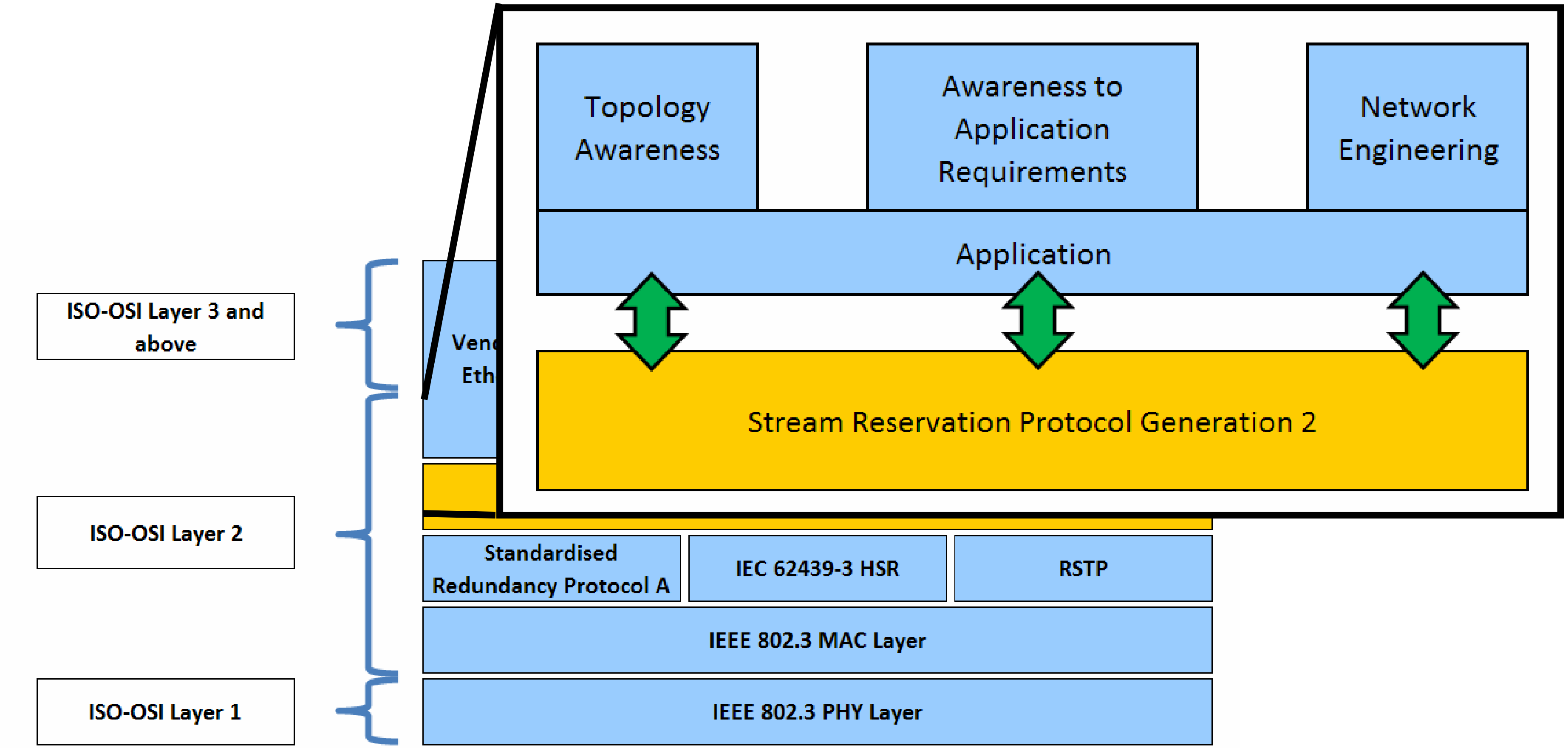



- For Gen.2 registration of multiple paths, see [slides_singapore]
- This allows SRP to achieve the switchover times that are in line with the underlying redundancy control protocol. (e.g. IEC-HSR, RSTP, SPB,…)
- The used redundancy protocol depends on application requirements

[slides_singapore] http://www.ieee802.org/1/files/public/docs2011/at-kleineberg-AVB-media-redundancy-0311-v02.pdf

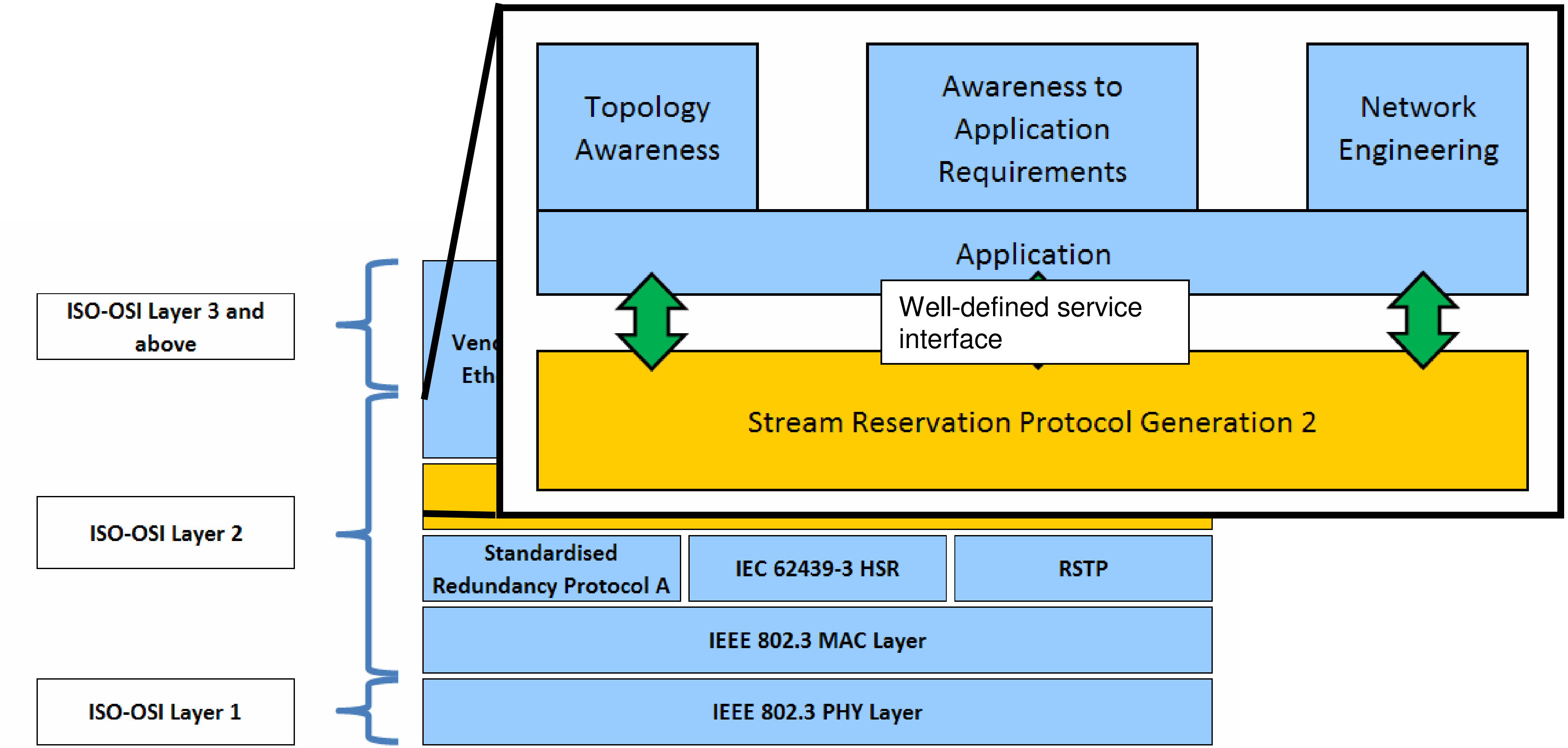
# Future-proof SRP: „Layering“ and interfaces to other technologies

- Higher Layer entities usually have a complete topology awareness (e.g. Industrial Engineering Tools, SCADA systems, …)
- Topology awareness and application req. awareness are used to configure / engineer stream flows



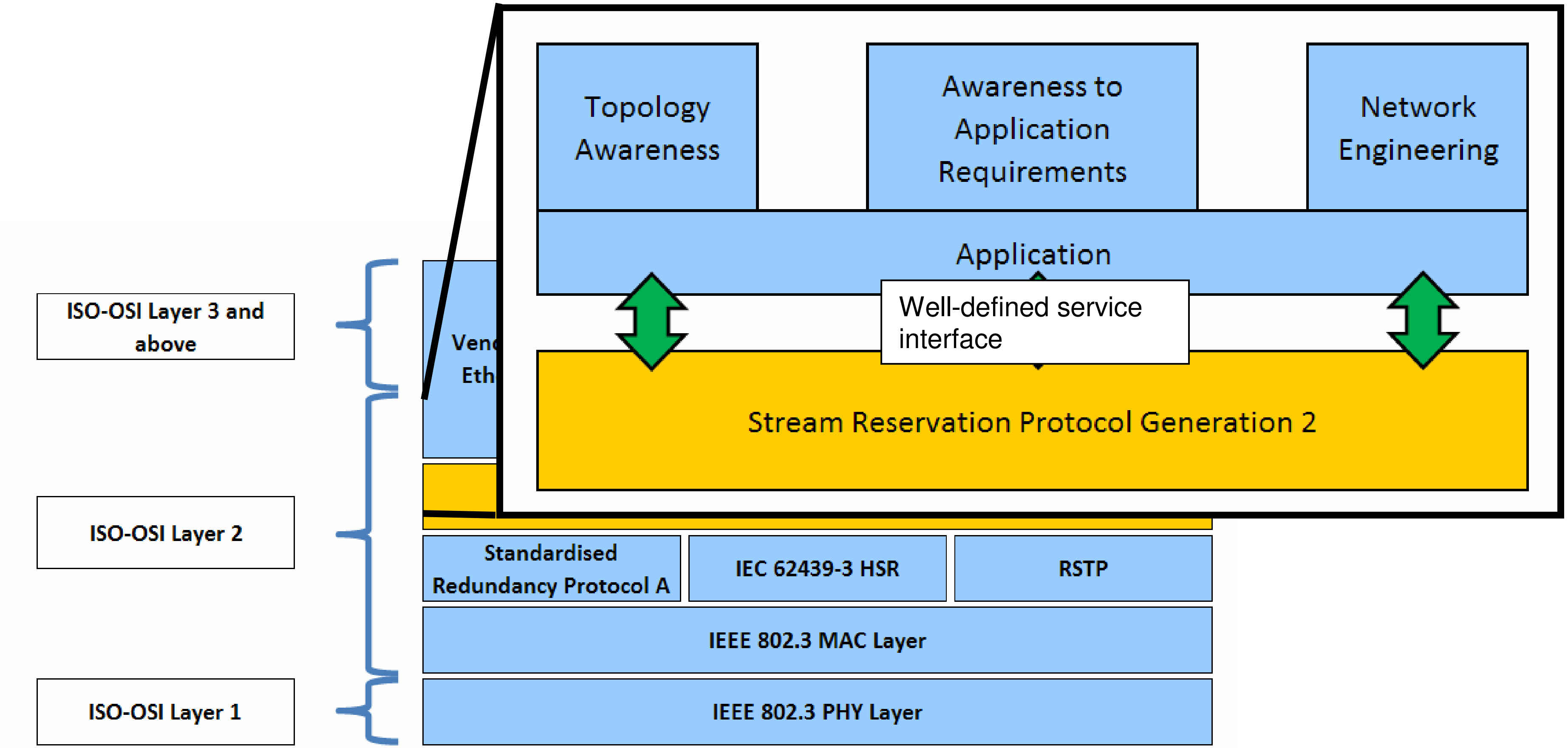
# Future-proof SRP: „Layering“ and interfaces to other technologies

- Higher Layer entities can: enable or disable streams entirely, control stream flow through enabling/disabling bridge ports, etc…
- Higher Layer entities are provided with information on streams and configure SRP through a well-defined service interface



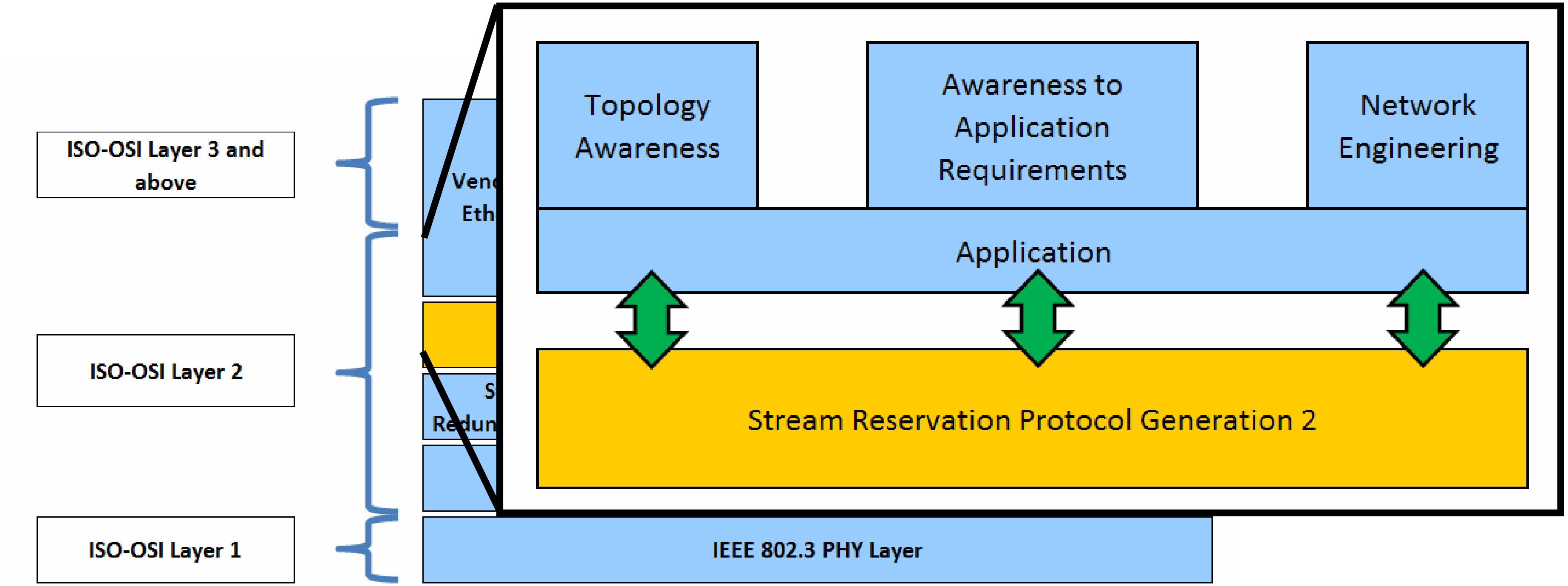
# Future-proof SRP: „Layering“ and interfaces to other technologies

- Information from SRP: e.g. Maximum worst case latency from talker to listener, based on multiple paths (i.e. all latency information for all paths registered)
- Each SRP Gen.2 device must provide worst case latency information independently and the worst case must observe all paths from talker to device



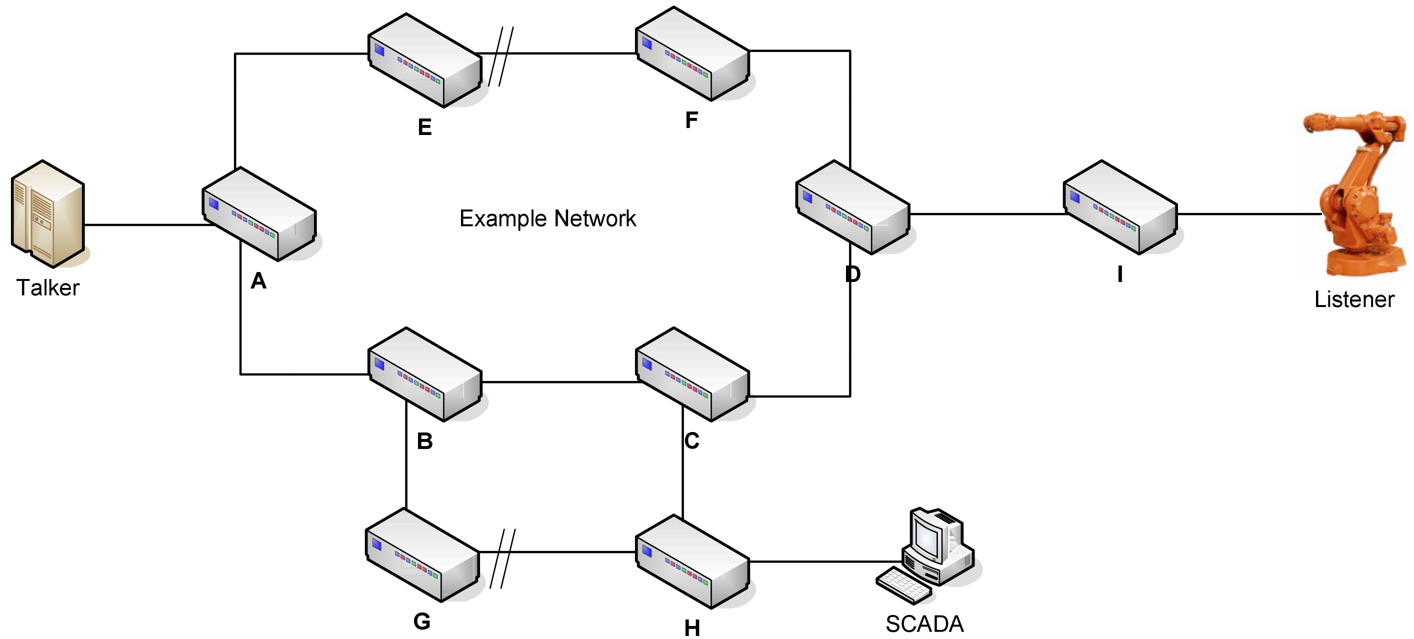
# Future-proof SRP: „Layering“ and interfaces to other technologies

- Higher layer entities could be (also see [slides_sanfrancisco]):
  - Not present at all (in that case, streams on all paths will be registered)
  - Automated or non-automated network engineering (e.g. Industrial Ethernet Engineering tool with algorithmic support)
  - Fixed configuration (for 100% static network configurations e.g. automotive networks)
  - …



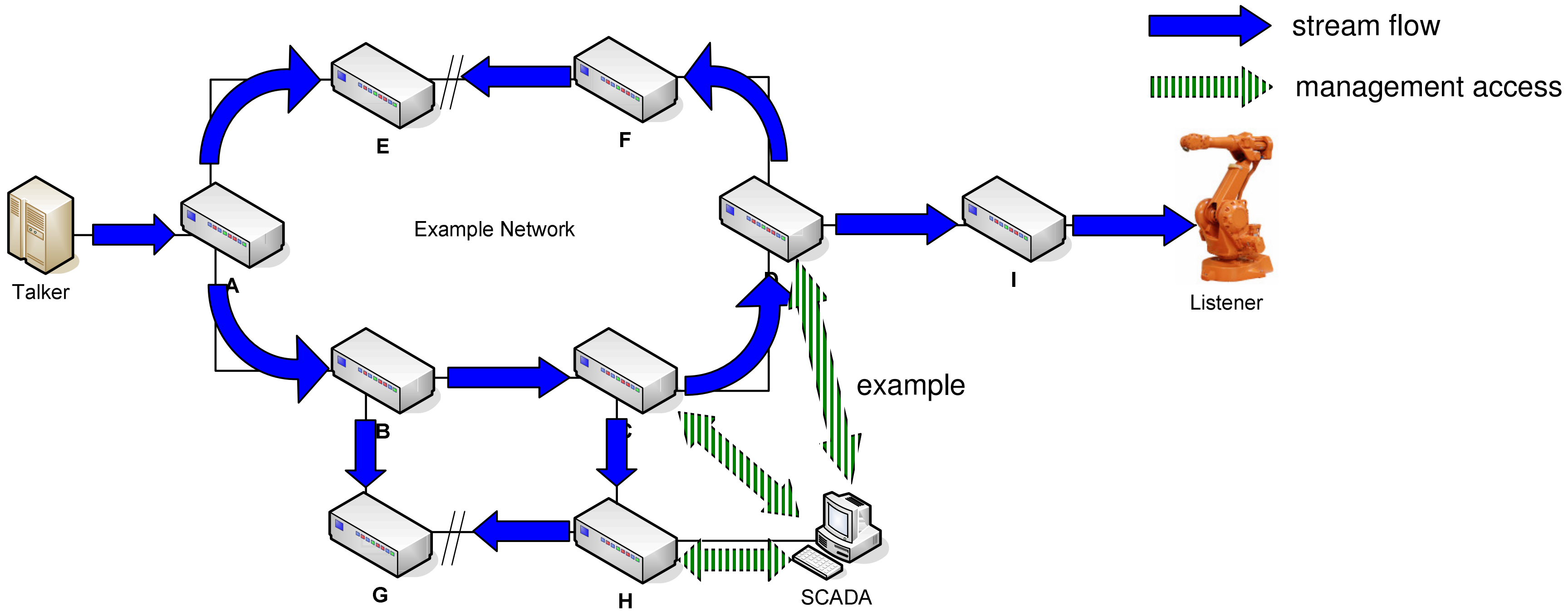
# Example application

# Example application




- A single talker and listener (Industrial control application) want to communicate through a fault-tolerant network
- The redundant paths in the network are administrated by RSTP
- A SCADA system is in place as an engineering workstation. It has full topology knowledge (e.g. through SNMP and LLDP) and management access to all bridges in the network (e.g. through SNMP)

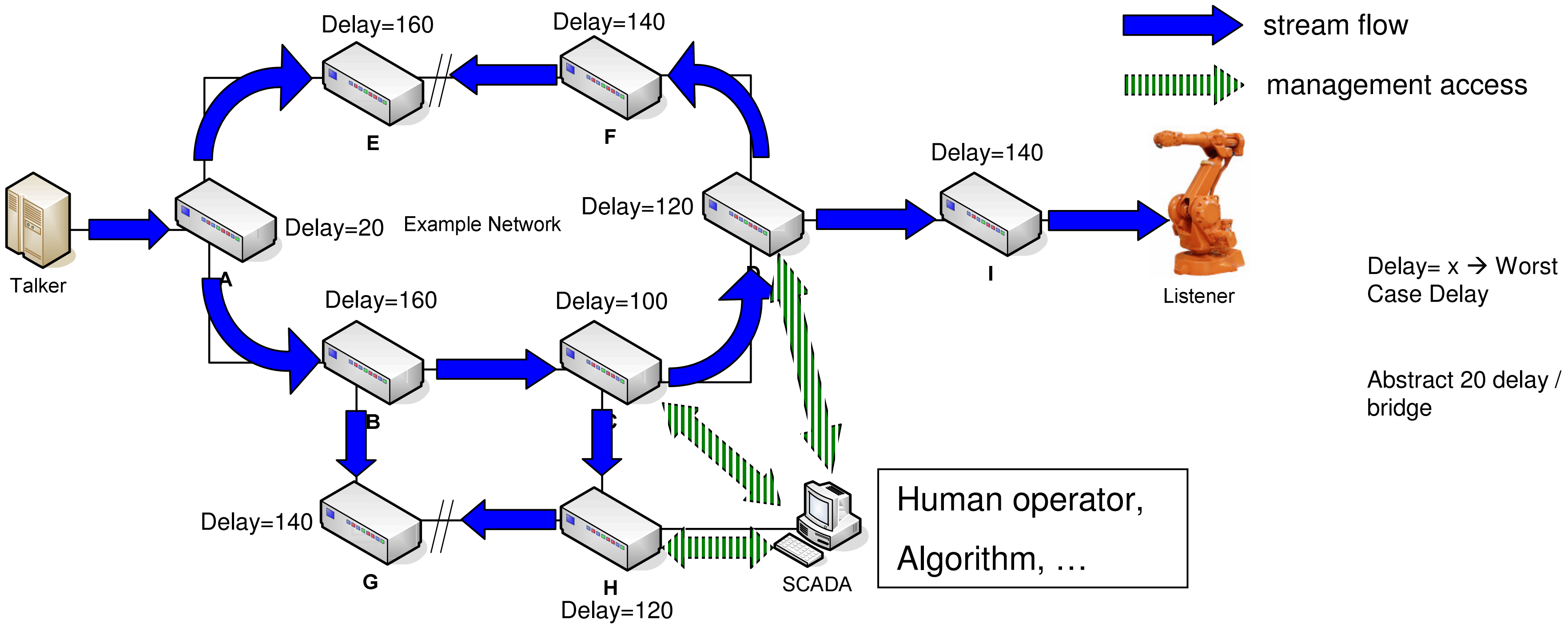
# Example application




1. SRP registers the redundant stream on all available paths from talker to listener (details on how this might be done: [slides_singapore])
2. The SCADA system collects all stream data from all the bridges, e.g. stream data, latency… (Only exemplary access is shown above to not overburden the picture)

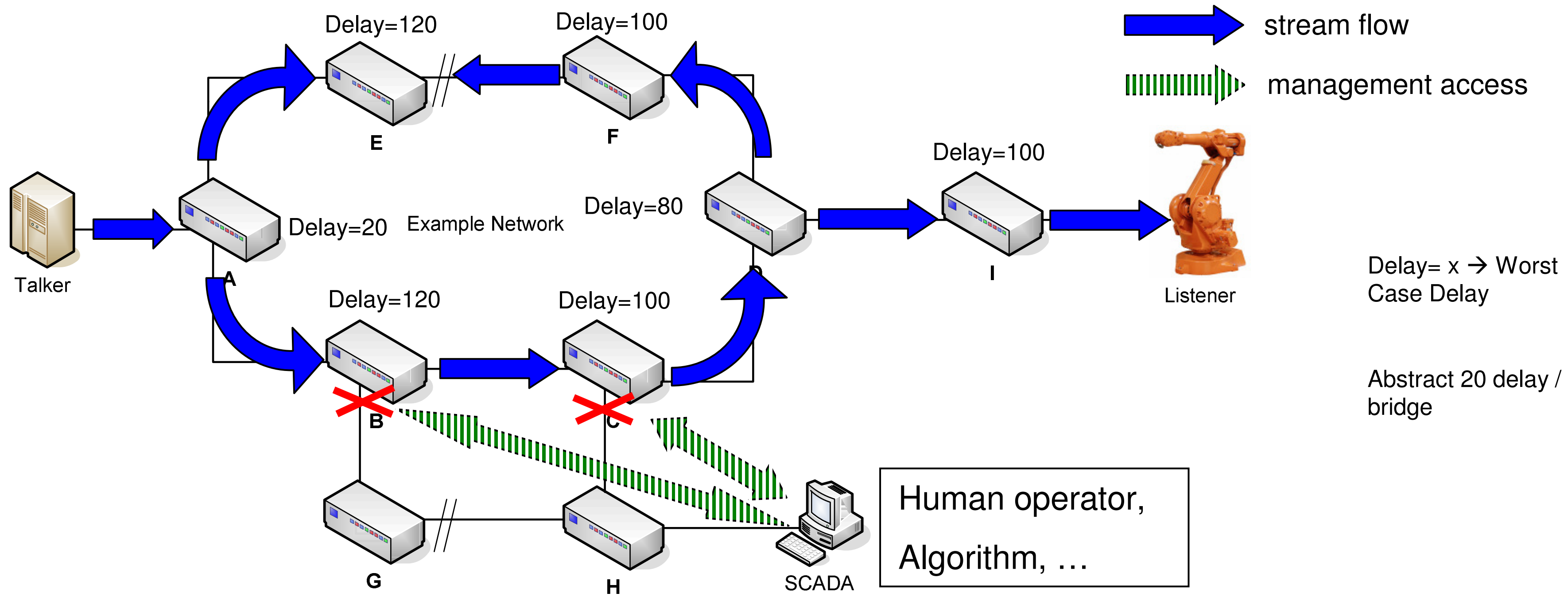
[slides_singapore] http://www.ieee802.org/1/files/public/docs2011/at-kleineberg-AVB-media-redundancy-0311-v02.pdf

# Example application




3. The SCADA system displays the topology, together with the stream flows and the worst case latency (when more than one path is available)
4. From this information, a human operator through a network engineering tool, an algorithm with application-specific knowledge, etc… can influence which paths are to be configured for stream transmission

# Example application




5. In this case, a human operator decides to cut off the sub-ring through G and H from this stream to reduce worst case latency, at the cost of some fault-tolerance
6. In another use case (and with other requirements in the background), the outcome of this decision could have been different!... E.g. when the additional fault-tolerance outweighs the

# FIN

**Thank you for your attention!**