Time Sensitive Networks
(TSN)
Overview

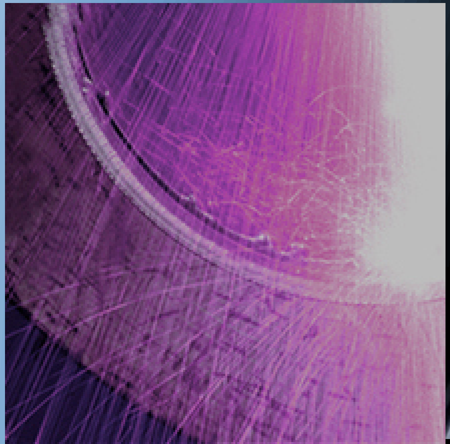
Bob Noseworthy

University of New Hampshire
InterOperability Laboratory

(UNH-IOL)

For the uninitiated, the curious and the bold

Last Updated:
November 24, 2015



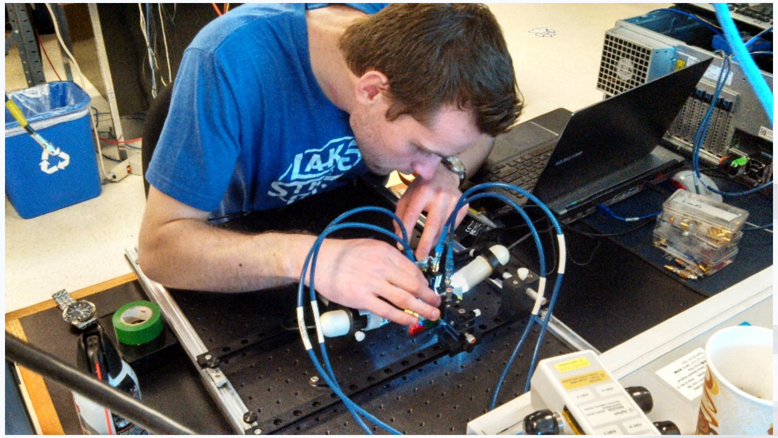
Outline

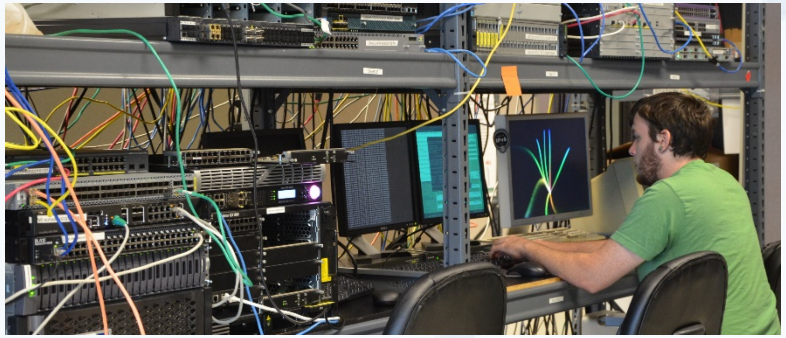
What's in a name?

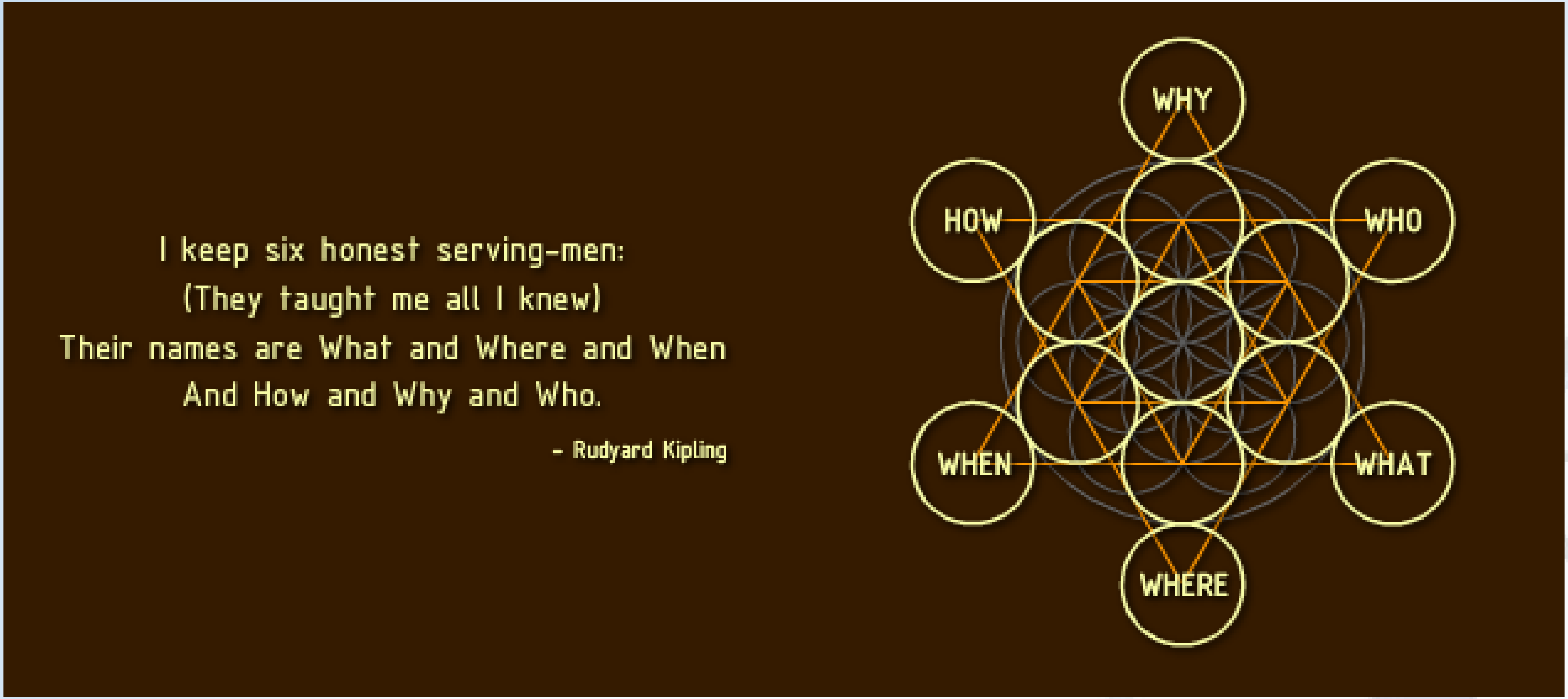
TSN in < 100 slides

What, Where, Who, Why, How

When: The future of TSN






I keep six honest serving-men:
(They taught me all I knew)
Their names are What and Where and When
And How and Why and Who.
- Rudyard Kipling
WHY
HOW
WHO
WHEN
WHAT
WHERE

WHAT

What's in a name?

They're all the same
Or closely related

DETERMINISTIC NETWORKING

INTERNET OF IMPORTANT THINGS (IOIT)

INTERNET OF THINGS (IOT)

CONNECTED CAR

INTERNET OF EVERYTHING (IOE)

TIME SENSITIVE NETWORKS (TSN)

AUDIO VIDEO BRIDGING (AVB)

DETERMINISTIC ETHERNET

INDUSTRY 4.0

SENSOR FUSION

CYBER-PHYSICAL SYSTEMS (CPS)

INDUSTRIAL INTERNET OF THINGS (IIOT)

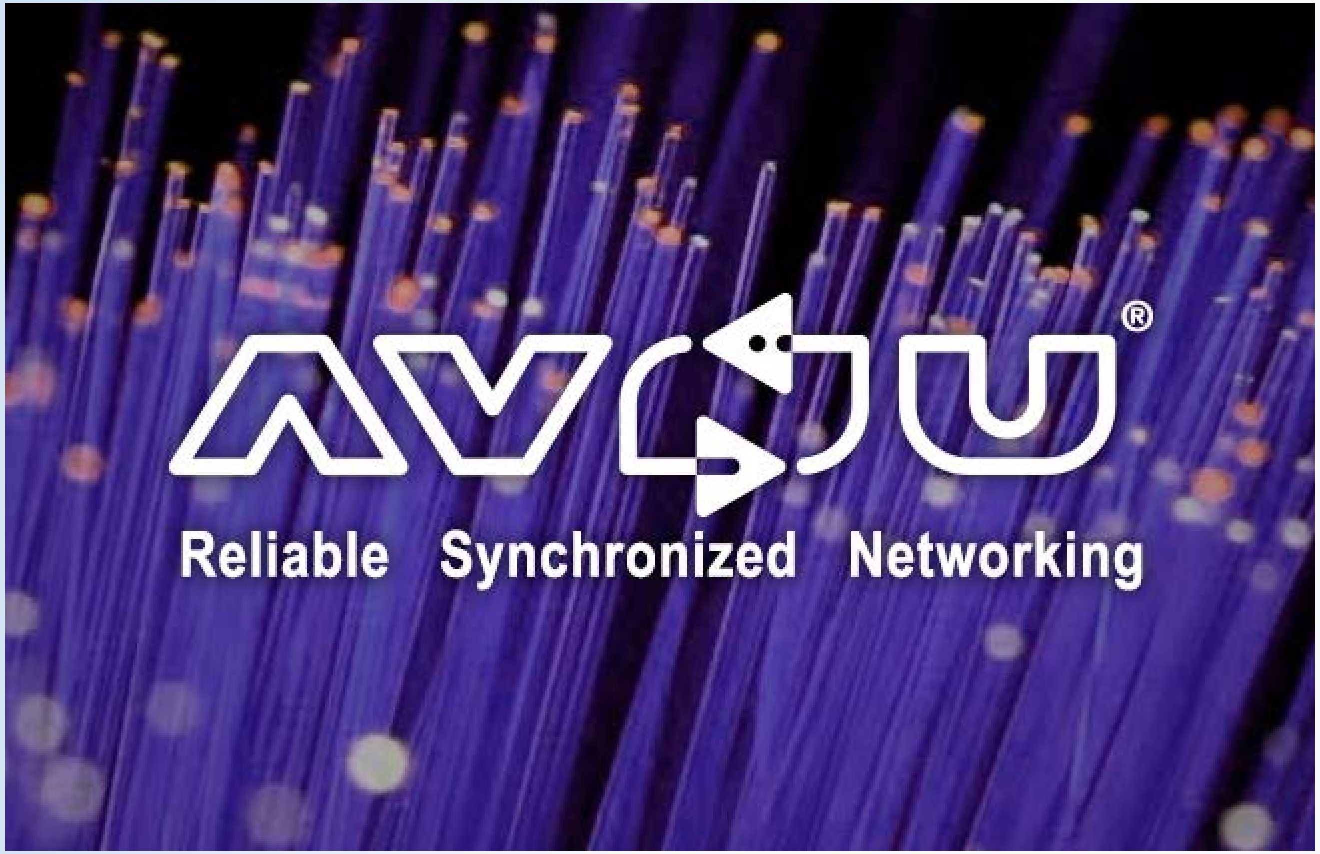
INDUSTRIAL INTERNET

MACHINE-TO-MACHINE(M2M)

WHERE / WHO

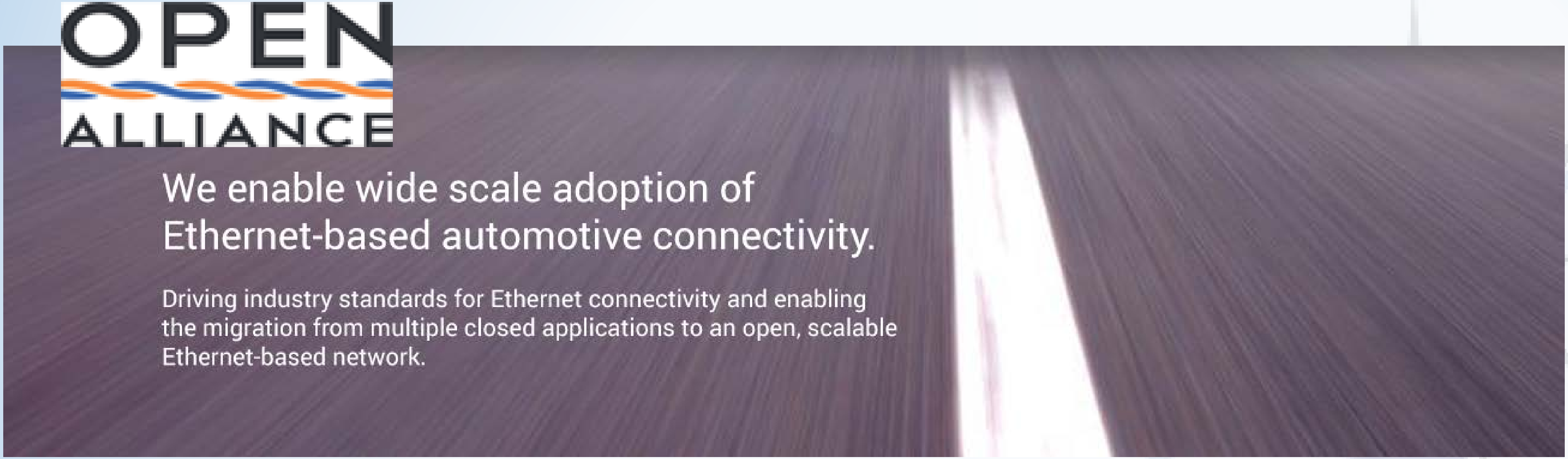
The following is an incomplete listing of related activities to demonstrate the scope breadth and depth of the current industrial activity

Get involved: Industry Forum – AVnu Alliance



- http://avnu.org/

Get involved: Industry Forum – OPEN Alliance



- http://www.opensig.org/

Get involved: Industry Forum – IIC

Professionals were asked to identify the **Biggest Challenge** Facing the Industrial Internet: **77% said Interoperability** (source: IoT Nexus)

- http://www.iiconsortium.org/index.htm
- http://www.iiconsortium.org/pdf/Industrial_Internet_Revolution_Infographic.pdf

Get involved: Conference – ISPCS 2016

International Symposium on Precision Clock Synchronization

- http://www.ispcs.org/
- http://www.ispcs.org/2015/files/R1_ISPCS-2016-Invitation.pdf

Get involved: Conference – WSTS 2016

Workshop on Synchronization in Telecommunication Systems

- https://www.atis.org/WSTS/
- https://www.atis.org/WSTS/2015/2015documents.asp
- https://www.atis.org/WSTS/2014documents.asp
- Highly recommended: https://www.atis.org/WSTS/papers/3-3-1_UCBerkeley_Lee_LeveragingClocks.pdf
- Tutorials galore (see 2015 documents for full list):
  - https://www.atis.org/WSTS/2015/papers/0-1_Shenoi_Qulsar_Fundamentals_and%20Clocks_WSTS-2015.pdf

Get involved: NSF CPS Virtual Organization

- http://cps-vo.org/

Get involved: TAACCS

Time-Aware Applications, Computers, and Communication Systems (TAACCS)

- http://www.taaccs.org/index.html

NIST TAACCS Whitepaper:

- http://nvlpubs.nist.gov/nistpubs/TechnicalNotes/NIST.TN.1867.pdf

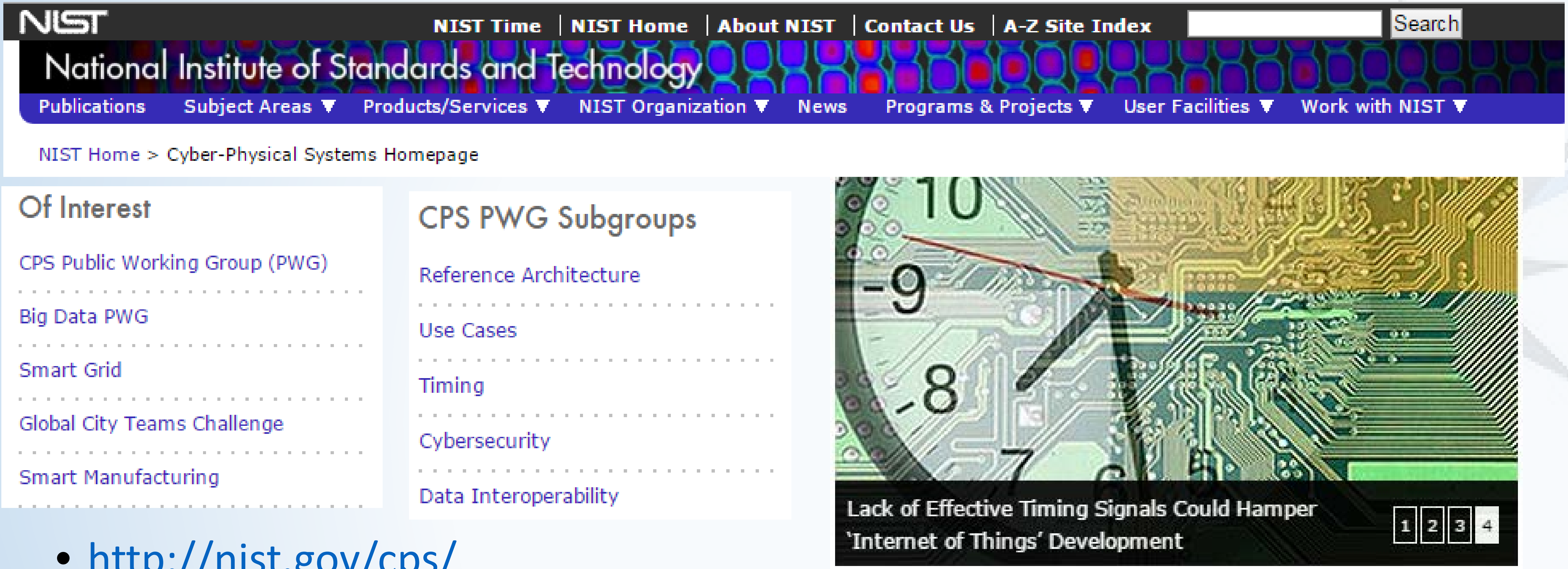
Get involved: CPS Week 2016

Medical Cyber Physical Systems Workshop 2016

Medical Device Interoperability, Safety, and Security Assurance

- http://www.cpsweek.org/2016/
- http://mlab.github.io/medcps_workshop/

Get involved: NIST CPS Public Working Group



- http://nist.gov/cps/
- http://www.nist.gov/pml/div688/timing-031915.cfm

Get involved: Conference – TSNA

Date: April - TBD - in San Jose, CA

- http://www.tsnaconference.com/
- https://goo.gl/r200rI
-

Get involved: Deterministic Ethernet Forum

2016 Dates TBD

- https://www.de-forum.com/
- https://www.de-forum.com/presentations/

Get involved: Standards – IEEE 802.1 / 1588

IEEE

IEEE 802

802 TSN time sensitive networks

- http://www.802tsn.org/
- http://www.ieee802.org/1/
- http://www.ieee802.org/1/pages/tsn.html
- https://ieee-sa.imeetcentral.com/1588public/

Get involved: Standards – IETF DetNet

I E T F

Deterministic Networking (detnet)

- https://datatracker.ietf.org/wg/detnet/charter/
- https://www.ietf.org/mailman/listinfo/detnet

Get involved: Standards – Others

- SMPTE: https://www.smpte.org/
- SAE (Society of Automotive Engineers (+Aerospace)): http://www.sae.org
- Open Platform Communications (OPC) https://opcfoundation.org/about/opc-technologies/opc-ua/
- IEEE P2413: http://standards.ieee.org/innovate/iot/
- And many more: http://standards.ieee.org/innovate/iot/stds.html

- Source: Paul Didier (Cisco) TSNA’15 - Survey of IoT Consortia and Community.pdf

Harbor Research

Postscapes
Tracking the Internet of Things

Industrial Standards Protocols

ProSTEP iViP

OPC Foundation
OMG DDS
FDI Cooperation
FDT Group

Administration and Data

Automation and Control

SAE International
OBDII
sercos the automation bus
CC-Link IE
CC-Link
Modbus TCP
EtherNet/IP
PROFINET
FlexRay
EtherCAT Technology Group
DeviceNet
Modbus
HART Communication Foundation
ISA100Wireless
WirelessHART
BACnet
dnp
DNP3
CANopen
CAN-bus
Ethernet POWERLINK Standardization Group
ControlNet
CompoNet
PROFIBUS
Fieldbus Foundation
IEC 61850

Electro-technical & Communication Standards

- Source: Paul Didier (Cisco) TSNA’15 - Survey of IoT Consortia and Community.pdf

Related News:

GPS World: The Internet of Everything: It's All in the Timing (2015-06)

http://gpsworld.com/the-internet-of-everything-its-all-in-the-timing/

Arstechnica: The future is the Internet of Things—deal with it (2015-10)

http://arstechnica.com/unite/2015/10/the-future-is-the-internet-of-things-deal-with-it/

WHY

Far from a solution looking for a problem

TSN is a solution for many of today's problems

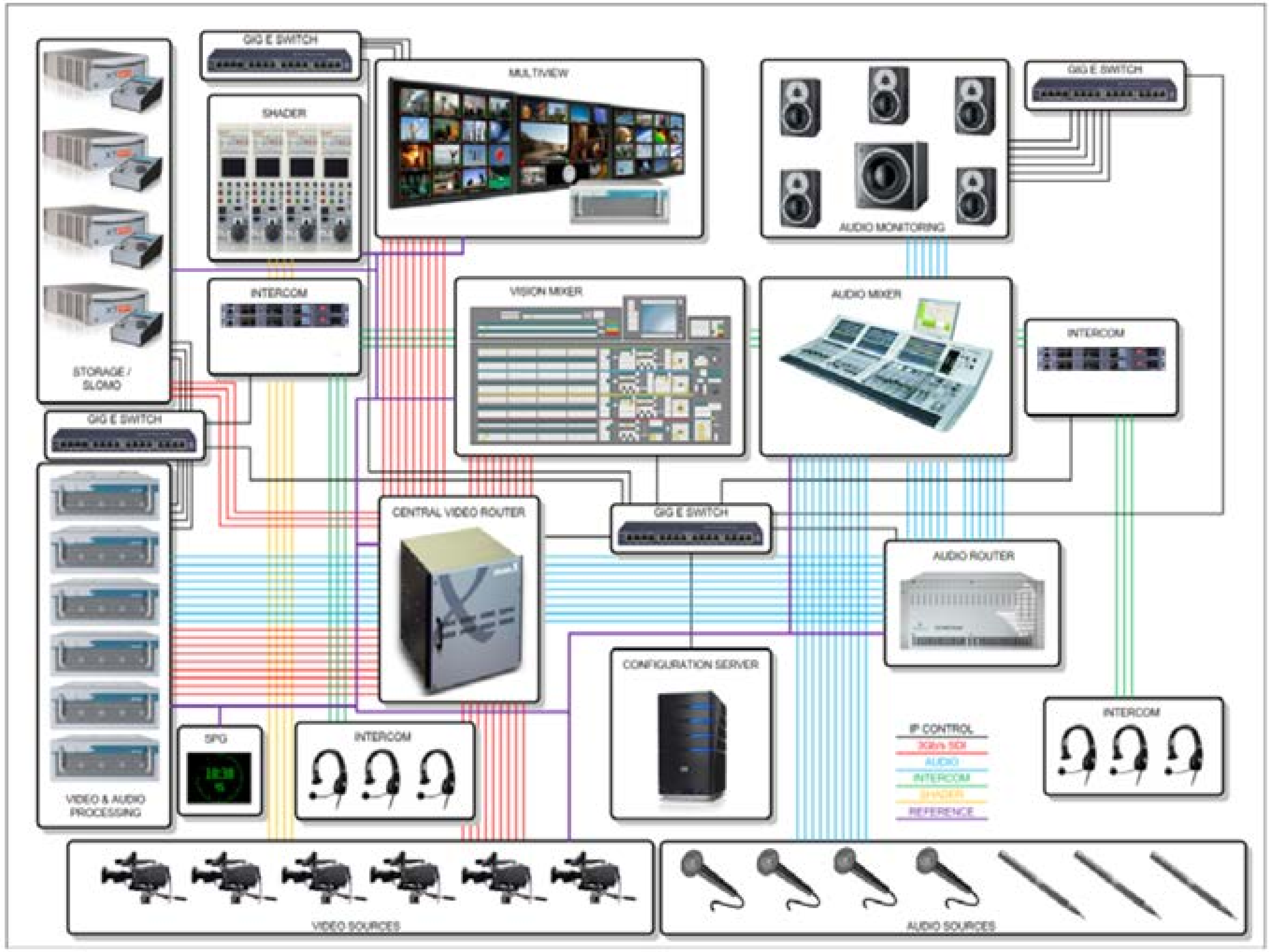
What does TSN solve?

Lets start with an example

- Consider a musician in a live audio performance
  - Musicians monitor their performance live in an ear-piece, >10ms delay on that audio monitor confuses performer {think echo on a conference call}
  - 10ms - Maximum delay between the musician "doing something" and hearing the same "something" in a monitoring ear-piece (speaker)
  - 8ms budgeted for the amount of delay of sound from mic pickup + DSP delays + mixer delays + monitor speaker to the musician (the analog mic/speaker takes time)
  - Leaves 2 ms for network from musician (microphone) through network to monitoring speaker
- Put another way, the above is a "Control Loop", where the feedback (audio in-ear monitor speaker) aids the system (musician's performance)
  - Loop is 10ms, of which network transport allocation is only 2ms
  - In most cases, a human is not "in the loop"

Another example – a land before TSN

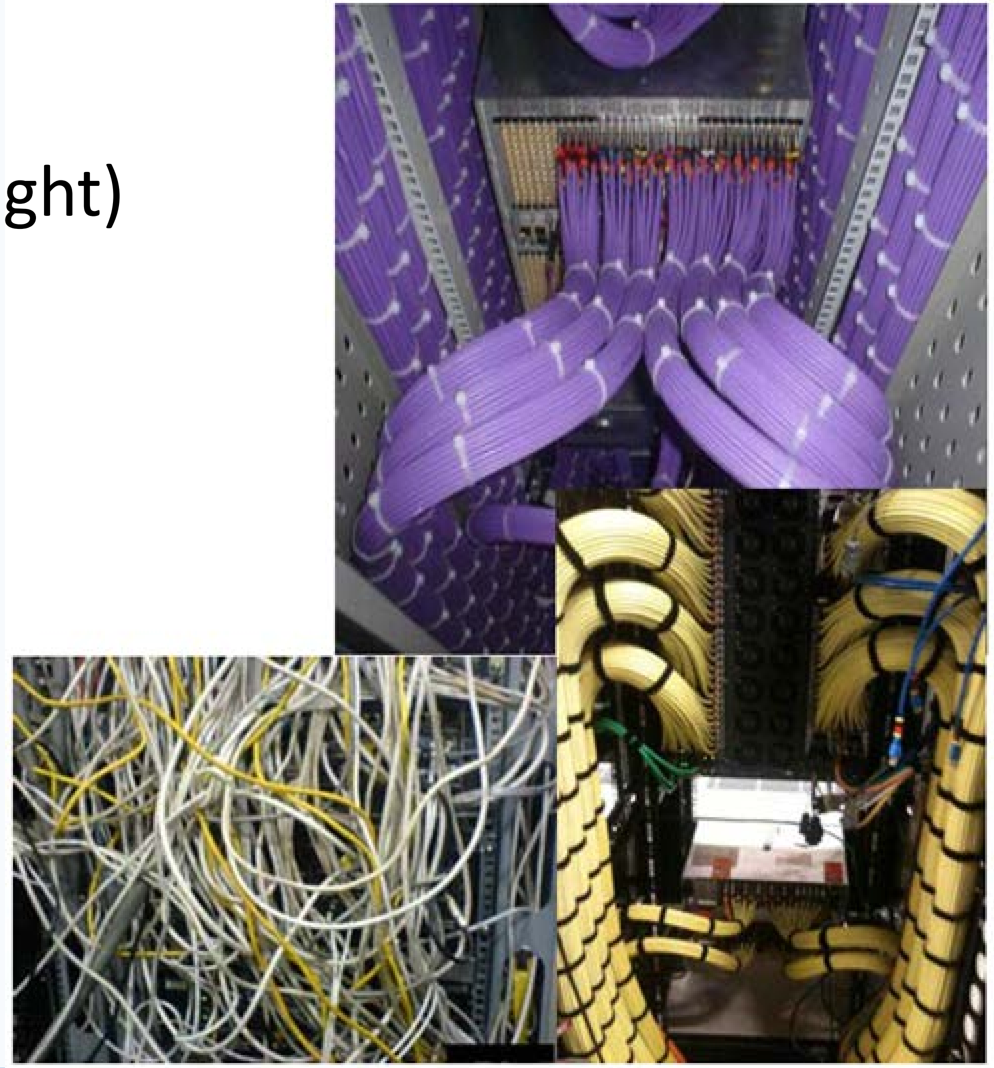
- A classic Audio/Video network – all video feeds are dedicated runs
  - non-networked – matrix switched, but not a packetized network



- Source: (Riedel) Professional Audio Video; Using Networks for Stadiums, Stage, and Studio.pdf

A land without TSN

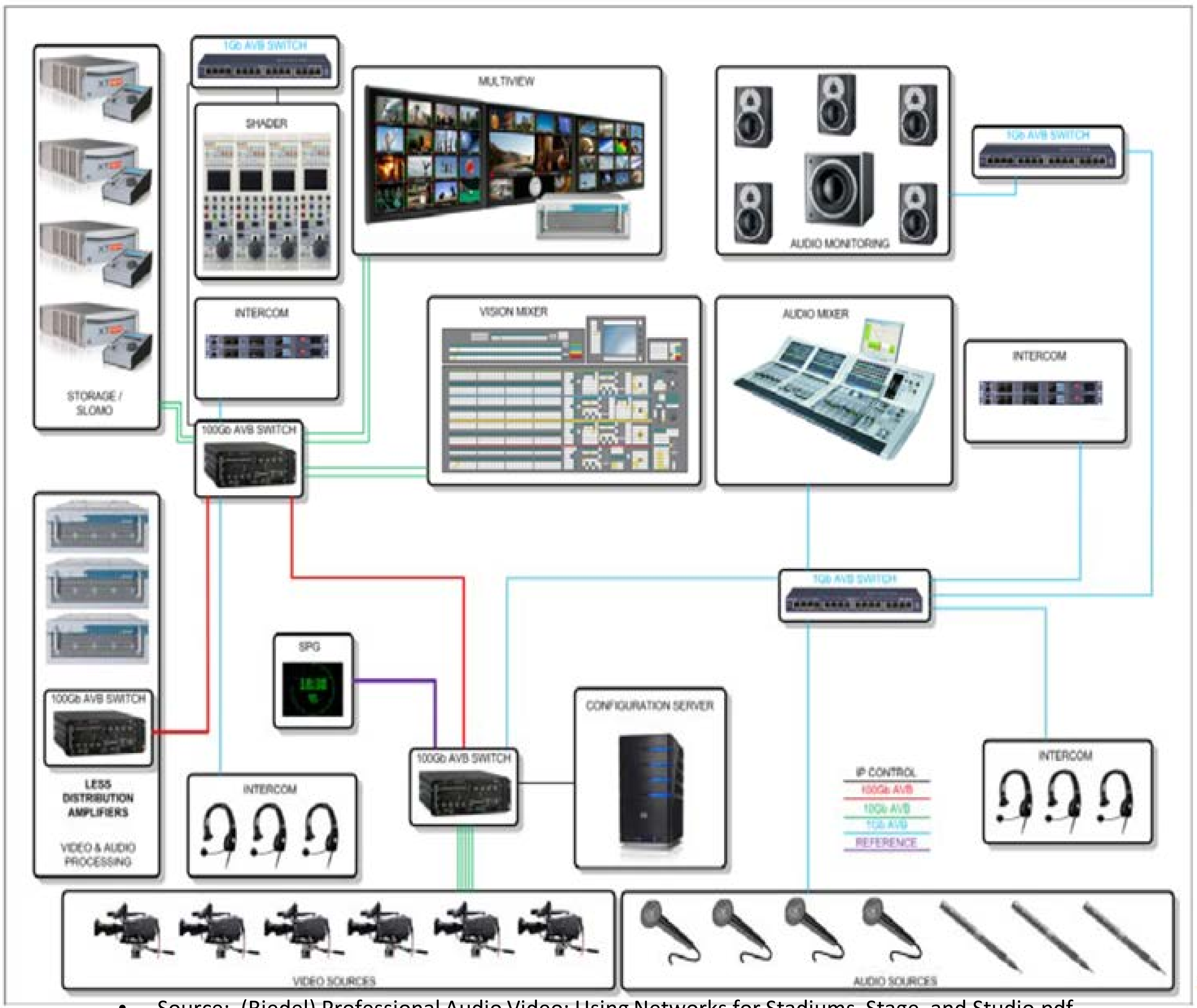
- Video matrix switch (purple cables @ right)
- Massive cabling
- Inflexible
- Dedicated cables
- One cable per signal
- Uni-directional



- Source: (Riedel) Professional Audio Video; Using Networks for Stadiums, Stage, and Studio.pdf

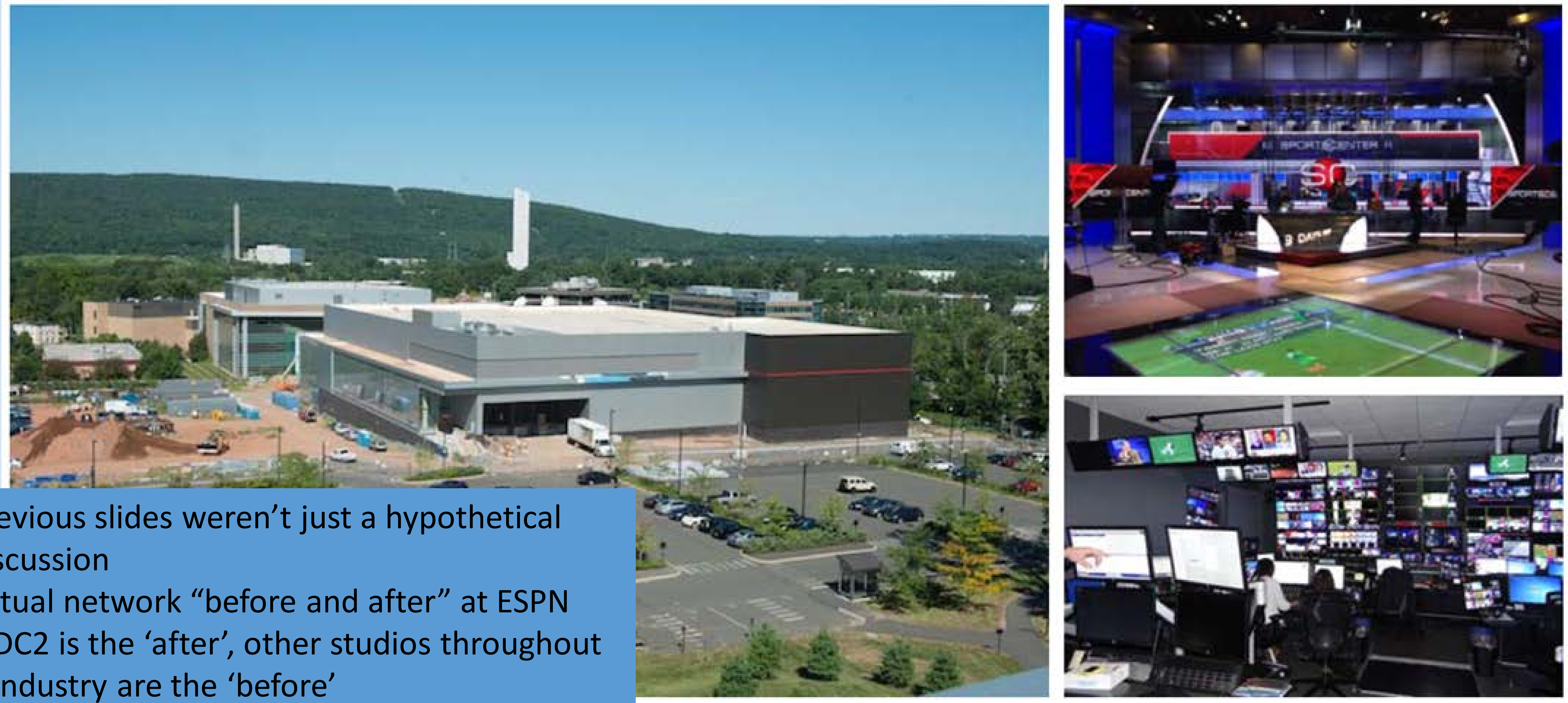
AVB/TSN System

- Distributed architecture
- Virtual cabling
- Distributed I/O
- Full-Duplex connections
- Allows legacy network traffic



- Source: (Riedel) Professional Audio Video; Using Networks for Stadiums, Stage, and Studio.pdf

Real world example - ESPN’s DC2



ESPN Digital Center 2 in Bristol, CT

Previous slides weren’t just a hypothetical discussion

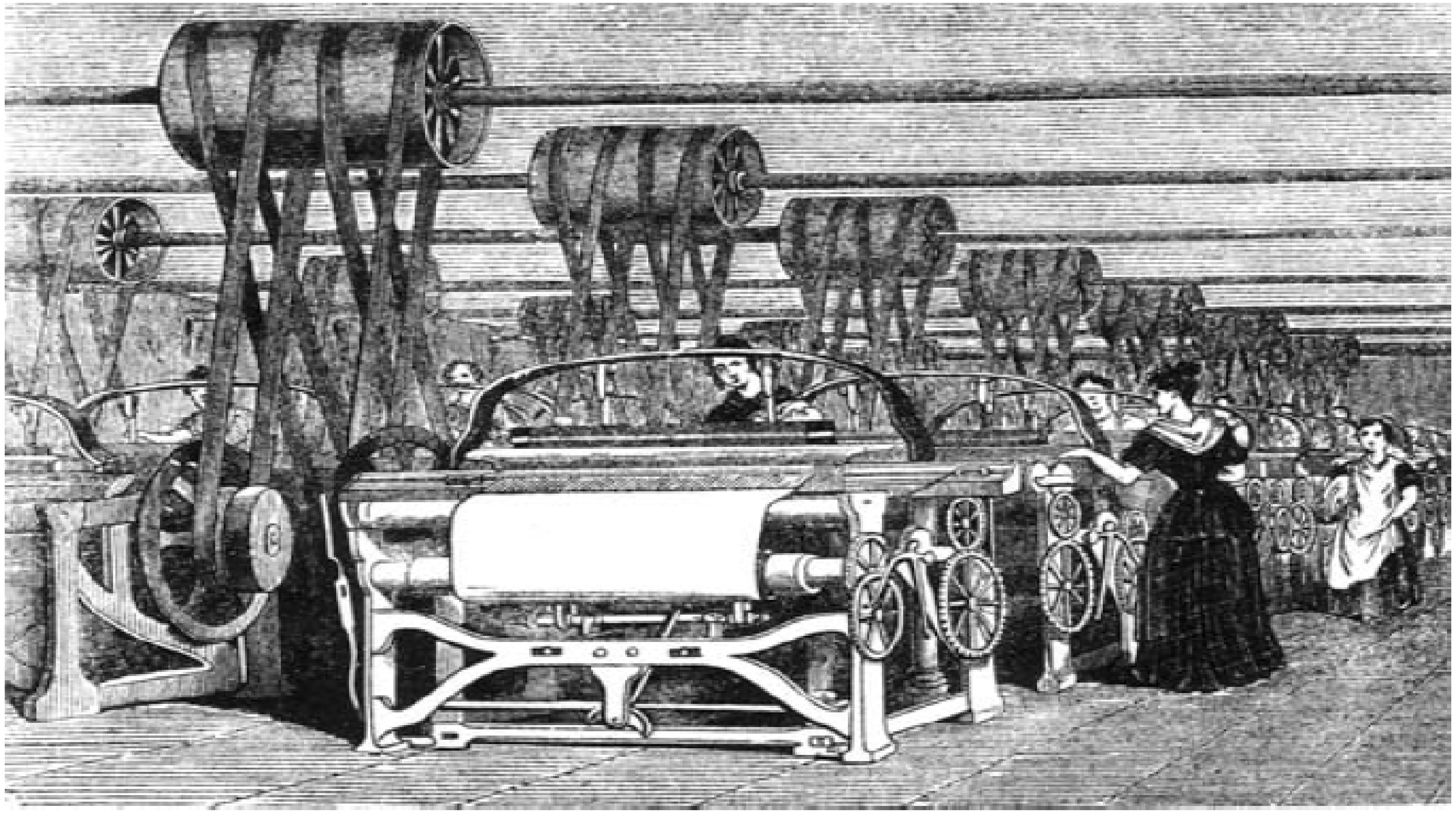
Actual network “before and after” at ESPN

- DC2 is the ‘after’, other studios throughout industry are the ‘before’

- Source: (Riedel) Professional Audio Video; Using Networks for Stadiums, Stage, and Studio.pdf

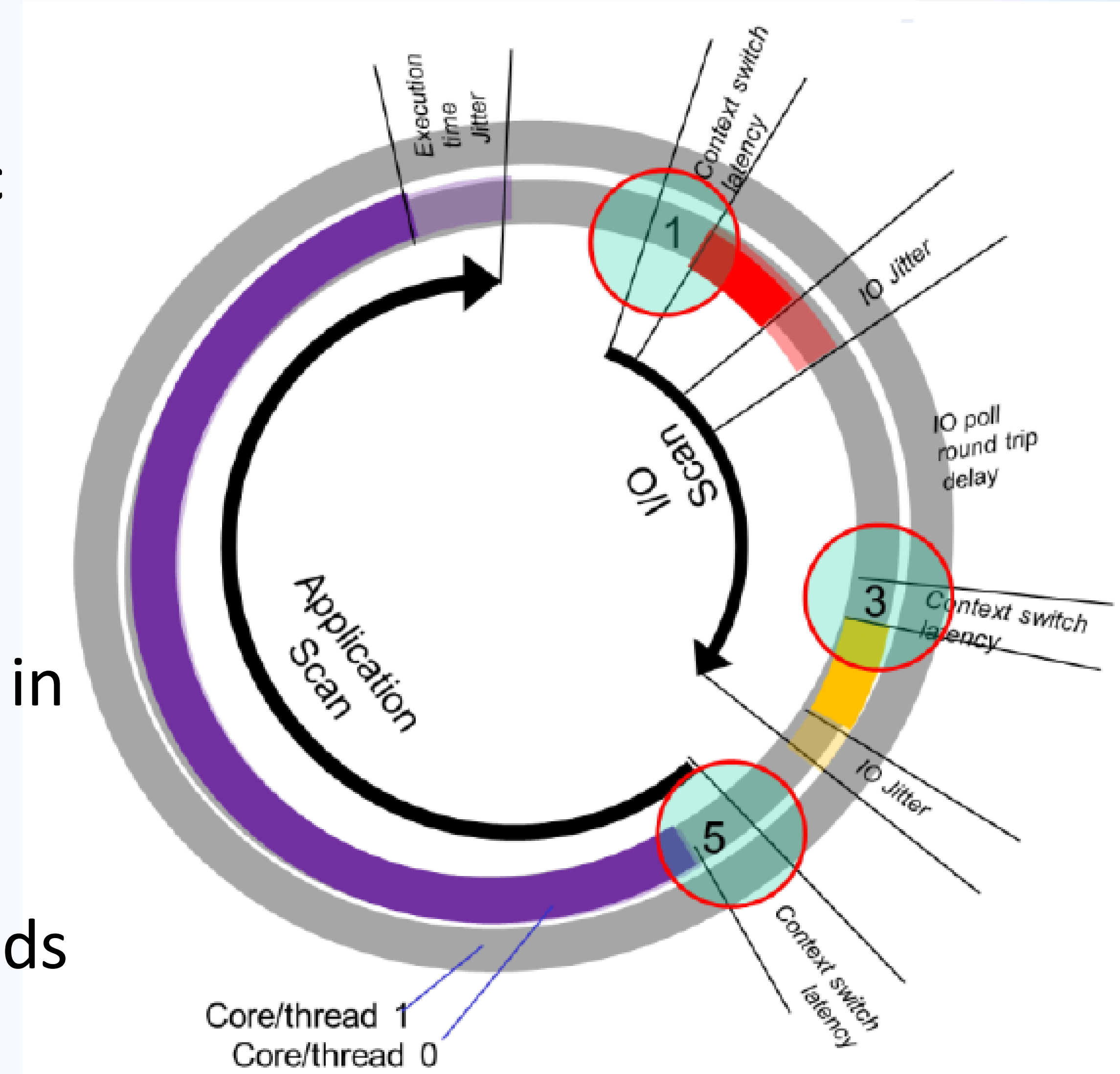
Another example – a land before TSN

- It wasn't long ago that machinery was 'timed' by belts, gears, chains, etc
  - Some still are - does your car have a timing belt?
- More recent machines have used motors and proprietary solutions where possible



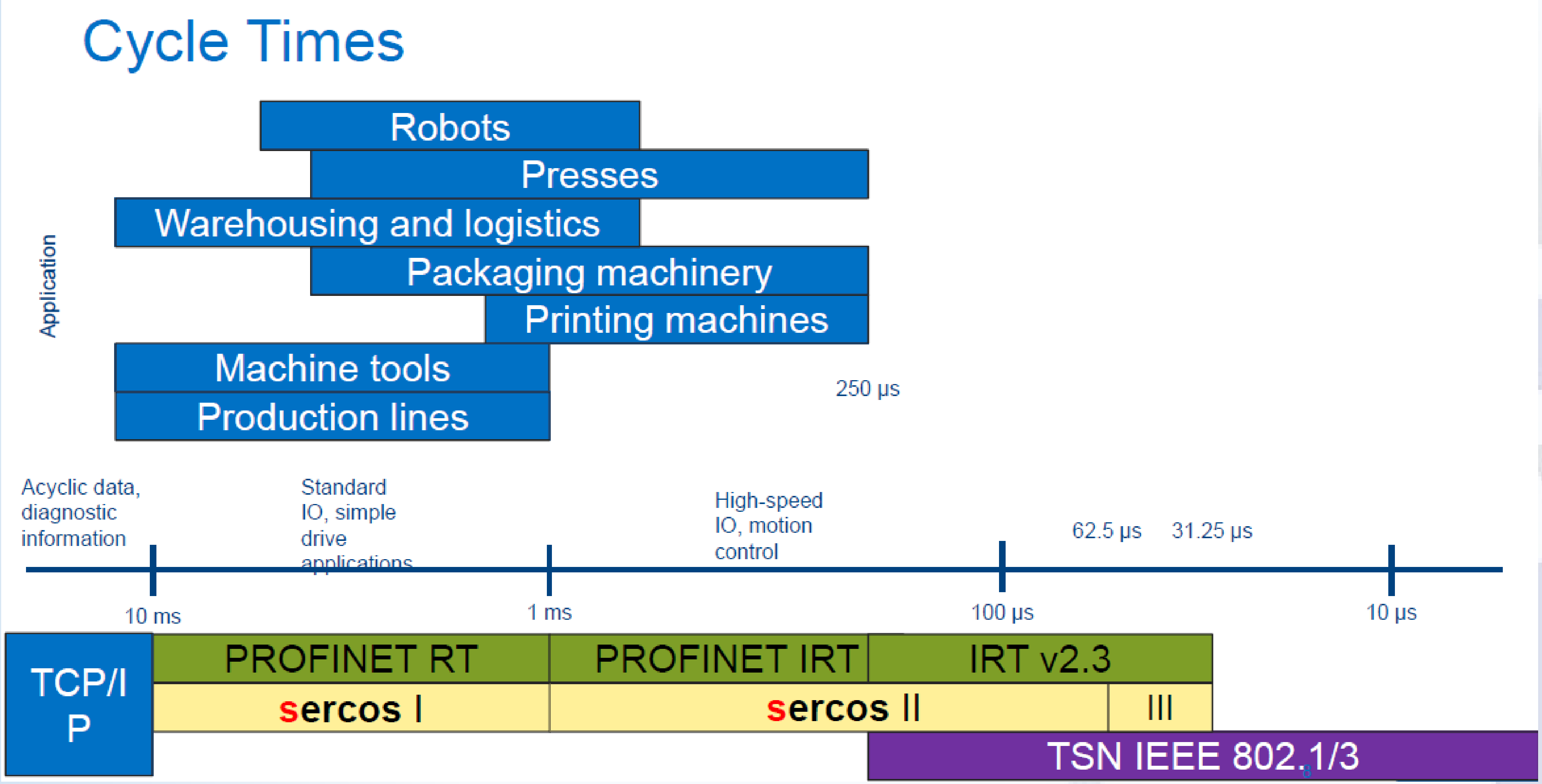
The Control Loop

- Move from mechanical to electronic synchronization has introduced an isochronous control loop
  - Sample the sensor data
  - Compute the action to take
  - Push out the new command(s)
- Low speed process have cycle times in 100s of milliseconds
- Today's high speed processes, this control loop runs at 250 microseconds



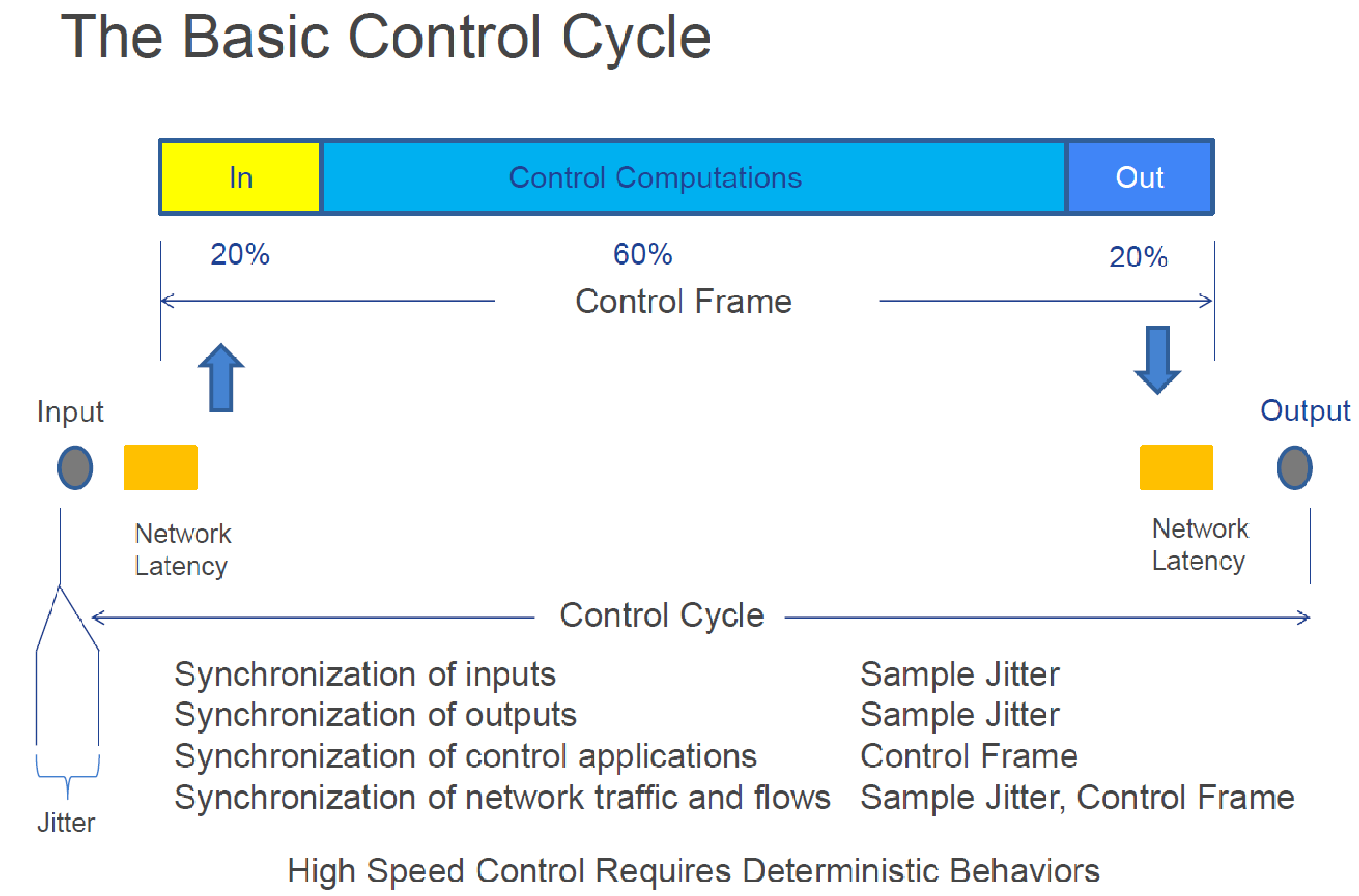
- Source: James Coleman (Intel) TSNA'15 - Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

Control Loop Speeds



- Source: James Coleman (Intel) TSNA'15 - Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

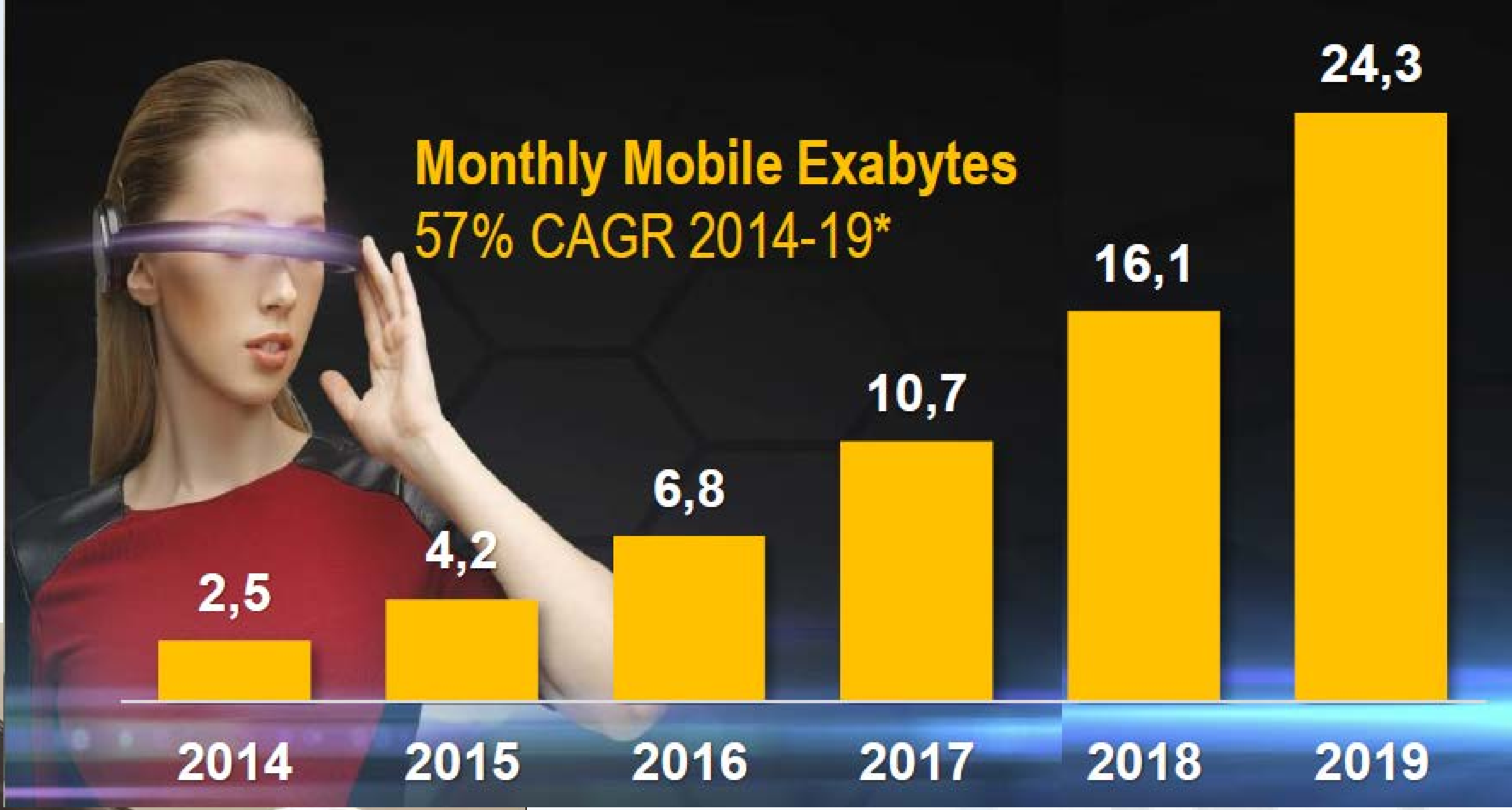
Another view of the Control Loop



- Source: Dan Sexton (GE) TSNA'15 - Industrial; Converging Control, Monitoring, and Enterprise Networks to Support Flexible Manufacturing.pdf

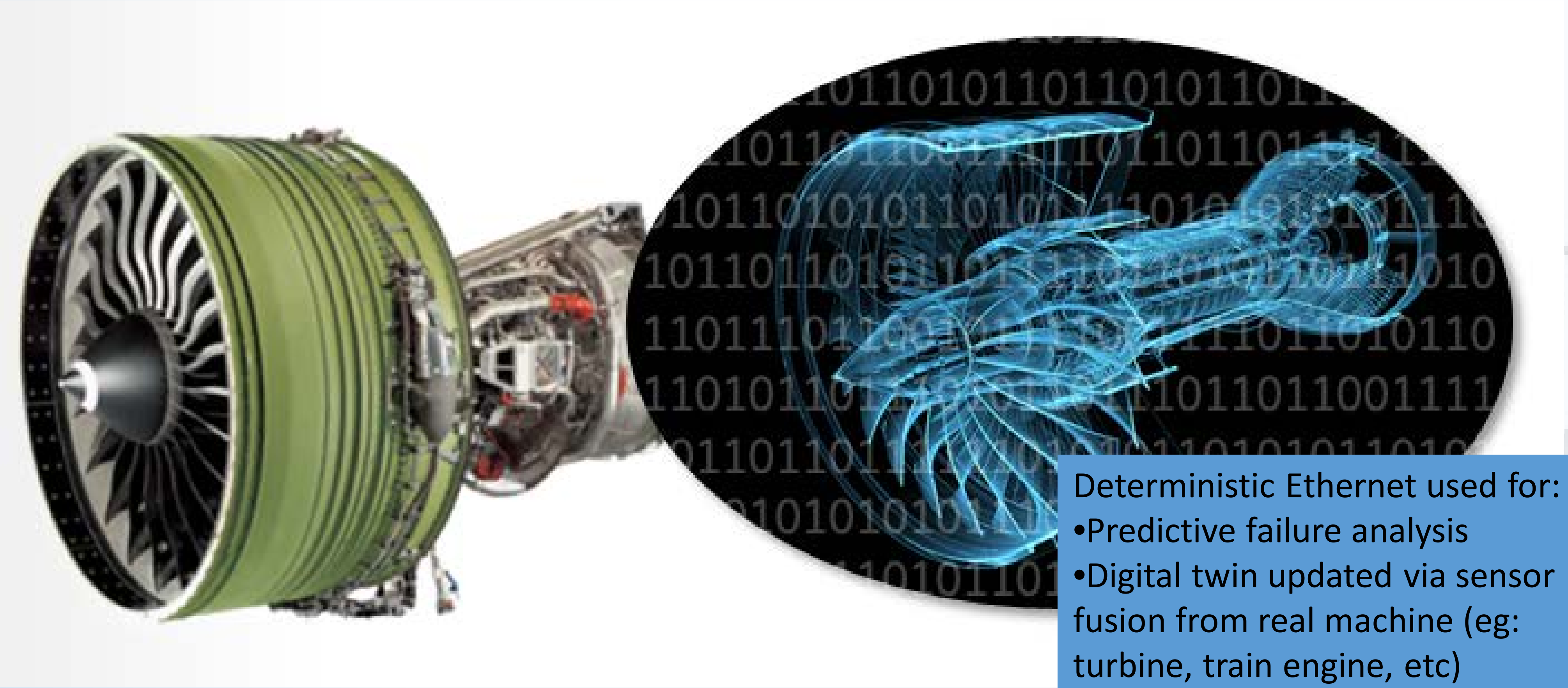
IoT Challenges

Great “gobs” of compute need, data/sensor movement



- ADAS – Advanced Driver Assist System
- CAGR – Compound Annual Growth Rate
- Exabytes – 1 billion billion bytes (1 Giga gigabytes) – 10^{18} bytes

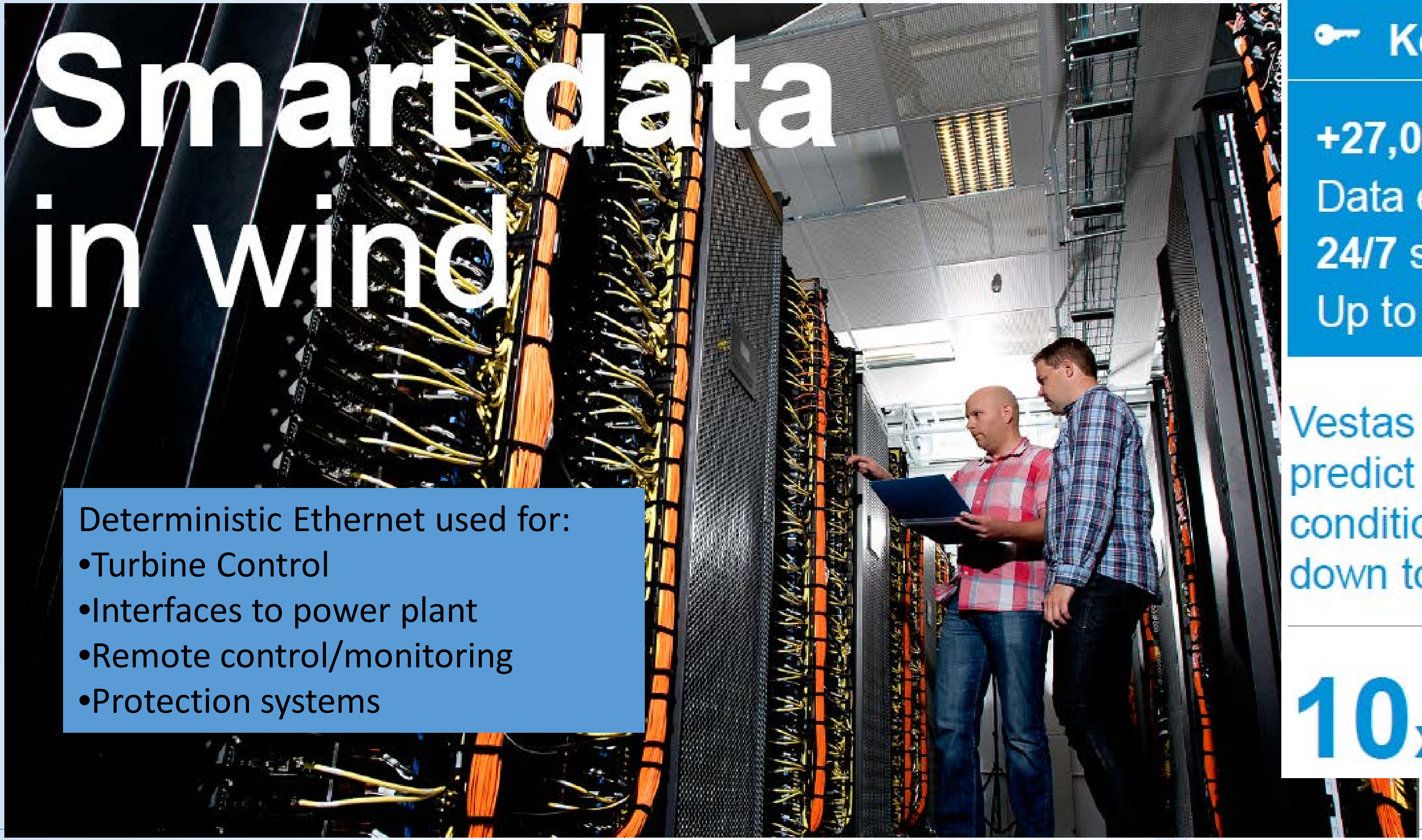
Digital Twins



Deterministic Ethernet used for:
•Predictive failure analysis
•Digital twin updated via sensor fusion from real machine (eg: turbine, train engine, etc)

- http://www.geglobalresearch.com/impact/physical-digital-the-new-power-couple
- http://www.geglobalresearch.com/impact/how-a-digital-twin-for-physical-assets-can-help-achieve-no-unplanned-downtime

Vestas – Wind Turbines



Deterministic Ethernet used for:
- Turbine Control
- Interfaces to power plant
- Remote control/monitoring
- Protection systems

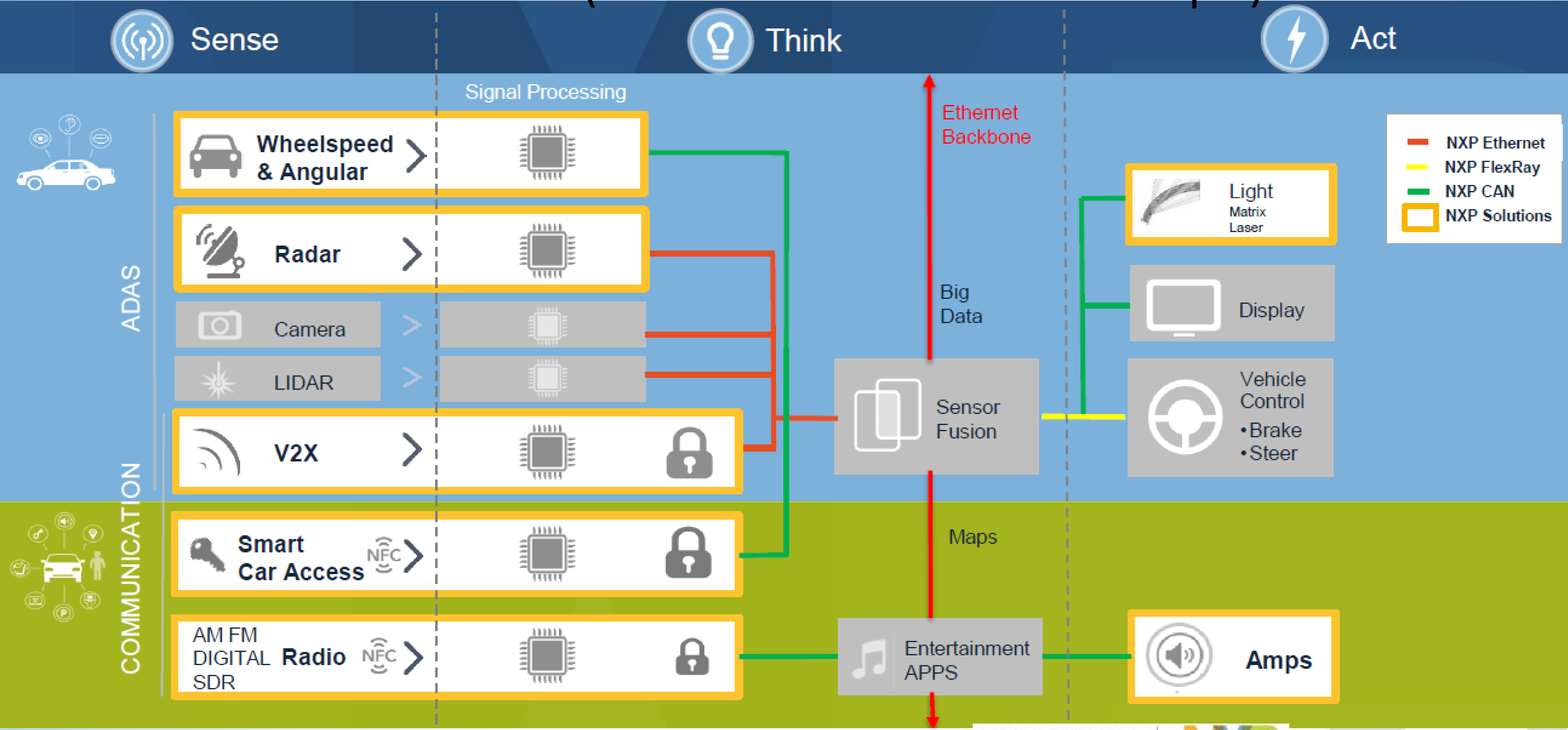
Key takes

+27,000 turbines
Data every **10th minute**
24/7 surveillance
Up to **500** data points

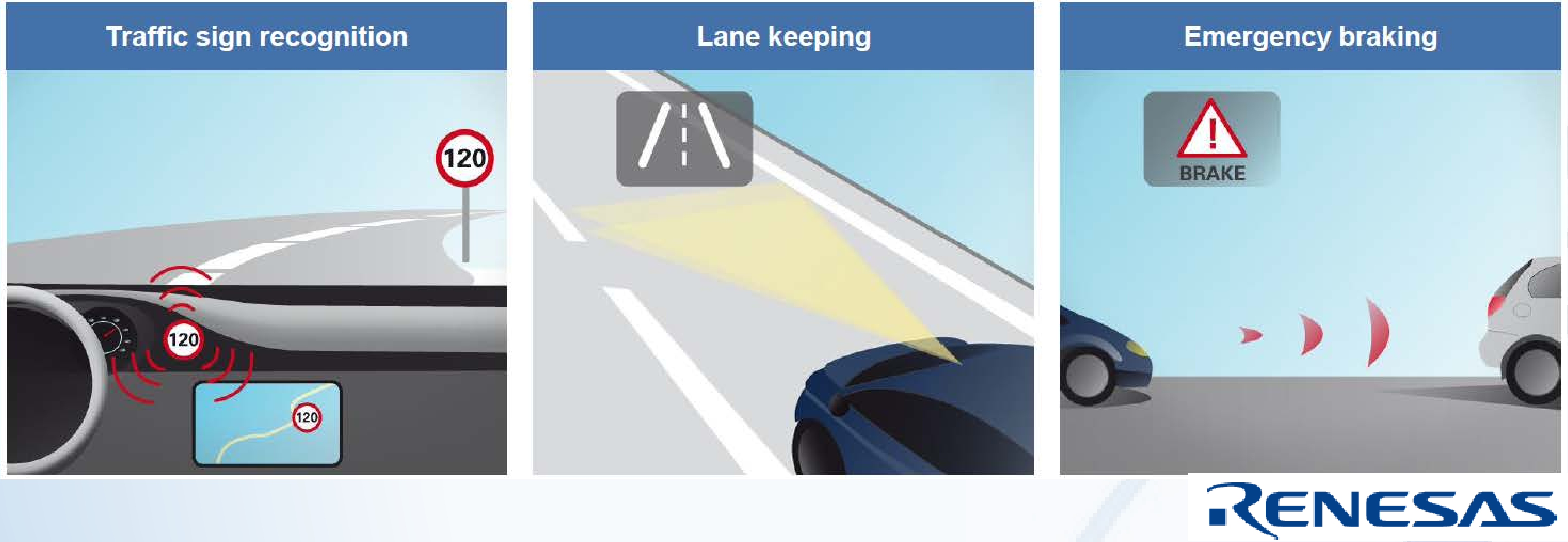
Vestas has the ability to predict meteorological conditions world-wide down to areas of :

10x10m

Connected Car (Another Control Loop!)



Connected Car (2)

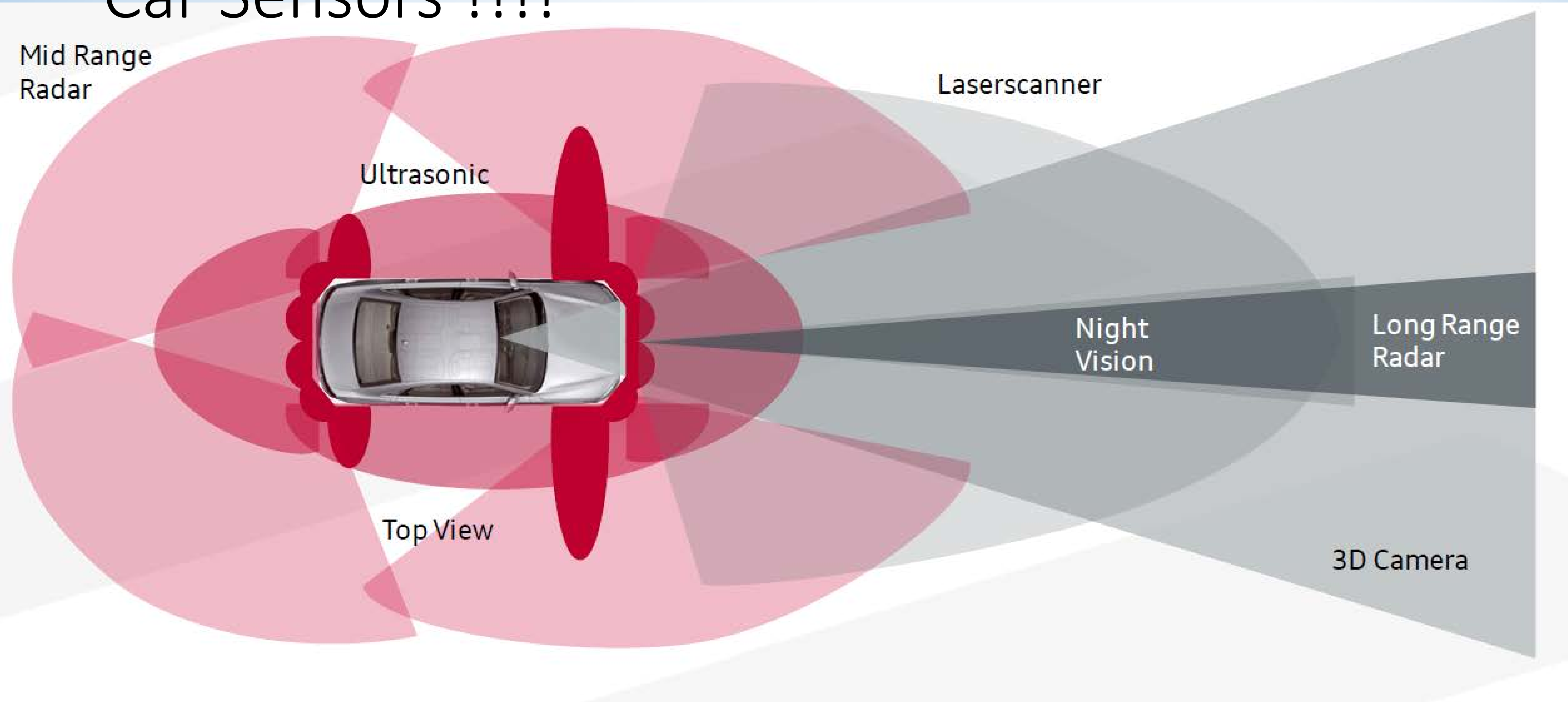


ADAS to Piloted

Assistance systems			Piloted (automated) systems		
Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
Driver only	**Assisted**	**Partial automation**	**Conditional automation**	**High automation**	**Full automation (driverless)**
Driver **permanently** in charge of **longitudinal and lateral control**	Driver permanently in charge of either longitudinal or lateral control Driver is ready to **take over immediately**	Driver permanently monitors Driver is ready to **take over immediately**	Driver does **not need to monitor** the dynamic driving task nor the driving environment at all times but **be attentive to take over.**	Driver is not required during **defined use case.**	System performs the lateral and longitudinal dynamic driving task in all situations encountered during the entire journey. No driver required. Vehicle does not have steering wheel or pedals.
	Vehicle takes charge of the other function	Vehicle takes charge of **longitudinal and lateral control** for **a certain time and in certain situations**	Vehicle takes charge of longitudinal and lateral control for **a certain time and in certain situations** Need to take over is announced with sufficient **advance warning, ancillary activities offered by vehicle** can be performed	Vehicle takes **full charge** of longitudinal and lateral control **in defined use case** The system is capable of establishing a **risk-minimized state** in all situations, all **ancillary activities** possible	
Today's driver assistance systems		**Next generation**	**New area of activity Piloted driving and parking**		

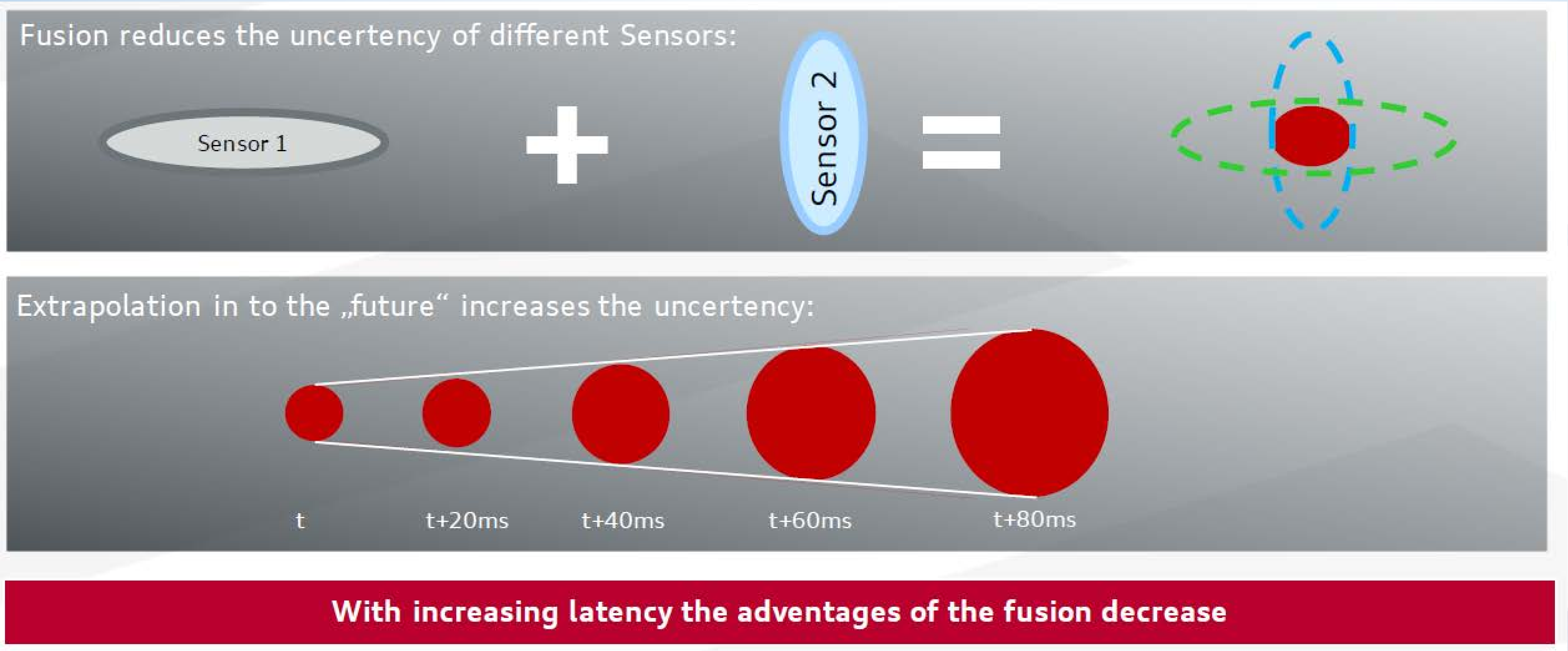
Source: SAE

Car Sensors !!!!



- Source: 07_VukotichRudolph_Audi.pdf – DE-Forum 2015

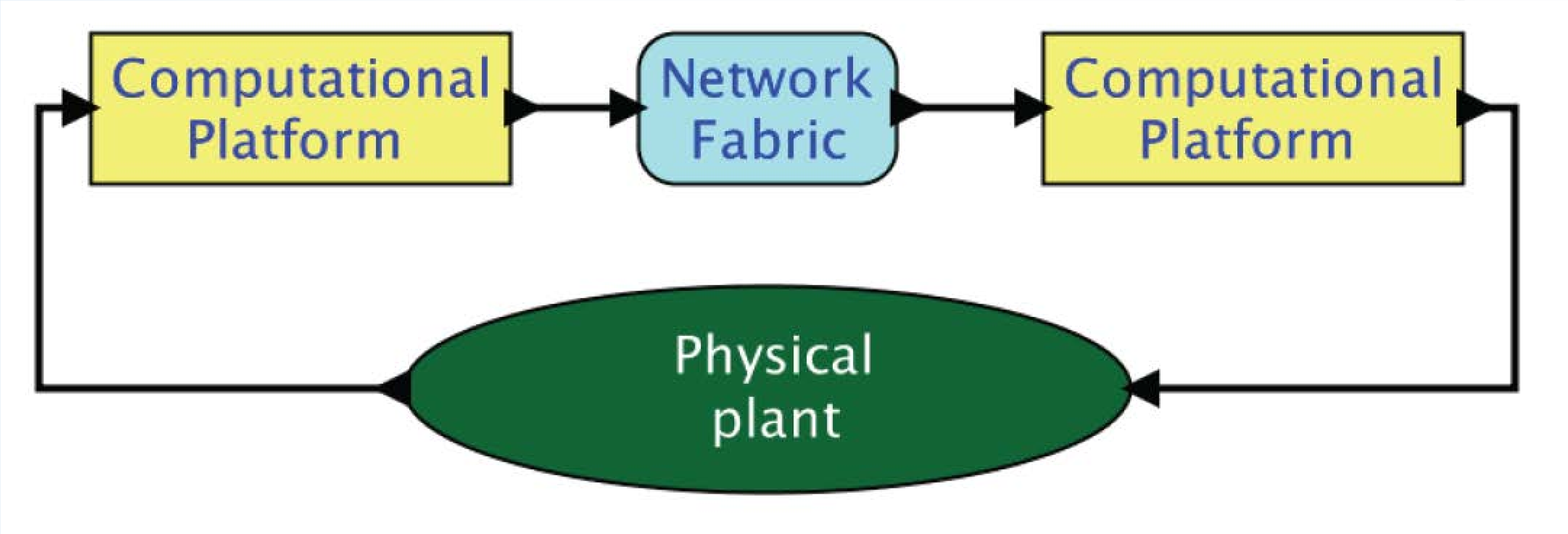
Sensor Fusion !



- Source: 07_VukotichRudolph_Audi.pdf – DE-Forum 2015

Audi
Vorsprung durch Technik

Another Control Loop – a simple CPS

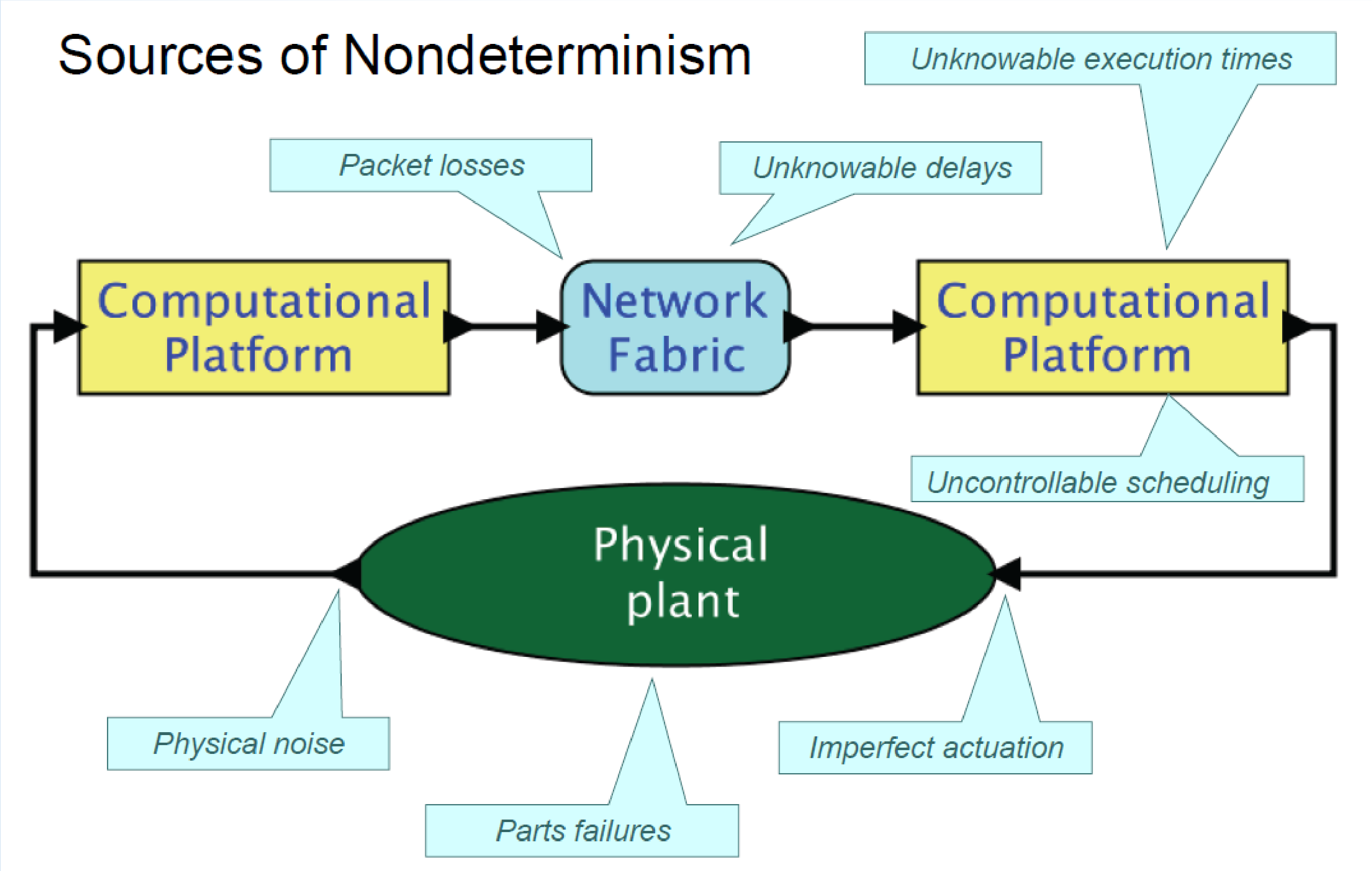


- Source: Dr Edward Lee (UC Berkley) TSNA'15 – The Internet of Important Things.pdf

Determinacy & Deterministic Systems

- It doesn’t mean super fast (throughput)
- It doesn’t mean super fast (latency)
- It means being definitely and unequivocally characterized
- A **deterministic system** is a system in which no randomness is involved in the development of future states of the system.[1] A deterministic model will thus always produce the same output from a given starting condition or initial state (Wikipedia – fount of all knowledge)

The Control Loop must be Deterministic



- Source: Dr Edward Lee (UC Berkley) TSNA'15 – The Internet of Important Things.pdf

Models vs Reality

- In the face of such nondeterminism, does it make sense to talk about deterministic models for cyber-physical systems? (Ibid: Dr Edward Lee)

- “A deterministic model will thus always produce the same output from a given starting condition or initial state”

Models vs Reality (2)

You will never strike oil by drilling through the map!

BUT

- this does not, in any way, diminish the **value of a map** !

- Source: Dr Edward Lee (UC Berkley) TSNA’15 – The Internet of Important Things.pdf

The Kopetz Principal

Prof. Dr. Hermann Kopetz

- Many (predictive) properties that we assert about systems (determinism, timeliness, reliability, safety) are in fact not properties of an *implemented system, but rather properties* of a *model of the system.*

- We can make definitive statements about *models, from which we can infer properties of* system realizations. The validity of this inference depends on *model fidelity, which is* always approximate.

- Source: Dr Edward Lee (UC Berkley) TSNA'15 – The Internet of Important Things.pdf

Deterministic Models of Nondeterministic Systems

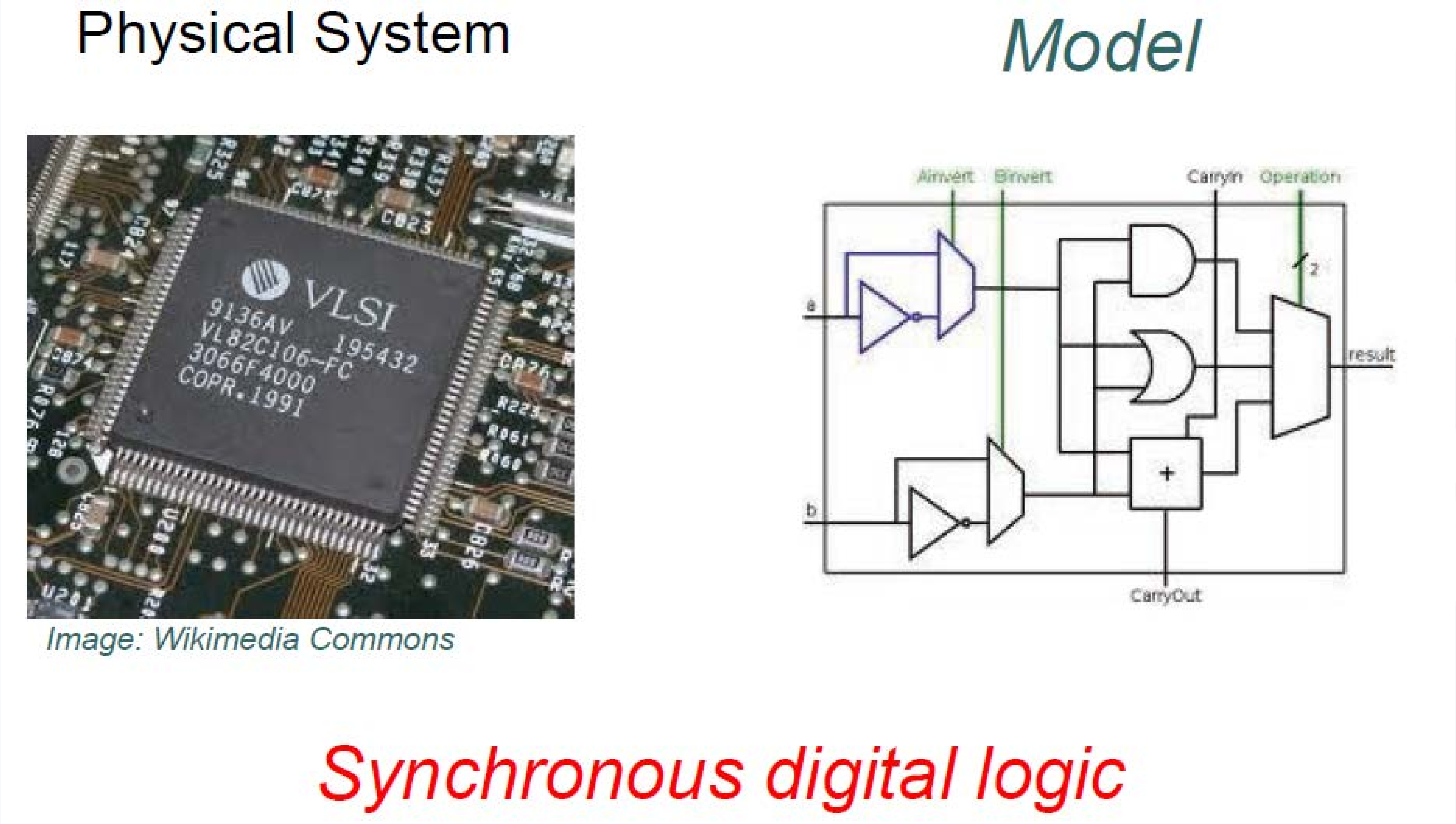


Image: Wikimedia Commons

- Source: Dr Edward Lee (UC Berkley) TSNA'15 – The Internet of Important Things.pdf

Deterministic Models of Nondeterministic Systems

Physical System

Image: Wikimedia Commons

Model

```java
/** Reset the output receivers, which are the inside receivers of
 *  the output ports of the container.
 *  @exception IllegalActionException If getting the receivers fails.
 */
private void _resetOutputReceivers() throws IllegalActionException {
    List<IOPort> outputs = ((Actor) getContainer()).outputPortList();
    for (IOPort output : outputs) {
        if (_debugging) {
            _debug("Resetting inside receivers of output port: "
                    + output.getName());
        }
        Receiver[][] receivers = output.getInsideReceivers();
        if (receivers != null) {
            for (int i = 0; i < receivers.length; i++) {
                if (receivers[i] != null) {
                    for (int j = 0; j < receivers[i].length; j++) {
                        if (receivers[i][j] instanceof FSMReceiver) {
                            receivers[i][j].reset();
                        }
                    }
                }
            }
        }
    }
}
```

Single-threaded imperative programs

- Source: Dr Edward Lee (UC Berkley) TSNA'15 – The Internet of Important Things.pdf

Deterministic Models of Nondeterministic Systems

Physical System

Image: Wikimedia Commons

Model

$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(0) + \frac{1}{M}\int_{0}^{t} \mathbf{F}(\tau)d\tau$$

Differential Equations

- Source: Dr Edward Lee (UC Berkley) TSNA’15 – The Internet of Important Things.pdf

Innovation Needed !!!

- A Major Problem for CPS: Combination of Deterministic Models are Nondeterministic
- A Key Challenge: Timing is not Part of Software Semantics
- **Correct** execution of a program in C, C#, Java, etc has nothing to do with how long it takes to do anything. Nearly all our computation and networking abstractions are built on this premise
- Programmers have to step outside the programming abstractions to specify timing behavior
- Programmers have no map!
- Today, for computers, timing is merely a **performance metric** – it needs to be a **correctness criterion**

- Source: Dr Edward Lee (UC Berkley) TSNA'15 – The Internet of Important Things.pdf

Predictions from Dr. Lee

Today

- Timing behavior in computers emerges from the physical realization

Tomorrow

- Timing behavior will be part of the programming abstractions and their hardware realizations

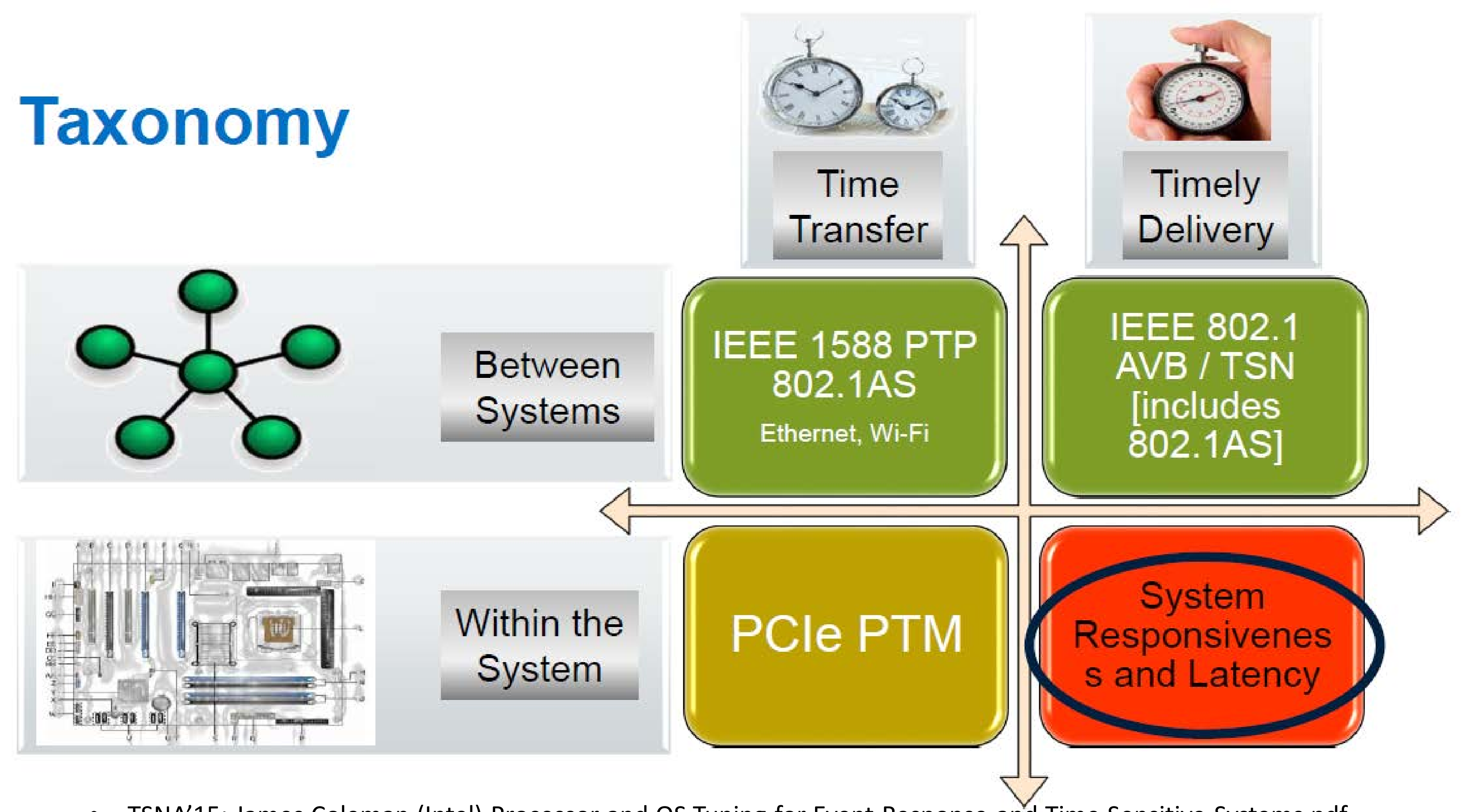
- Source: Dr Edward Lee (UC Berkley) TSNA’15 – The Internet of Important Things.pdf

HOW

How does TSN fit into the problem space

TSNs solve a part of the problem

- But not all
- Dr Lee raises the programmatic issues
  - Lets delve into the problem in the system
  - PCIe PTM – Precision Time Measurement



- TSNA'15: James Coleman (Intel) Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

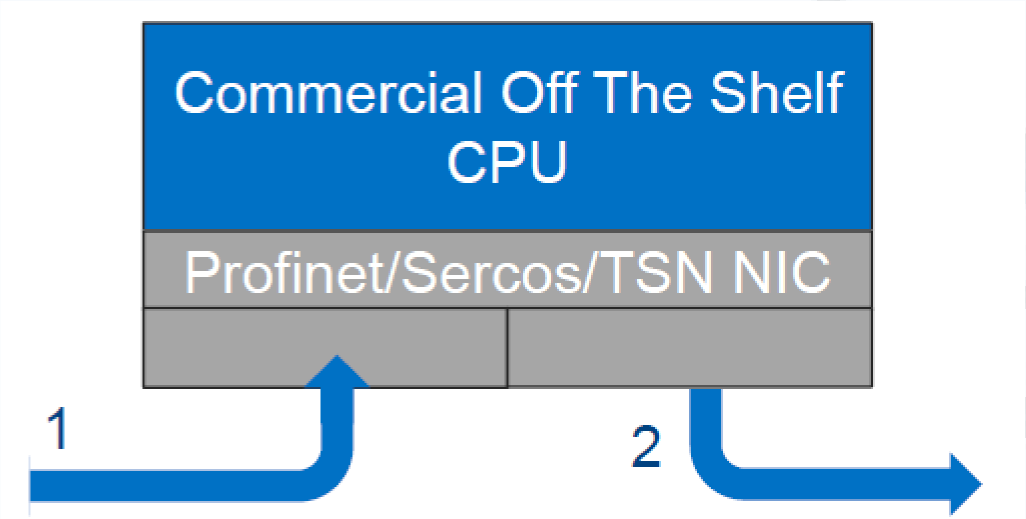
Minimum Cycle Time – Hardware Limits

The minimum cycle time is measured at the network from:

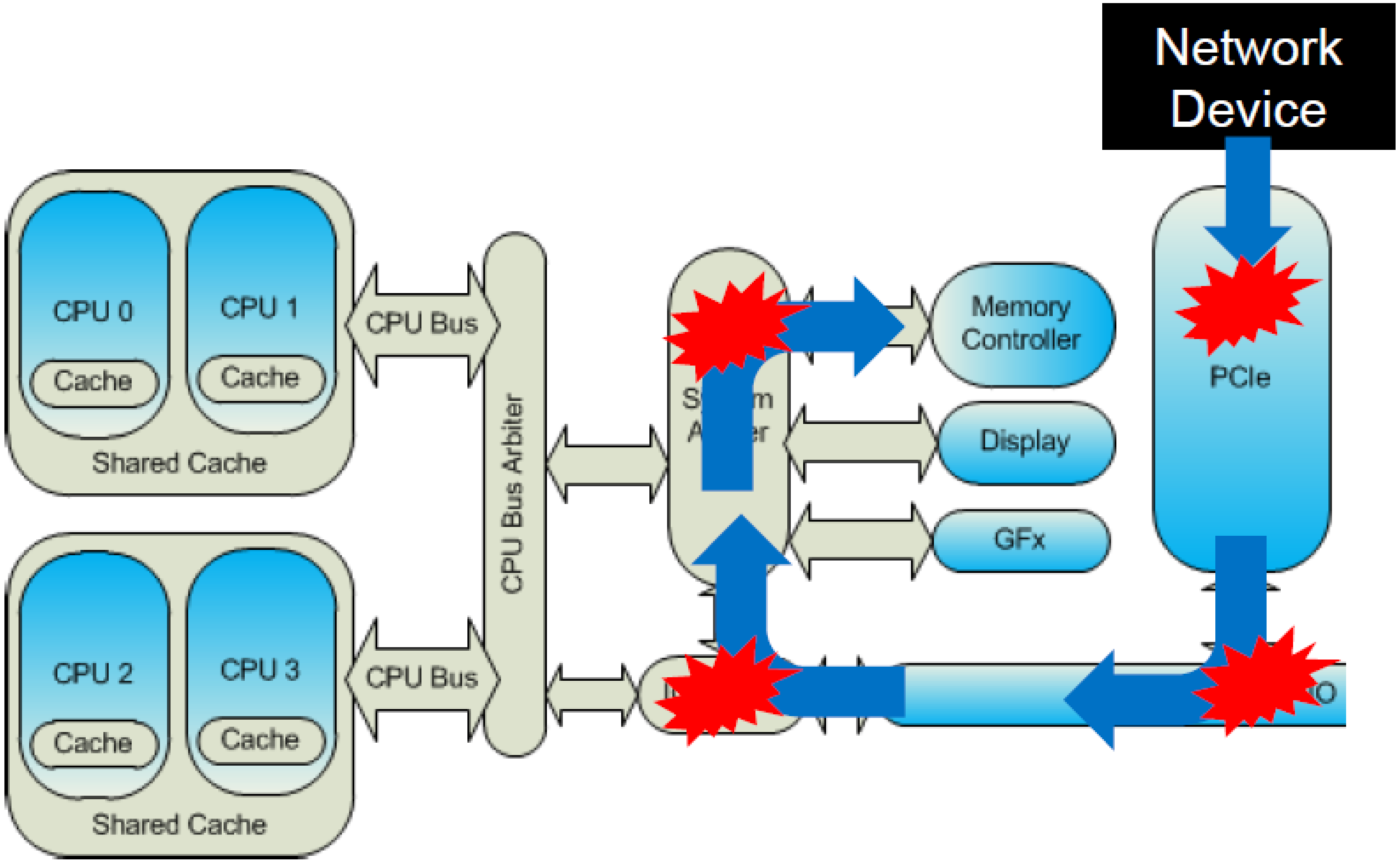
1) Sensor Data packet on the wire

To

2) Command Data packet on the wire

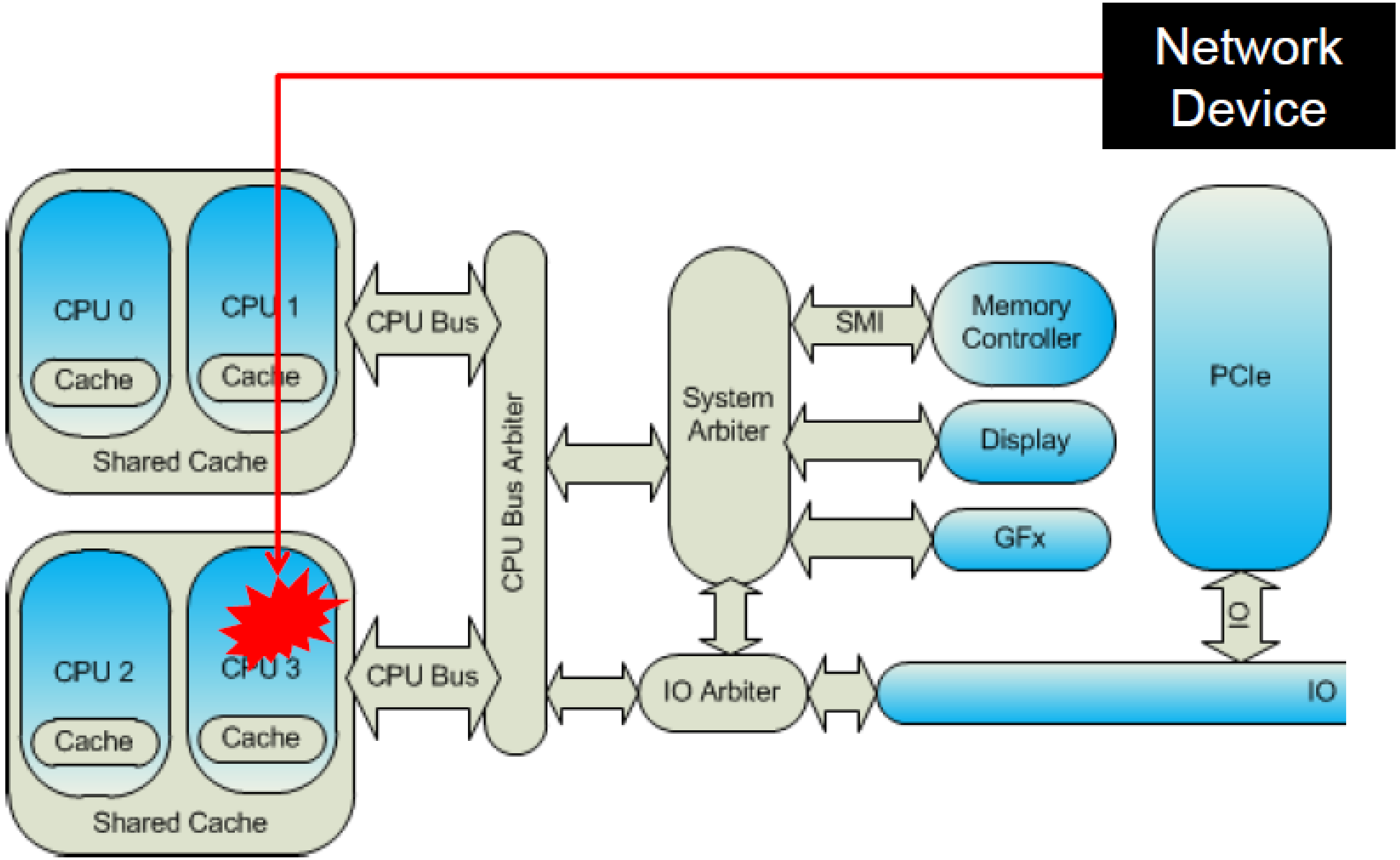


IO Device pushes data into memory



1) **IO device pushes data into system memory**
2) IO device notifies the CPU that data is ready to be processed
3) CPU context switches to handle data
4) CPU access data from memory
5) CPU processes data
6) CPU writes results to memory
7) CPU notifies the IO device that data is ready
8) IO device pulls the data from memory

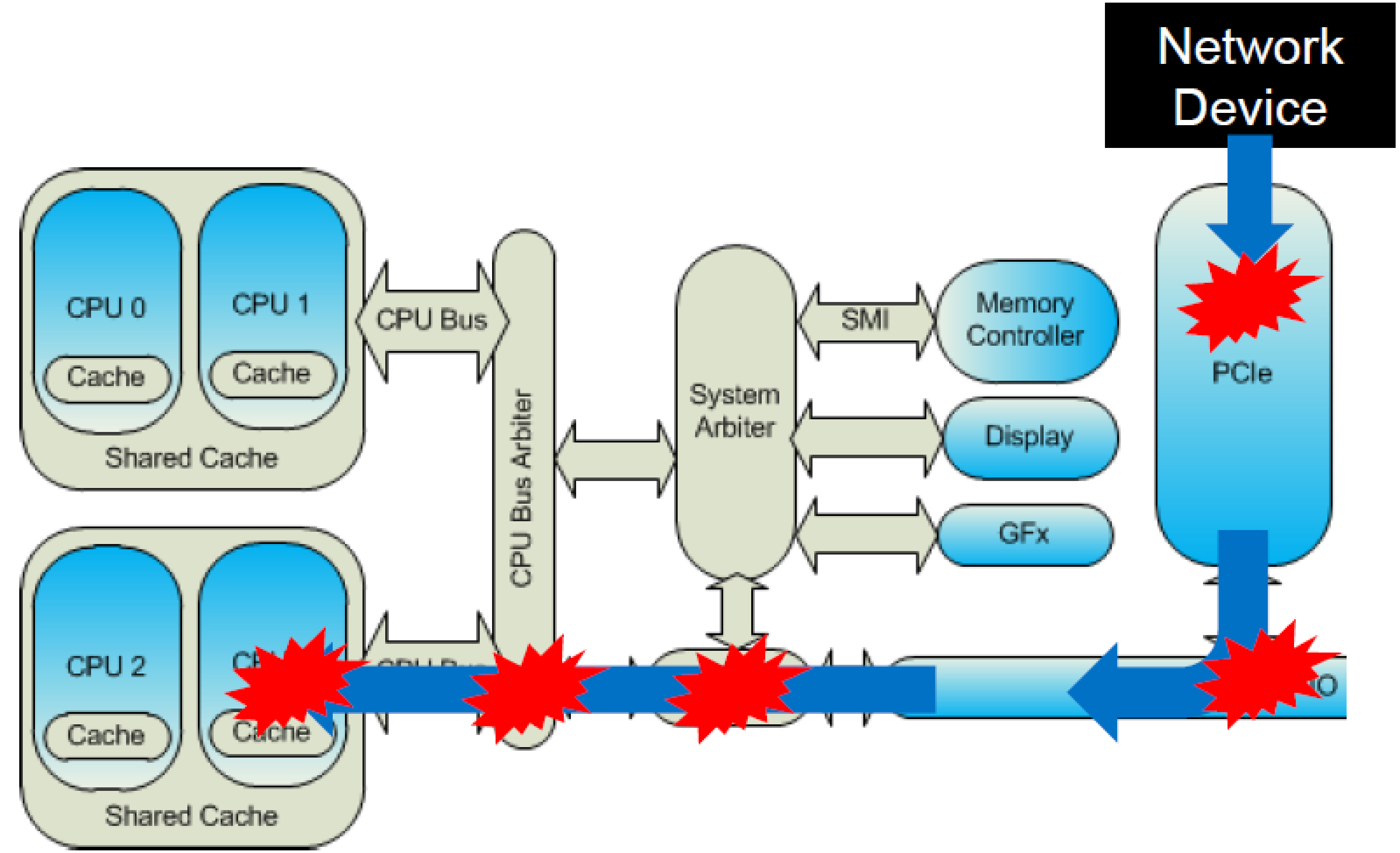
IO Device notifies CPU data is ready



1) IO device pushes data into system memory
2) **IO device notifies the CPU that data is ready to be processed**
3) CPU context switches to handle data
4) CPU access data from memory
5) CPU processes data
6) CPU writes results to memory
7) CPU notifies the IO device that data is ready
8) IO device pulls the data from memory

- TSNA'15: James Coleman (Intel) Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

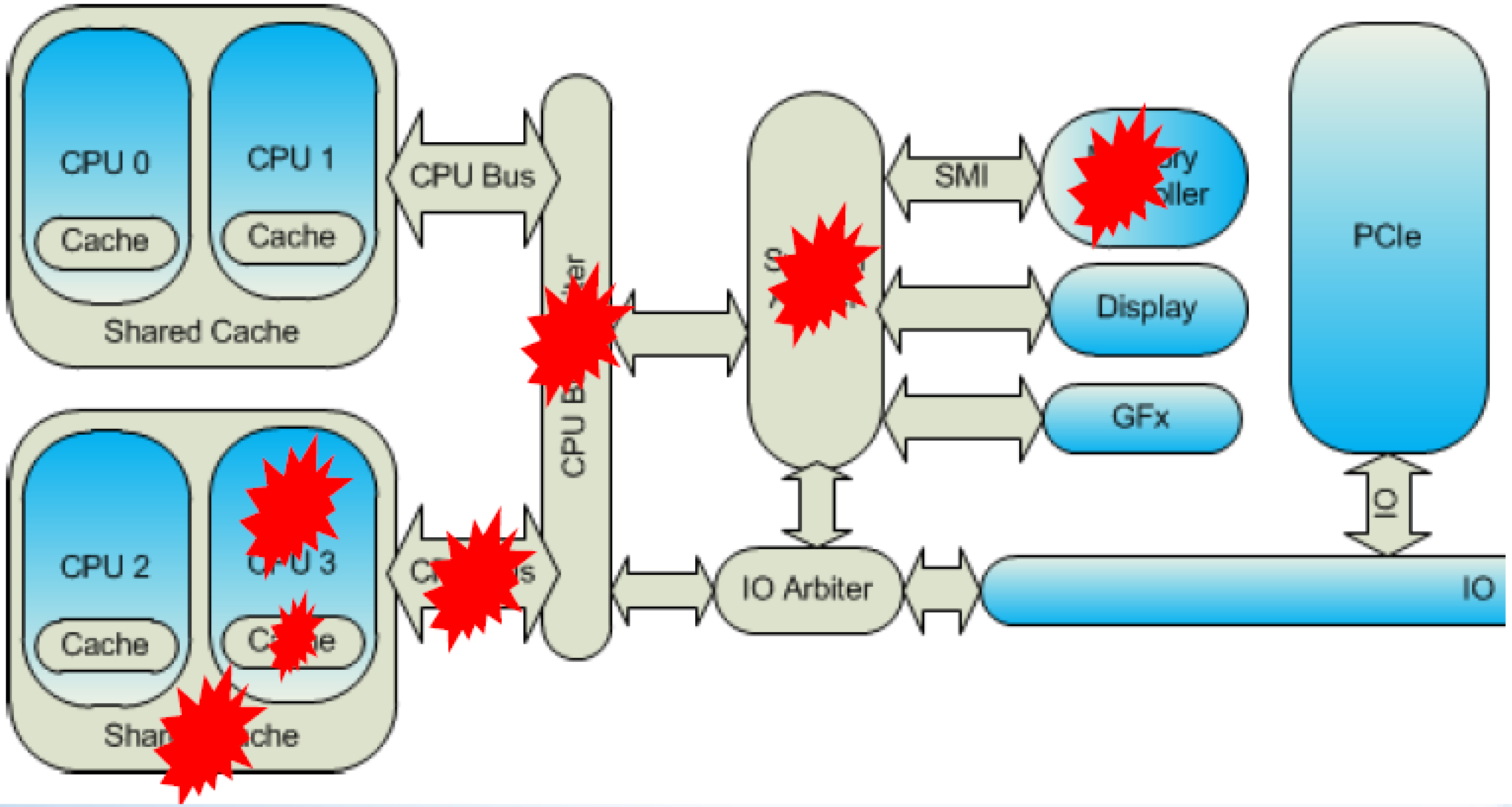
IO Device notifies CPU data is ready



1) IO device pushes data into system memory
2) **IO device notifies the CPU that data is ready to be processed**
3) CPU context switches to handle data
4) CPU access data from memory
5) CPU processes data
6) CPU writes results to memory
7) CPU notifies the IO device that data is ready
8) IO device pulls the data from memory

- TSNA'15: James Coleman (Intel) Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

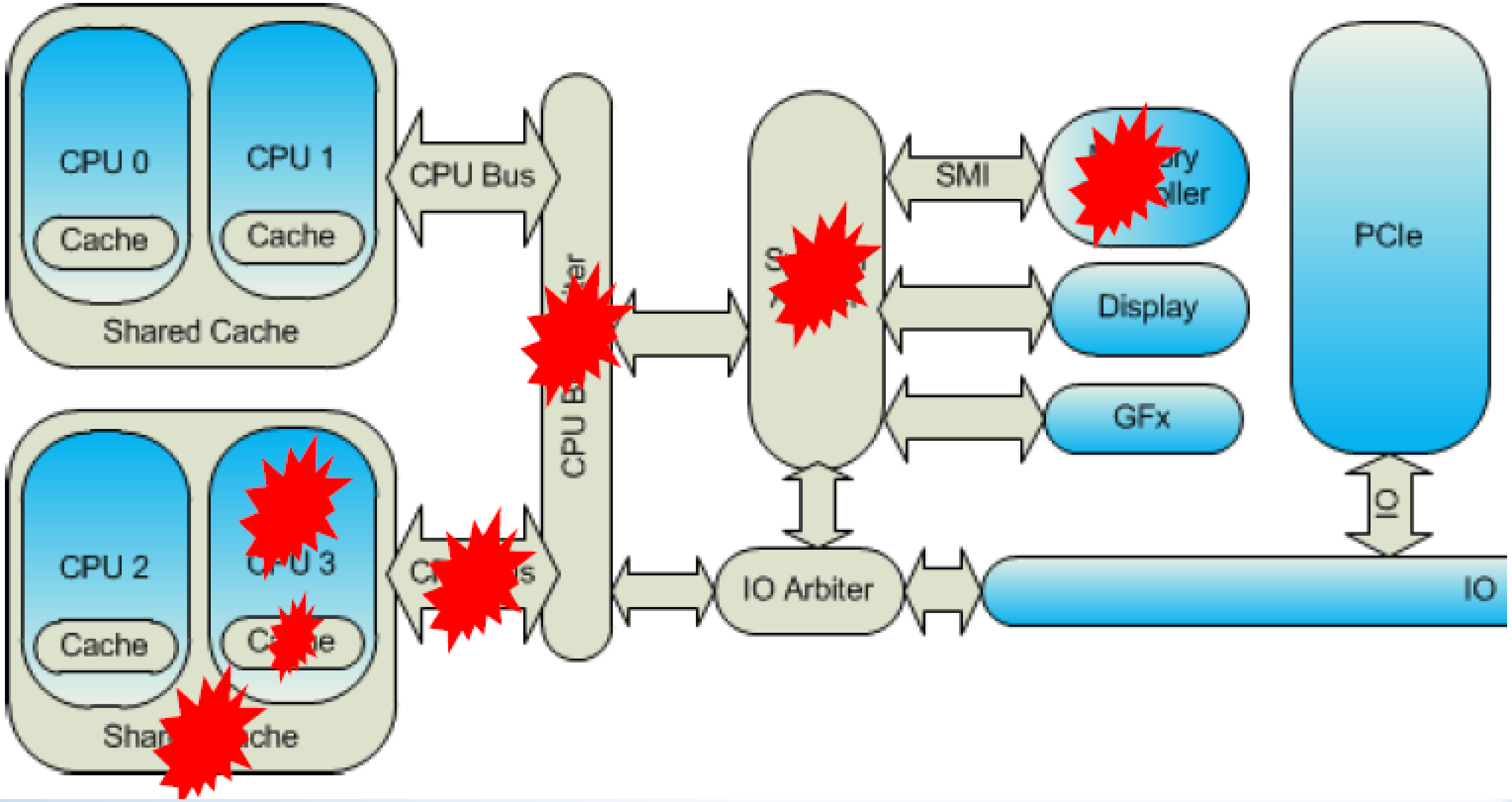
CPU Context switches to handle data



1) IO device pushes data into system memory
2) IO device notifies the CPU that data is ready to be processed
3) **CPU context switches to handle data**
4) CPU access data from memory
5) CPU processes data
6) CPU writes results to memory
7) CPU notifies the IO device that data is ready
8) IO device pulls the data from memory

- TSNA'15: James Coleman (Intel) Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

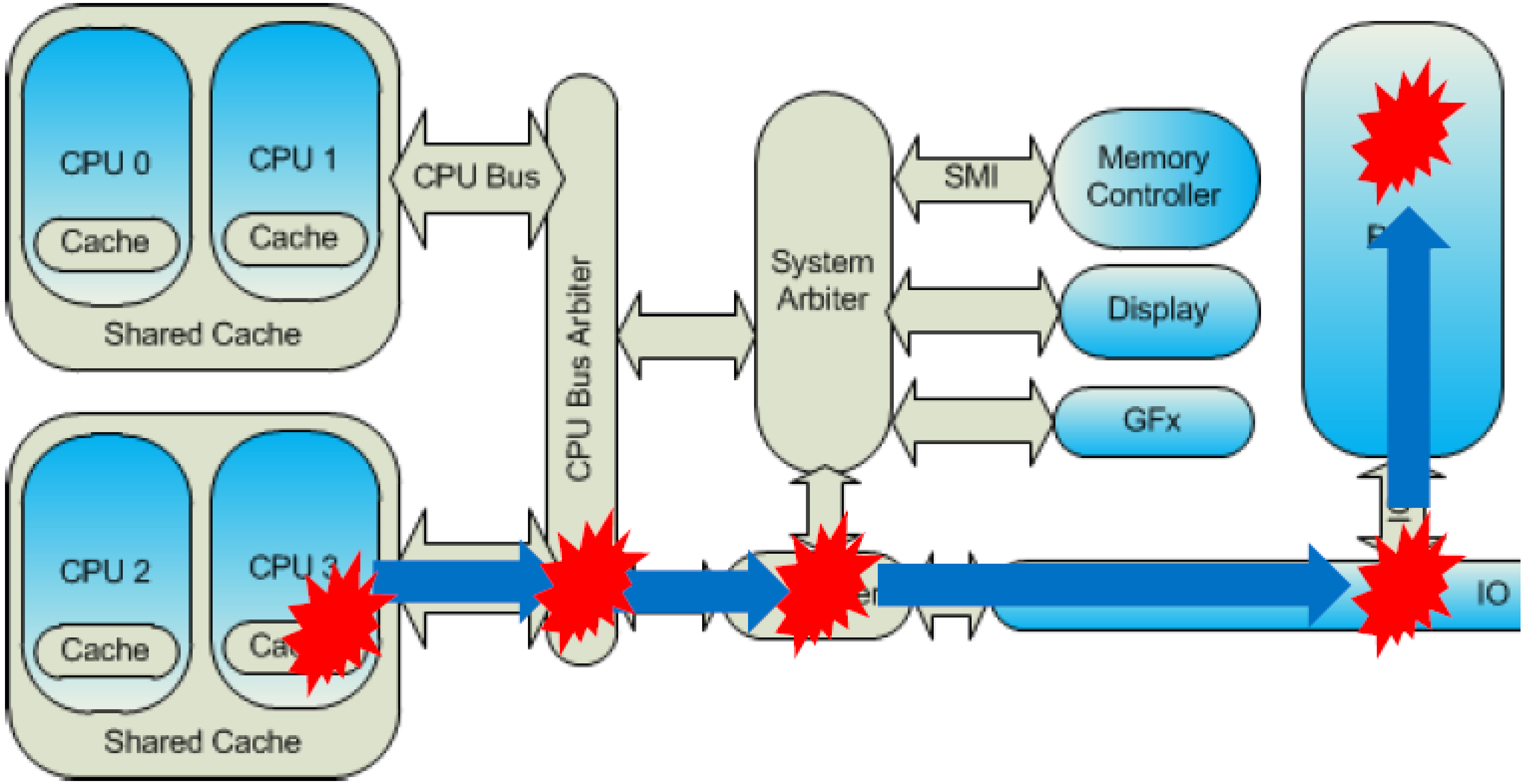
CPU: Access/Processes/Writes Data back to memory



1) IO device pushes data into system memory
2) IO device notifies the CPU that data is ready to be processed
3) CPU context switches to handle data
4) **CPU access data from memory**
5) **CPU processes data**
6) **CPU writes results to memory**
7) CPU notifies the IO device that data is ready
8) IO device pulls the data from

- TSNA'15: James Coleman (Intel) Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

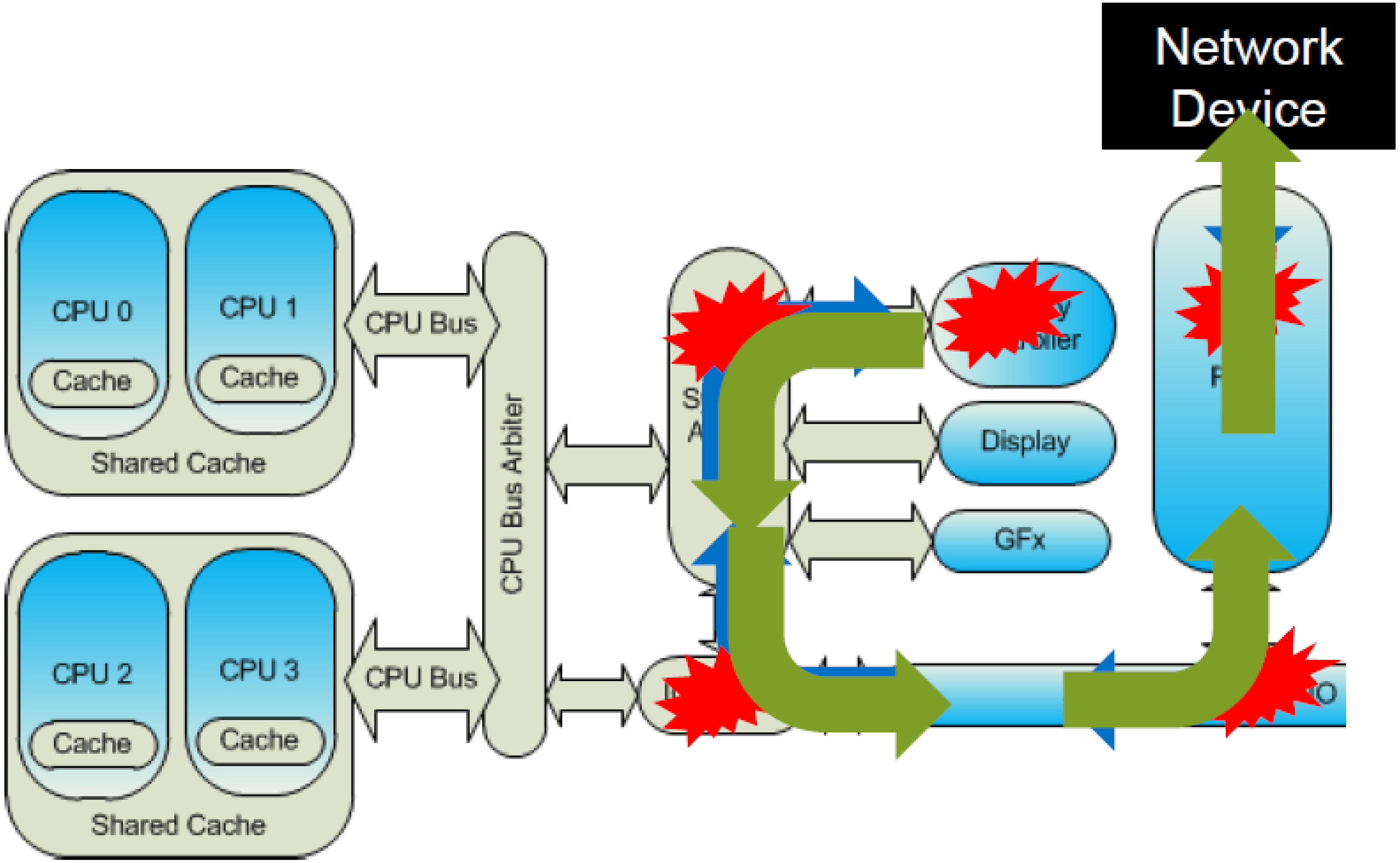
CPU Notifies the IO Device the data is ready



1) IO device pushes data into system memory
2) IO device notifies the CPU that data is ready to be processed
3) CPU context switches to handle data
4) CPU access data from memory
5) CPU processes data
6) CPU writes results to memory
7) **CPU notifies the IO device that data is ready**
8) IO device pulls the data from memory

- TSNA'15: James Coleman (Intel) Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

CPU Notifies the IO Device the data is ready



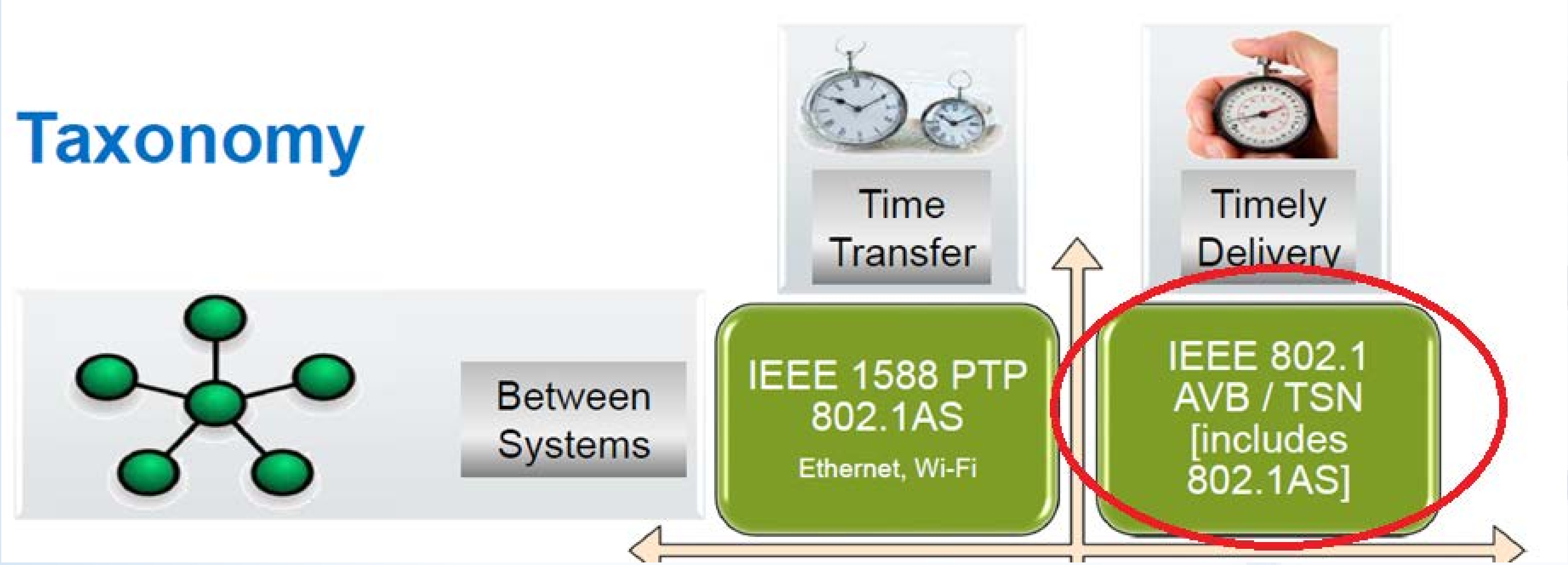
1) IO device pushes data into system memory
2) IO device notifies the CPU that data is ready to be processed
3) CPU context switches to handle data
4) CPU access data from memory
5) CPU processes data
6) CPU writes results to memory
7) CPU notifies the IO device that data is ready
8) **IO device pulls the data from memory**

- TSNA’15: James Coleman (Intel) Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

Solving Timely Delivery Within the System

- Today – You pretty much need a hard Real-Time Operating System (RTOS) to do the work – or use dedicated hardware (FPGA, etc)
  - Someone should come in and give a talk on an RTOS – oh, they did – IntervalZero's RTX64 RTOS is but one of several to choose from
    - Note, "Linux RT" is not quite there yet, but may be a viable solution 'soon' with push from Cisco and National Instruments
- Tomorrow – Don't be surprised if certain processor companies based in Hillsboro push further determinism into the CPU (memory QoS arbitration, CPU Fabric Virtual Channels, Cache locking, etc)
  - Will still need programming paradigms as Dr Lee has highlighted
    - See his team's PTides solution!

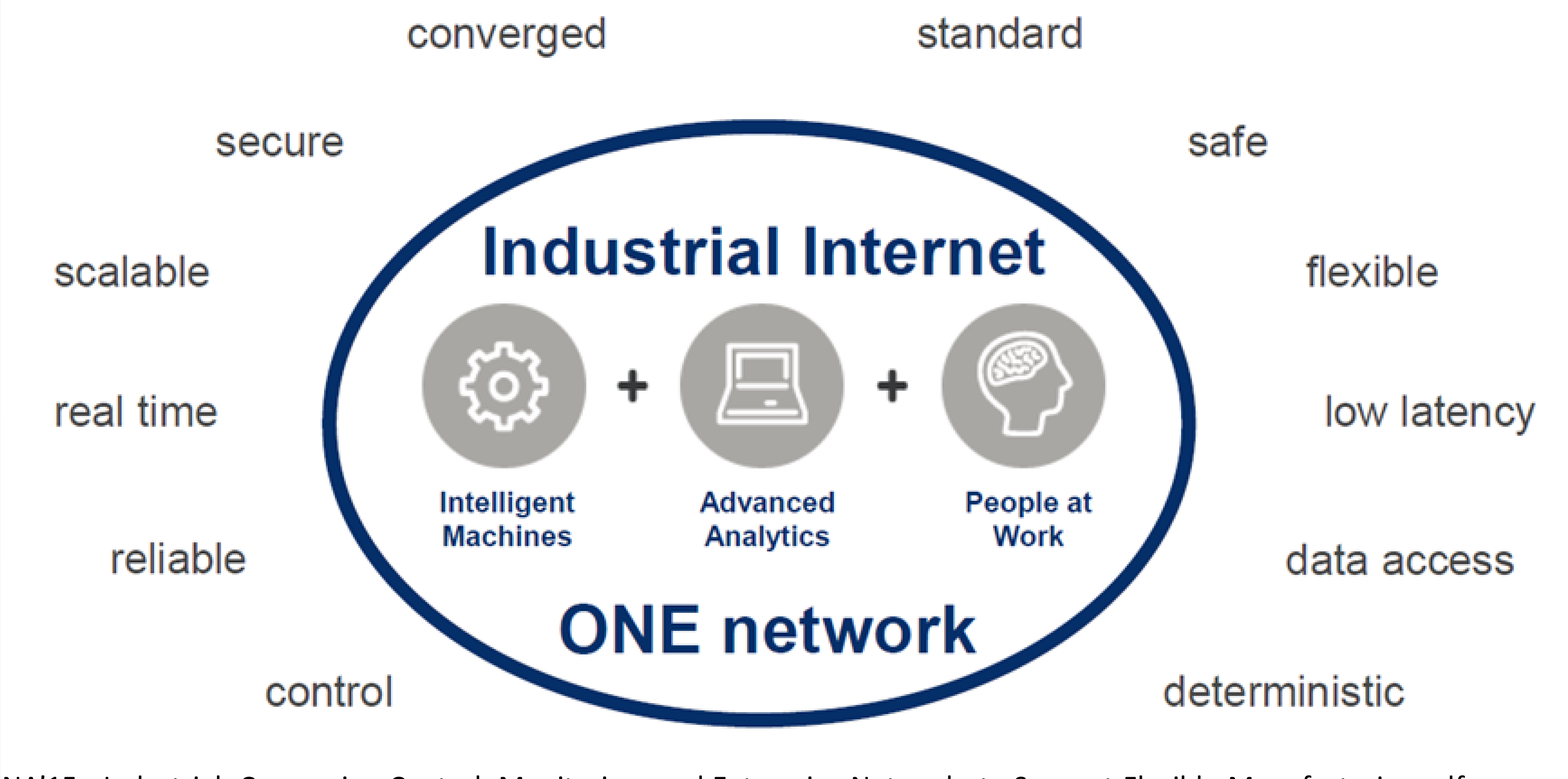
And now – to talk about TSNs…



- TSNA'15: James Coleman (Intel) Processor and OS Tuning for Event Response and Time Sensitive Systems.pdf

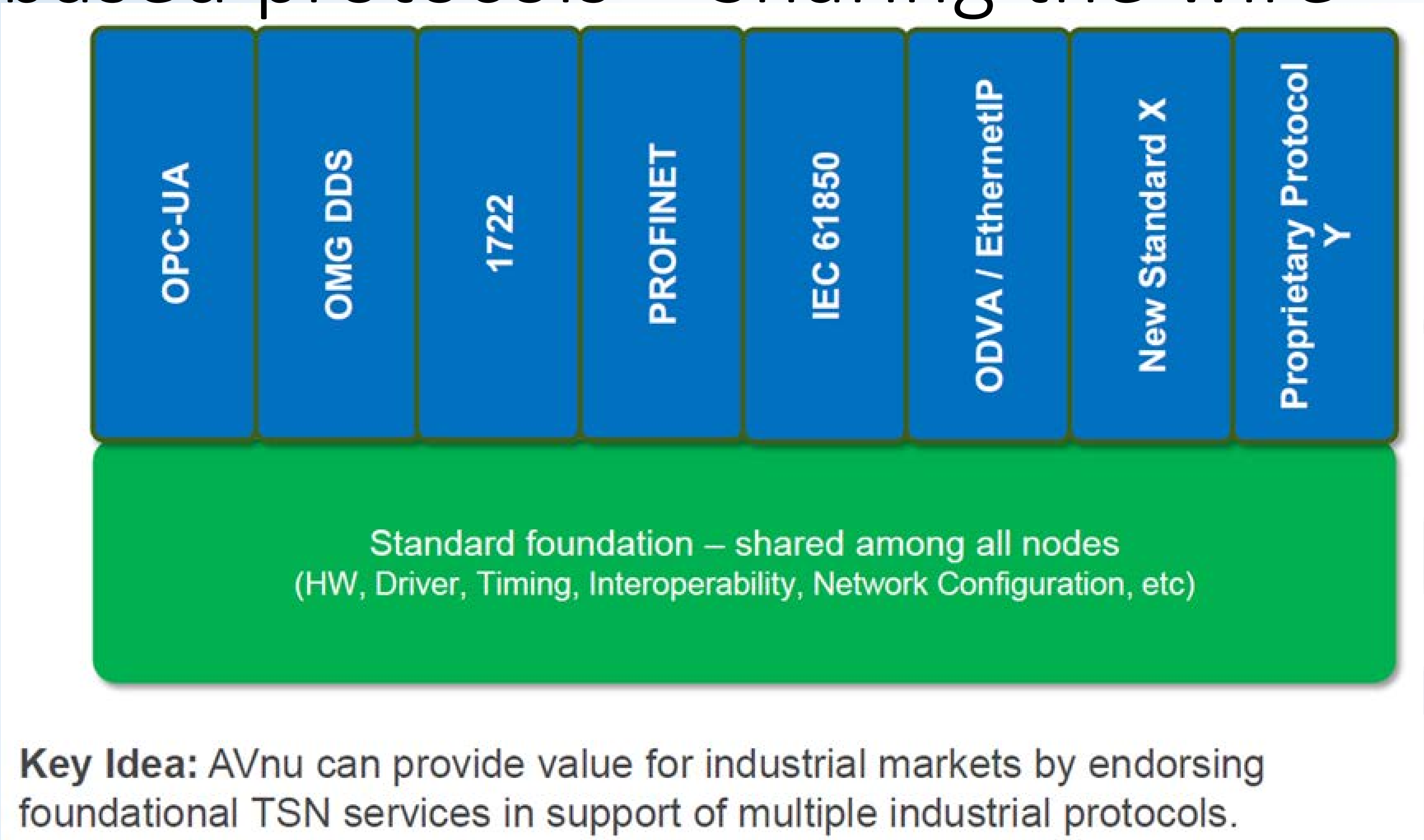
Why TSN: Industrial Internet places higher demands on the network

- Characteristics required of the network by the Industrial Internet



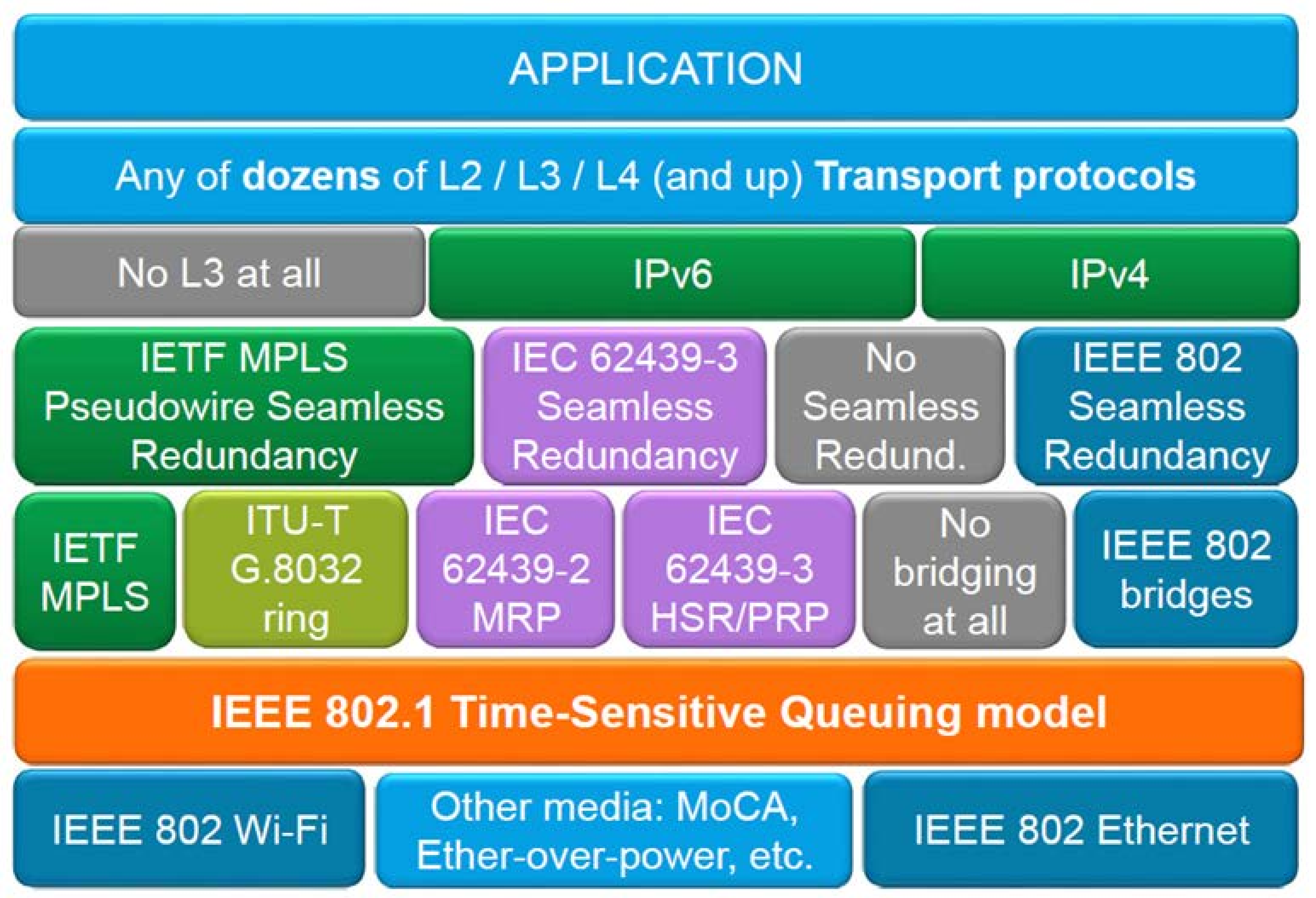
- Source: Dan Sexton (GE) TSNA’15 - Industrial; Converging Control, Monitoring, and Enterprise Networks to Support Flexible Manufacturing.pdf

TSN-based protocols – Sharing the wire



- Source: Dan Sexton (GE) TSNA'15 - Industrial; Converging Control, Monitoring, and Enterprise Networks to Support Flexible Manufacturing.pdf

Deterministic Data Plane Menu



- Source: Norm Finn (Cisco) TSNA'15 - The Magic of Layering, Support for Routers in Time Sensitive Networks.pdf

TSN / Deterministic Ethernet

Deterministic Ethernet =

Bandwidth

+ realtime communication

+ network clock sync

+ inter CPU scheduler

- Source: 07_VukotichRudolph_Audi.pdf – DE-Forum 2015

How / When

‘How’ in more detail

When – as current standard adoption grows, and referenced new standards complete

Note: the bulk of the following TSN slides are re-purposed/trimmed from the IEEE 802.1 TSN Chair, Michael Johas Teener’s presentation at ISPCS Oct’15 – A Time-Sensitive Networking Primer: Putting it all together www.ispcs.org/2015/files/K2_TimeSensitiveNetworkPrimer_Teener.pdf

Some history – The Internet and LANs

- The internet and LANs have coevolved
  - from a bunch of wildly varying link and protocol technologies …
    - remember token ring, ARCNET, LocalTalk, X.25, DecNet, SNA, TP4 …
  - to a common model based on "internet protocols" (IP) running on top of LANs based on the IEEE 802 architecture
- The driving use case was "business data"
  - bulk data transfer
  - transactions
- The important metrics were average delay and average speed
  - "best effort delivery" was the basic mode of operation
- **Anything more "timely" still used point-to-point connections and circuit switching**
  - or specialized or proprietary specifications
  - e.g. IEEE 1394 ("Firewire"), Profinet, EtherCAT

“Best effort delivery”

- According to Wikipedia:

“does not provide any guarantees that data is delivered or that a user is given a guaranteed quality of service level or a certain priority”

Hmm … what is “best” about that?

In practice, it really means:

transfer data as quickly as possible

“best” in this case means “quickest”

“Best effort” works!

- In many, many cases, “best effort” is “best”
  - in lightly loaded networks
  - where average delay is the primary metric
  - if we can’t, or don’t want to, or it’s too much trouble to differentiate between different classes of traffic
- But, of course, that’s almost never enough
  - so we have higher layer services like TCP
  - or we ignore the problem and have audio and video dropouts
- And “best effort” isn’t “best”
  - when the worst case delay is the important metric

Time

- **“Time” means delay**
  - the metric is maximum delay, not average
  - here’s where we need something better than “best effort”
- **In addition, “time” means time**
  - wall clock time, synchronization, coordination, phase locking
- **Both bounded delay and a well-known time are required in time-sensitive networks**
  - live audio and video streaming
  - control and sensor networks

Some Objectives

Live audio and video networks

- For speaker arrays …
  - the maximum synchronization error between speakers must be less than 10us …
  - and, of course, the designers want (and can use) better: down to 1us

Control and sensor systems

- Both large and small physical extent
  - Refinery or automotive assembly line: 1 km or more
  - Work cell (robot) up to 5 hops, factory up to 64 hops
  - Homes/offices/autos/aircraft/continents
  - Coexistence of bulk traffic
- Precise timing
  - Within the factory ±100 µs
  - Within the work cell (robot) ±500 ns
  - “radio over Ethernet” ±65 ns
- Deterministic and very small delays
  - Within the work cell < 5 µs
  - “radio over Ethernet” < 100µs
  - Within the factory < 125 µs (≈ 4 µs per hop)
  - Within the continent < 100ms
- Safety!
  - Redundant control / data paths with “instant” switchover
    - Seamless, or at the very least < 1 µs

A new way to implement an old model

- **Back when almost flawless streaming QoS was required …**
  - before the internet and “buffering …”
  - before mobile phones and “can you hear me now?”
  - In other words, when we had land-line circuit-switched telecom networks
- Network connections were based on “circuits”
  - nailed-up paths from end to end with deterministic characteristics
- The internet changed this model to “connections”
  - highly adaptive, very robust, but timing is very sloppy
  - use the sloppy timing budget as a way to get the data through (retries, adaptive routing, etc)
- TSN goes back to “circuits”
  - But we call them “streams”
  - but interoperates with existing models of “the internet”
  - how do we do this?

How to build a time sensitive network

1. **Provide a network-wide precision clock reference**
2. **Limit network delays to a well-known (and hopefully small) value**
3. **Keep non-time-sensitive traffic from messing things up**

- There are many possible ways to do this, but first we need to fix the low-level plumbing, and for networks based on IEEE 802 we use

IEEE802.1 Time-Sensitive Networking ….

“AVB” and “TSN” … some history

- AVB is “Audio Video Bridging”
  - IEEE 802.1 project started in 2005 largely to address the needs of the professional audio market –“IEEE 802.1 Audio Video Bridging Task Group”
    - Originally called “Residential Ethernet”, but that was too limiting and not really in scope for the IEEE 802.3 Ethernet standards
  - But also very, very useful to consumer electronics, professional video, and automotive “infotainment”
    - Associated industry compliance and marketing group called “Avnu”
- **TSN is “Time-Sensitive Networking”**
  - Capabilities of AVB-capable network were also very interesting to other groups
    - Industrial and automotive control and sensing, “IoT” to factories to motorcycles
  - Wider spectrum of requirements, “Audio/Video” was an inappropriate tag

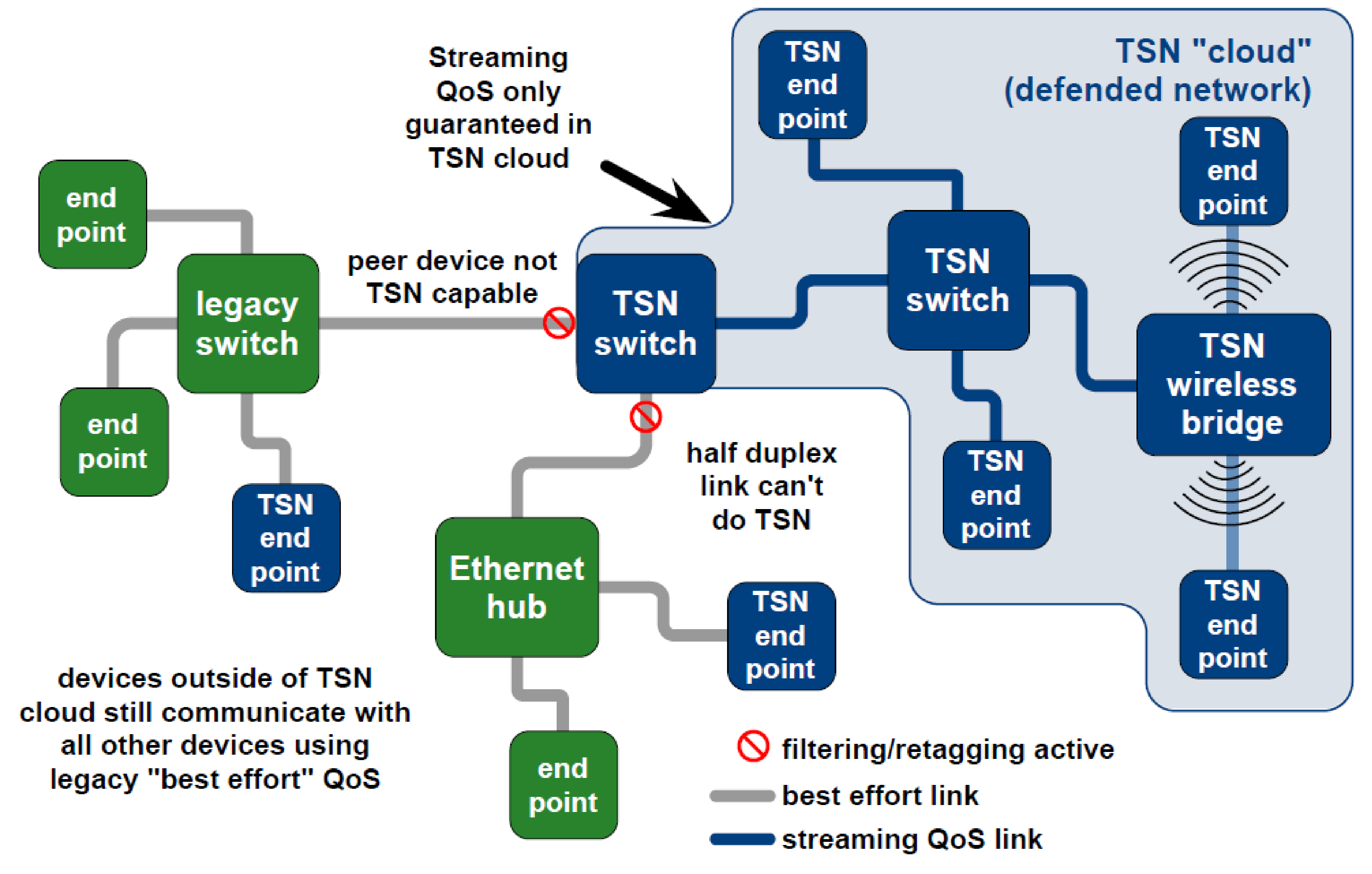
Building the foundation: IEEE 802.1 TSN

- **The IEEE 802.1 Time-sensitive Networking Task Group is "responsible for developing standards that enable time-sensitive applications over IEEE 802 networks"**
  - the IEEE 802.1 Working Group is responsible for bridging (including Ethernet "switches") between LANS
  - interoperability between networks of differing layer 2 technologies
- The primary projects include:
  - queuing and forwarding of time-sensitive streams,
    - 802.1Qav credit-based shapers, new P802.1Qbu preemption, P802.1Qbv time-aware queuing, P802.1Qch cyclic queueing, P802.1Qci input gating, and P802.1CB seamless redundancy
  - registration and reservation of time-sensitive streams,
    - 802.1Qat –a distributed "stream reservation protocol", extended in new P802.1Qcc to support preemption, scheduling, centralized control, and interaction with higher layer IETF services
  - time synchronization –IEEE Std802.1AS (based on IEEE 1588) -and
  - overall system architecture –IEEE Std802.1BA "audio video systems", P802.1CM fronthaul systems for mobile ("radio over Ethernet") and a new –unnamed-for control systems

Unified layer 2 quality of service

- Enhance network bridging
  - Define common QoS services and mapping between different layer 2 technologies
  - IEEE 802.1 is the common technology
- Common endpoint interface for QoS
  - "API" for QoS-related services for ALL layer 2 technologies
  - Toolkit for higher layers
    - IEEE 1588 time synch, IEEE 1722 and RTP streaming, IEEE 1722.1, RSVP and SIP session establishment
  - Provide network independence for endpoints without giving up QoS
    - Endpoints don't have to be aware of the particular link technologies (Ethernet, WiFi, EPON, MoCA, powerline, etc.) … common API, support for multiple link types in a path

TSN Cloud



TSN "cloud" (defended network)
Streaming QoS only guaranteed in TSN cloud
end point
legacy switch
peer device not TSN capable
TSN switch
TSN end point
TSN switch
TSN end point
TSN wireless bridge
end point
TSN end point
half duplex link can't do TSN
TSN end point
Ethernet hub
TSN end point
TSN end point
devices outside of TSN cloud still communicate with all other devices using legacy "best effort" QoS
end point
filtering/retagging active
best effort link
streaming QoS link

Identifying Streams

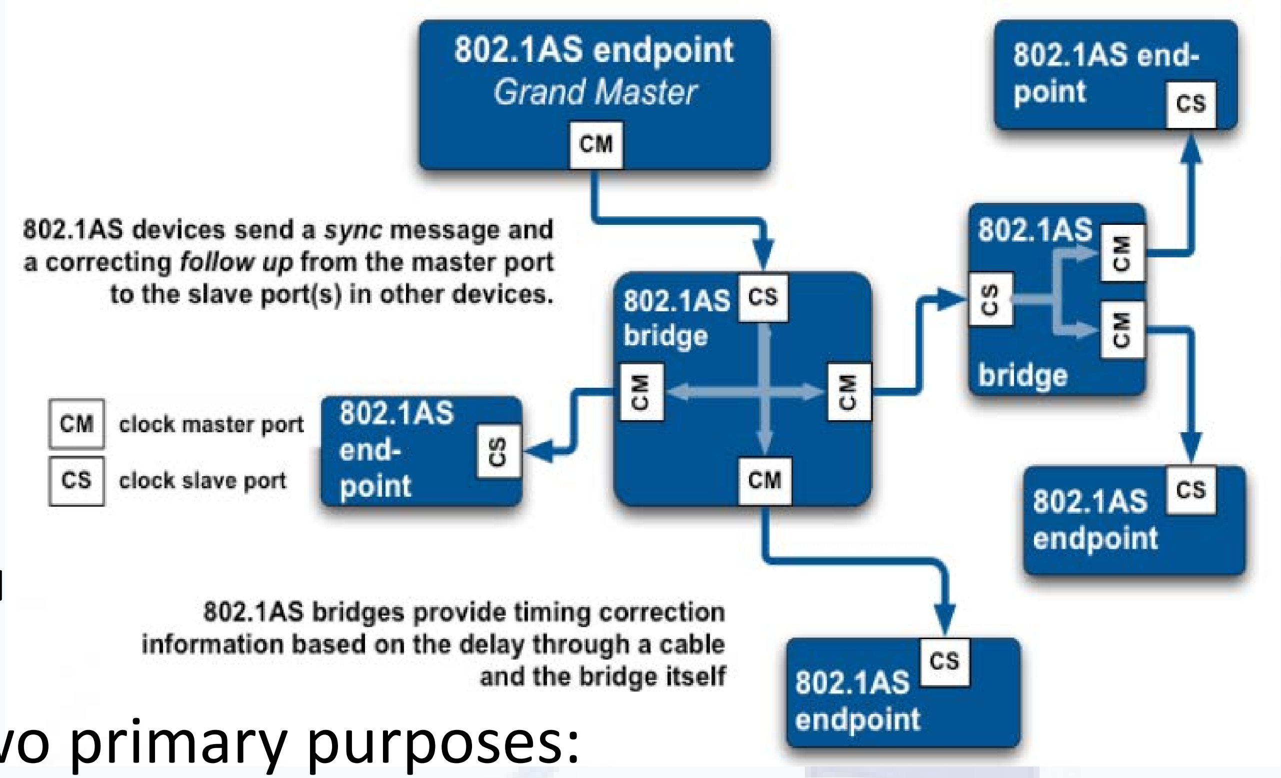
- **Providing QoS for a stream first requires some way to identify the stream**
- TSN uses three types of identifying labels:
  - “stream ID”, which is a 48-bit EUI-48 (usually the MAC source address) concatenated with a 16-bit handle to differentiate different streams from the same source
  - “traffic class”, which is determined by the 3 priority bits
    - TSN normally only uses one or two classes, by default 2 and 3
  - “stream destination address”, which, oddly enough, is the MAC destination address combined with a VLAN ID
    - TSN normally uses locally-managed or group addresses (sometimes called “multicast addresses”)

Using Stream Labels

- Stream ID is used for management
  - Unique within a TSN cloud
  - Used by the "control plane" to reserve resources
- **Traffic class and stream destination address identify which "data path resources" are used by a stream**
  - Normally the traffic class and stream destination address stay the same across the network from end to end
    - Not required, however … "circuit" is an abstract concept, and the actual labels used for packets to identify a circuit can change on a hop-by-hop basis
    - A circuit can be labeled by an IP octuple by a transmitter, and relabeled by the originating end station or by a bridge on the network edge.
  - Used by the "data plane" to provide QoS and forwarding (routing) services

IEEE 802.1AS: Precise synchronization

- **All "time-aware systems" participate in a "native IEEE 802 layer 2 profile" of the IEEE 1588v2 Precision Time Protocol**
  - a very tightly defined subset of standard 1588v2 for Ethernet
    - Compatible enhancements for much faster clock locking and easier/lower cost filtering at endpoints
  - superset of 1588v2 to support 802.11 WiFi, EPON and "coordinated shared networks"



- This precise synchronization has two primary purposes:
  - allow multiple streams to be synchronized and
  - provide a common time base for sampling data streams at a source device and presenting those streams at the destination device with the same relative timing

802.1AS and 1588 Default Profile differences

- All devices participate in BMCA
  - BMCA almost identical to default 1588-2008
  - Intermediate systems (bridges) have same performance as transparent clock for Ethernet, so no “boundary clock” vs “transparent clock” issue
- No “partial on path” support
  - Sync only between participating devices, non-participating device blocks sync path
- Fewer options
  - Ethernet is L2 two-step with peer delay
  - Single domain
- Ports connect via link-technology-dependent methods
  - WiFi uses 802.11v “timing measurement”
  - EPON and other coordinated shared network links also use link-dependent delay measurement technology
- All filtering done at ordinary clocks (endpoints)
  - Intermediate systems (bridges/switches) syntonizer with GM almost instantly using neighbor rate ratio calculation and sharing via TLV included in Sync/Follow up.
  - Endpoints lock with GM within a few seconds max

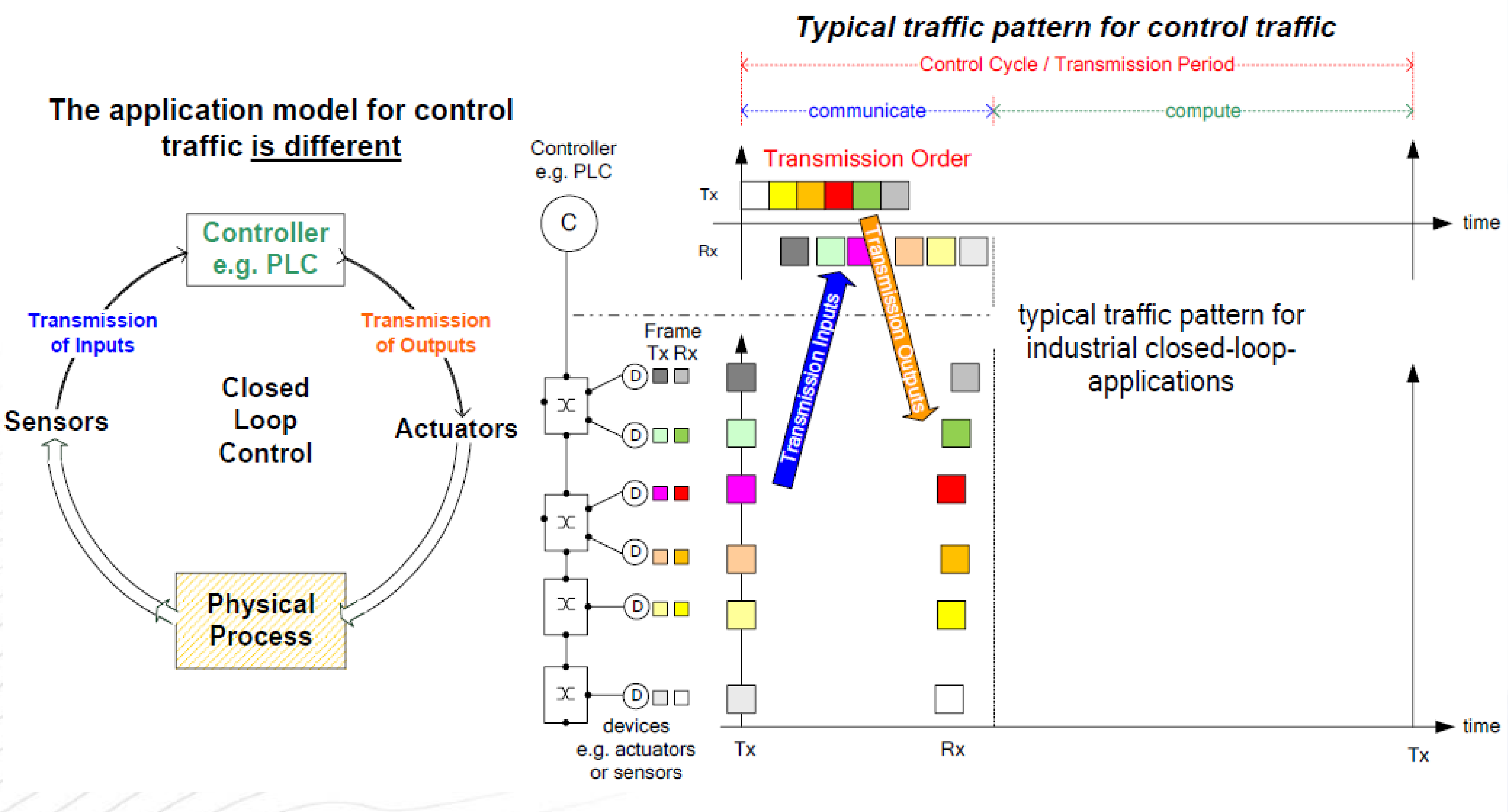
Time Sync Updates

IEEE 802.1AS-revision is an update to 802.1AS for:

- Enhanced link support
  - Support for *all* of Ethernet, including link aggregation
    - Working with IEEE 802.3 to improve delay reporting and allow proper standardization of “one-step” time stamping
  - Support for 802.11 “fine timing measurement” for picosecond-level timestamps
- Improve performance and usability
  - Support multiple domains
    - Domain zero required, must be locked to TAI, other domains may be “working clock”
  - Responsiveness and reliability, support for “one step” links
    - “one step” is a port attribute, so a system can have ports operating in both one-and two-step modes
    - Master port sync transmission can be locked to slave port sync (like a transparent clock) or not (like a boundary clock)
  - “sourcePortID” is identifier of GM, not just immediate upstream system
  - Timing quality reporting via managed objects and/or signalling
  - Explicit support for centrally-managed systems via port role
- Start the process towards protocol unification
  - End the 1588 vs 802.1AS vs NTP confusion
  - 802.1 is coordinating with the 1588 revision project

Reducing delays

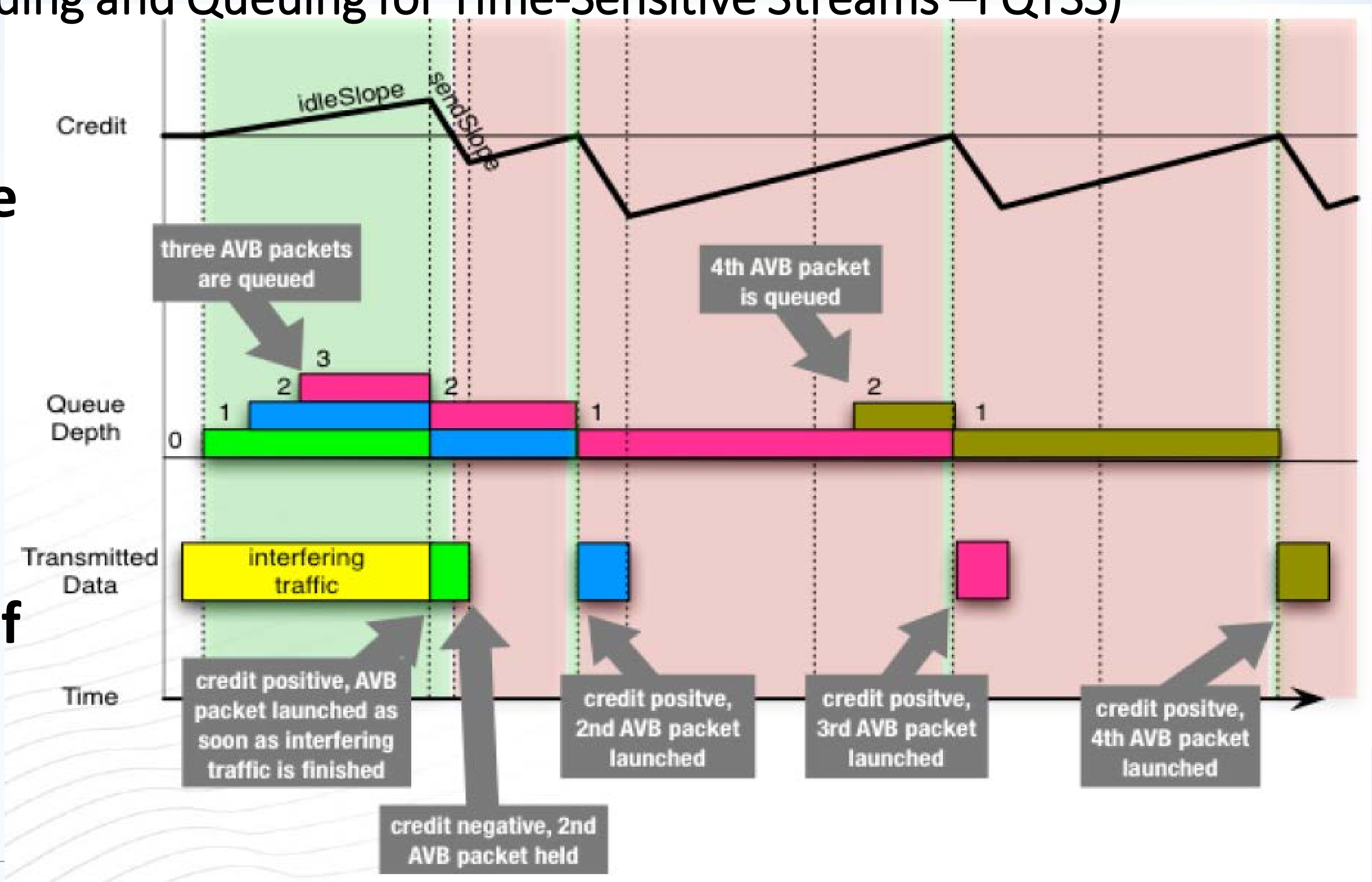
- TSN applications do NOT care about average delay, nor fastest delivery
- **The important number is “worst case” delay**



Reducing Traffic Delays

(IEEE Std802.1Qav –Forwarding and Queuing for Time-Sensitive Streams –FQTSS)

- **Devices in AVB network must "shape traffic"**
- **Schedule transmission of packets to prevent bunching, which causes overloading of network resources**

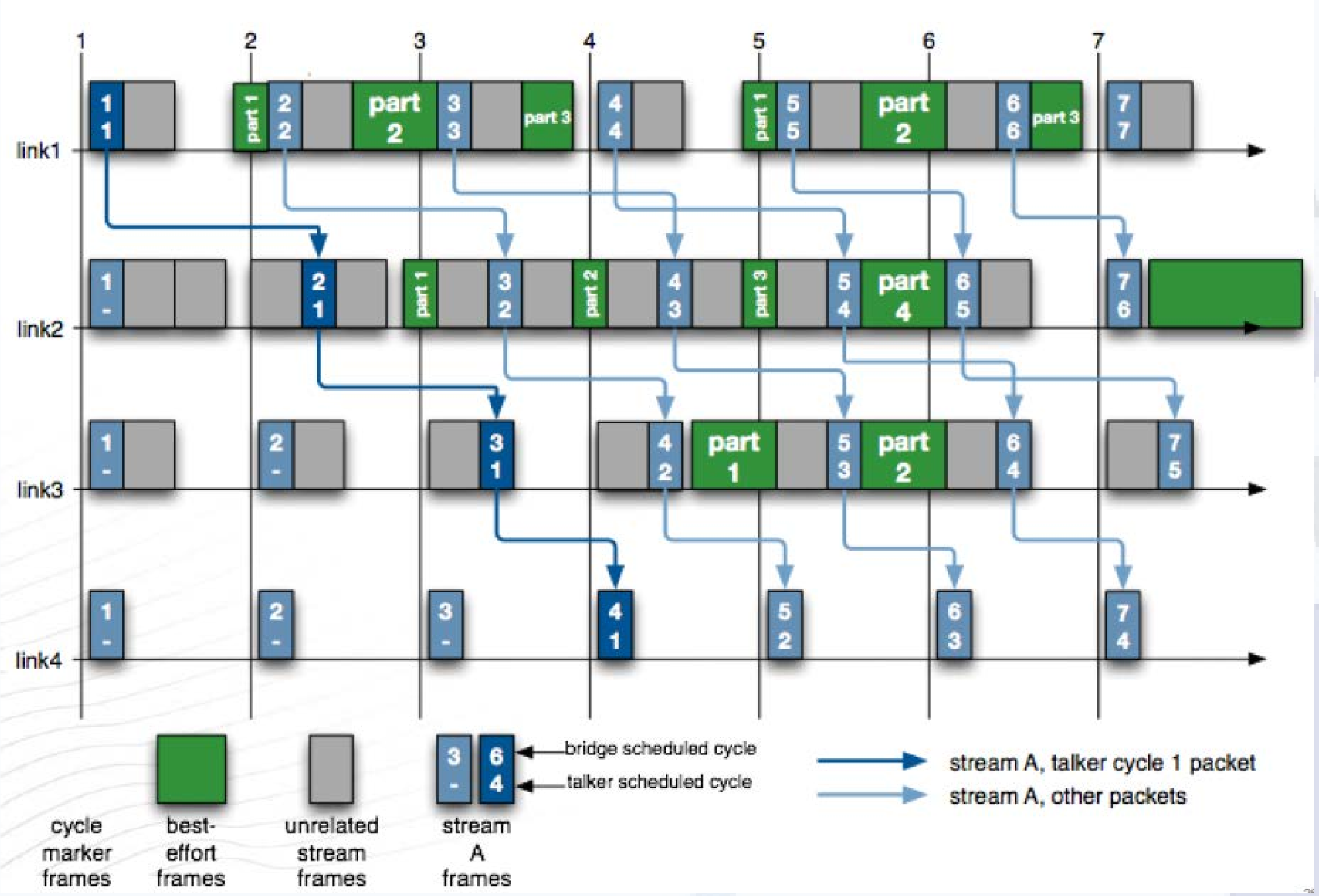


Improvements for streaming
100% reliable delays and fixed delivery jitter

- Existing FQTSS has some issues
  - Pathological topologies can result in increased delays
  - Worst case delays are topology dependent, not just a count of hops
    - Buffer requirements in switches are topology dependent
- New "Cyclic Queuing and Forwarding" (CQF) 802.1Qch
  - Every switch introduces a fixed delay for each stream in a particular traffic class
    - Buffer requirements are fixed by switch design, topology of network has no effect
    - Delivery jitter is fixed, based only on traffic class
  - Intent is to *replace* the existing FQTSS
    - Compatible with FQTSS, worst-case delay improves with the number of CQF shapers in the path

Completely deterministic delays
(IEEE 802.1Qch – Cyclic Queuing and Forwarding – CQF)

- **Expected to be combined with 802.1Qbu/802.3br (preemption) to reduce limits on minimum cycle time**



Limits on Delay

- That’s nice, but what can we do that’s better?

The fundamental problem is interfering traffic!

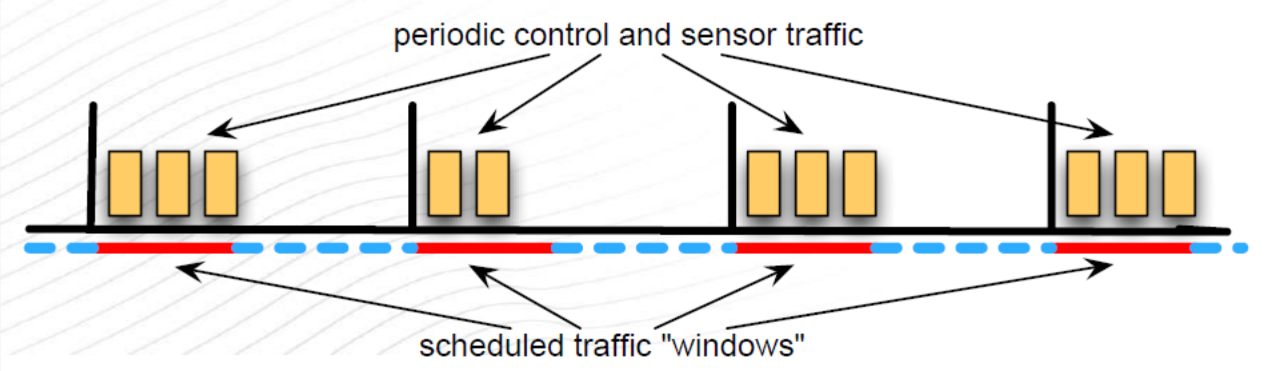
If a packet is to be transmitted on a particular egress port, then all traffic, regardless of the priority, must wait until the egress port has completed transmitting that packet.

(head of queue blocking)

Avoiding Interfering traffic

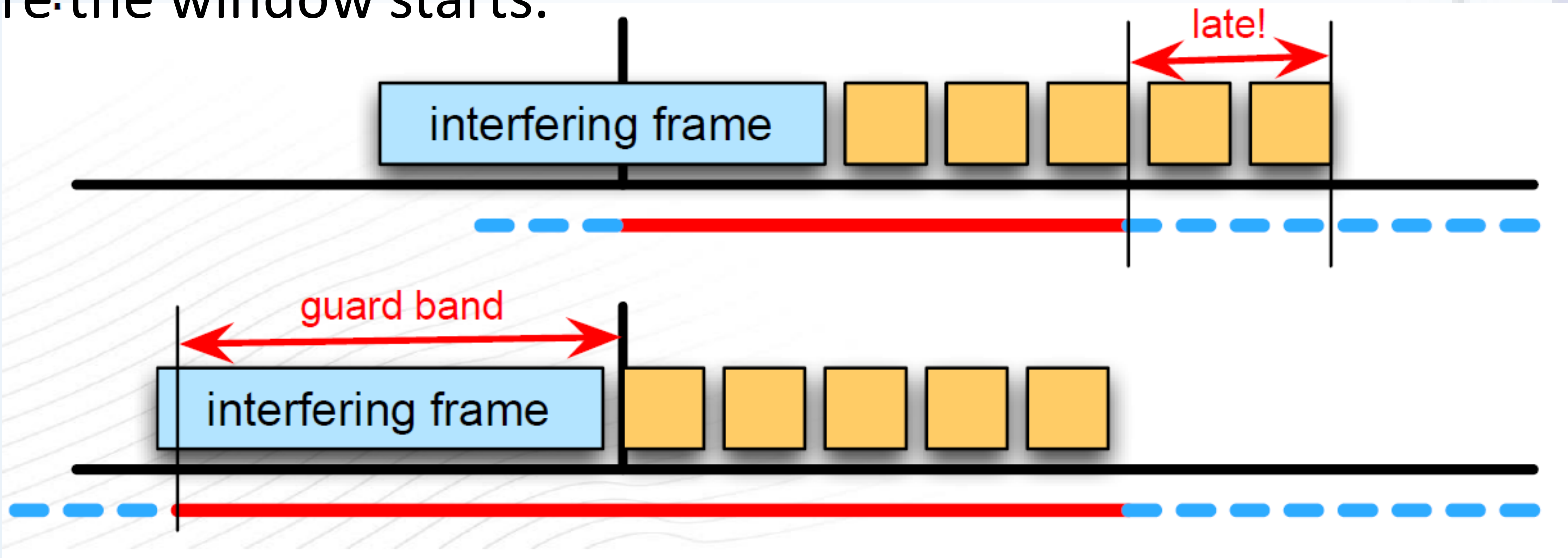
802.1Qbv Time-Aware Shaper

- Make switches aware of the cycle time for control traffic:
  - Block non-control traffic during particular windows of time to ensure that the egress port for a control stream is idle when the control traffic is expected.
  - Each egress port would have a separate schedule.
- Nontrivial calculation in nontrivial networks:
  - Requires a fully managed network.
  - This is a well-understood but difficult problem currently implemented in proprietary networks such as Siemens' "Profinet."



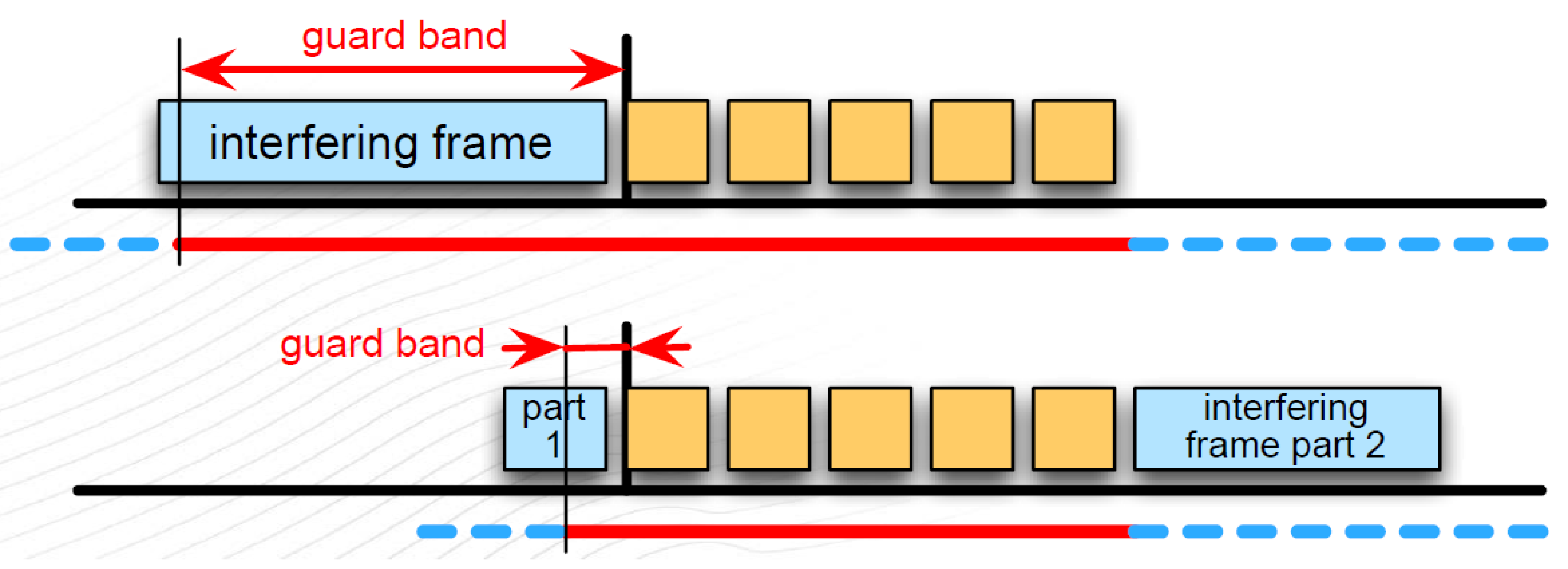
Time-Aware Shaper Issues

- A Guard Band is necessary
- If an interfering frame begins transmission just before the start of a reserved time period, it can extend critical transmissions outside the window.
- Therefore a guard band equal in size to the largest possible interfering frame is required before the window starts.



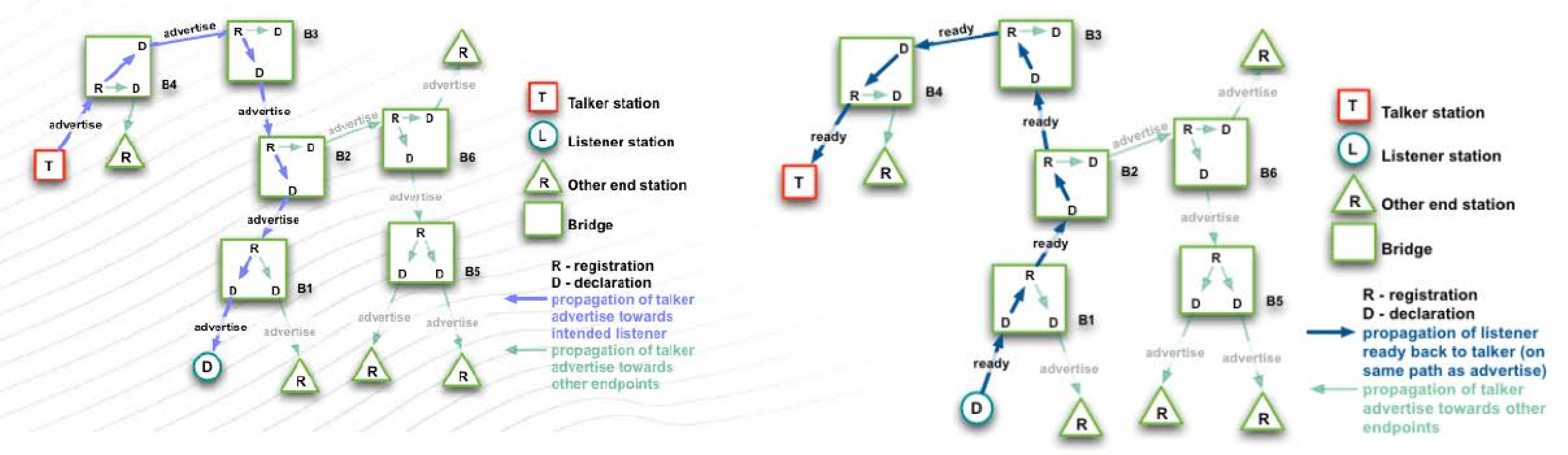
Reducing the guard band

- **Preemption (802.1Qbu/802.3br) is a solution**
- **If preemption is used, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame.**



Admission controls - IEEE Srd 802.1Qat

- **Priorities and shaping work only if the network resources are available along the entire path from the talker to the listener(s)**
  - AVB "talkers" guarantee the path to the listener is available and reserve the resources
- Done via a "Multiple Registration Protocol" application: SRP ("Stream Reservation Protocol")
  - Registers streams as a source MAC address combined with a higher level ID (frequently the IP port address)
  - Reserves resources for streams based on bandwidth requirements and latency class



Enhanced Stream Reservation 802.1Qcc

- **Explicit interoperation with "God box" centrally-managed systems**
  - Allow centrally-managed and ad hoc systems to coexist
  - Management compatible with IETF YANG/NETCONF
    - Or the simpler IEEE 1722.1 and IETF "constrained application" protocols
- **Explicit interoperation with higher-level reservations**
  - A "UNI" … unified network interface, a common way to request L2 services
  - Higher layers (such as RSVP) can use this as an API the same way IEEE 1722.1 uses the existing SRP
- Reduce the size and frequency of reservation messages:
  - *Compatible with existing 802.1Qat Stream Reservation Protocol*
  - Relaxed timers, updates only on link state or reservation changes

TSN management use models

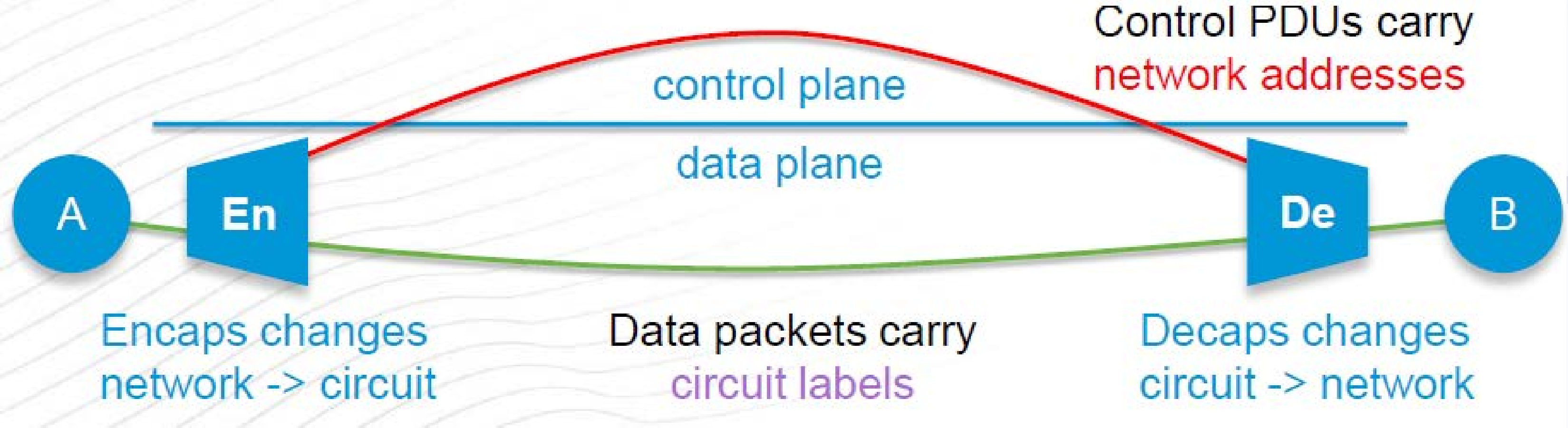
- Self-managed/distributed management applications
  - Small office / studios / “field” deployments
  - Many ad-hoc arrangements
- Centrally managed applications
  - Very large systems –or -
  - Highest performance / lowest delays / most reliable –or -
  - Commercial constraints for remote management
    - “my movie is breaking up”
  - “best” path or “safest” path or “most efficient” path hard to determine without global knowledge

Use what exists

- Ad hoc / distributed management uses …
  - RSTP (simplest) or 802.1Qca (IS-IS-based path control –complex but capable) for path selection
  - New SRP for resource reservation using existing paths
- Centralized management could …
  - Use new SRP as the “UNI” … the unified network interface
    - The “API” for network endpoints to request services
  - Endpoints would not need to know what kind of network management is being used
- This requires …
  - Updates to SRP to carry information useful for centralized managers to determine appropriate paths
  - Selection of a “default” centralized management system

Example: existing IP-based applications

- **TSN streams are identified by both a traffic class tag and a locally-managed group destination MAC address ("circuit label").**
    - IP stacks don't work well with locally-managed addresses
    - Never had to in the past, no mechanisms exist
- **Use enhanced SRP (802.1Qcc) to carry tunneling information (address, traffic class) between endpoints**
    - Enables straight-forward encapsulation/deencapsulation("encap/decap") using the TSN stream identification function
- **Higher level reservation system (IEEE 1722.1, new RSVP, centralized management) must in**

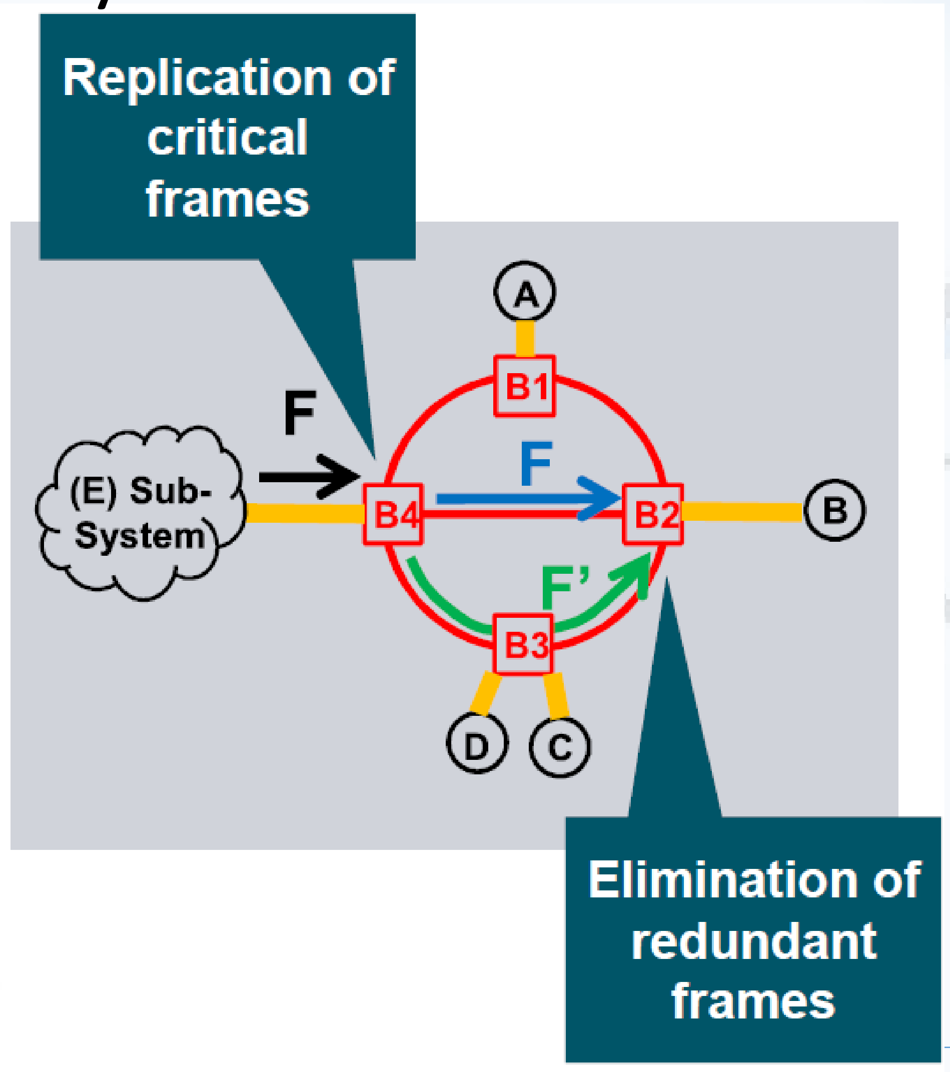


Centralized management

- Newest recommendations are YANG/NETCONF
  - YANG is a much-improved way to describe the data model for configuration and status
    - Replacement for SMI MIB modules, *MUCH* easier for humans to read and parse
  - NETCONF is the protocol used to use the YANG models for actually configuring network devices and getting status
    - Somewhat complex/session based using SSL, uses XML encoding, very verbose
  - Simplified, but compatible subsets to be considered
    - RESTCONF is an HTTP, sessionless version of NETCONF
    - Proposed binary encodings using well-known schema and IEEE 1722.1 or “constrained system” protocols
- PCEP (path control element protocol)
  - Existing protocol used for router management in traffic-engineered networks

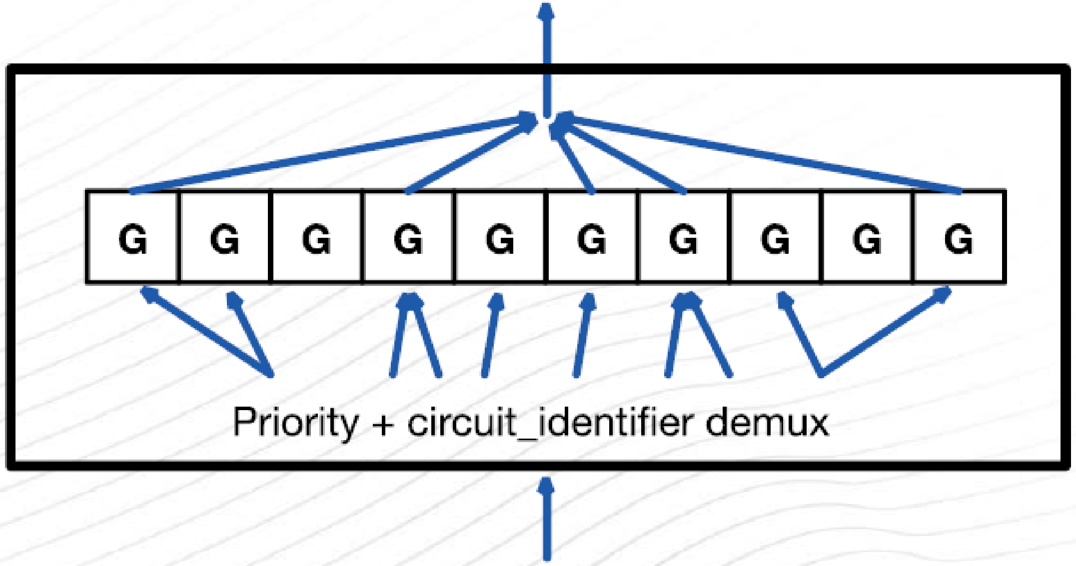
802.1CB Seamless redundancy

- **Selective packet replication based on address/traffic class and path information**
  - done by TSN stream identification plus "sequence generation" functions
- Duplicate frame elimination
  - based on address/traffic class (TSN stream), sequence number and timing
  - timing information needed to limit memory needed for duplicate frame detection
- Compatible with existing industrial architectures
  - E.g., HSR, PRP
- Management NOT TRIVIAL!
  - Almost certainly requires centralized management



802.1Qci – Input gates

- **Need to provide protection for the QoS and redundancy features**
  - Mainly to protect against software bugs on endpoints, but maybe switches/bridges, maybe hostile devices
- Make sure streams don't exceed their contracts!
  - Excess bandwidth, burst sizes, packet sizes, misuse of labels



Each gate has:

1. A pass / don't pass switch
2. Optional policers
3. Counters
4. Service class output specifier

All of these perhaps based on time

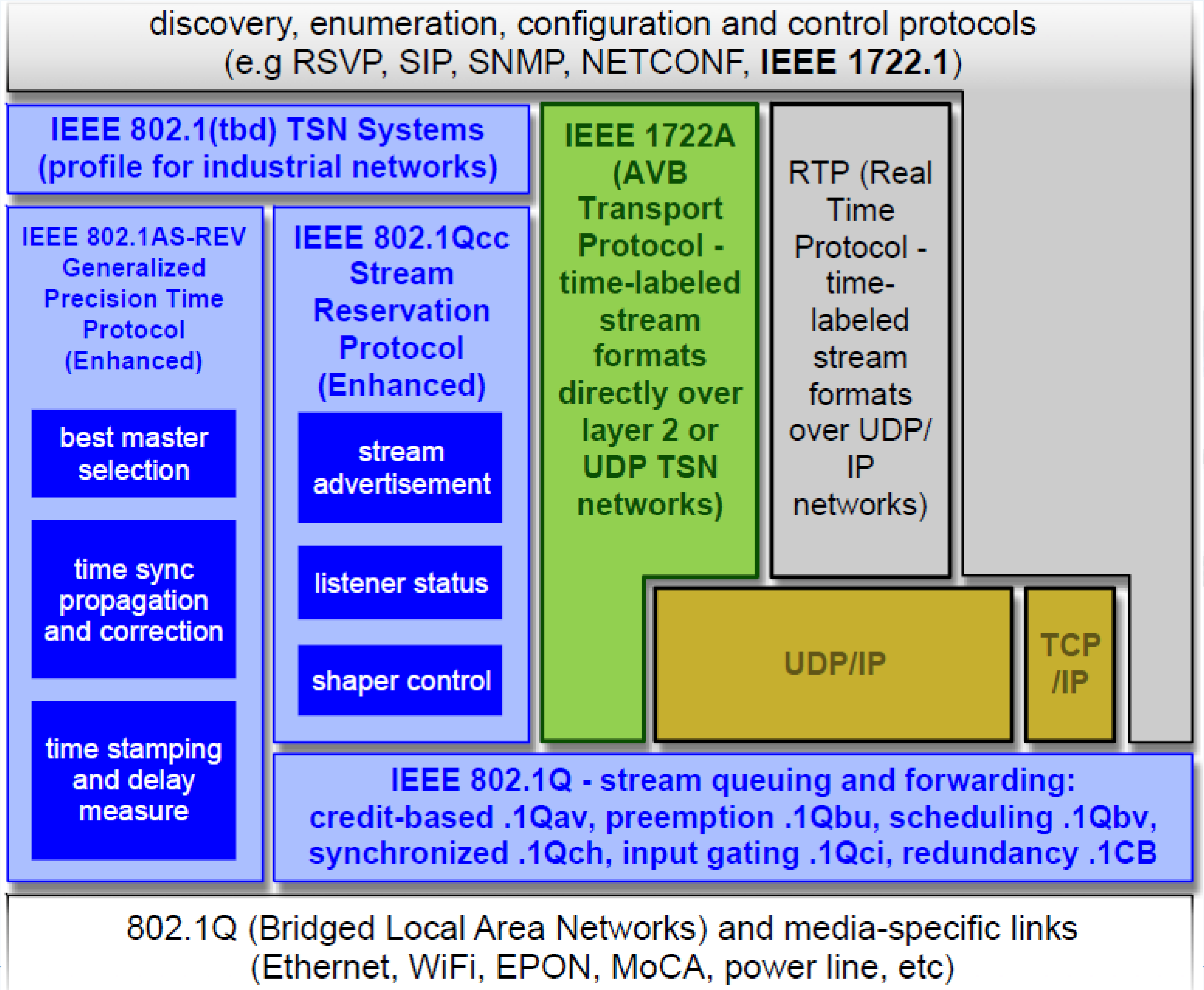
Higher Layer standards: IEEE

- IEEE 1722 "AVB transport" (AVBTP)1
  - Stream format for time-sensitive streams
    - Based on IEC 61883 formats used in IEEE 1394/Firewire… common in the pro audio market
    - Originally assumed direct IEEE 802 addressing, no IP encapsulation
    - New 1722-revision adds many new simplified formats, UDP/IP encapsulation, security
  - Method for allocating group MAC addresses used by SRP as stream ID's
- IEEE 1722.1 "Discovery, enumeration, configuration and control" (DECC)
  - Fills in the protocols needed to build systems based directly on AVB and 1722
  - Assumes direct 802 addressing, but UDP/IP encapsulation will come with 1722-revision

Higher layer standards: IETF and others

- RTP (real time protocol) for stream encoding
  - Extremely broad range of applications from VoIP to TV broadcast
    - Many, many options and profiles … no such thing as a universal RTP
    - Works well in heavily managed networks, highly variable quality otherwise
- RSVP (path and QoS reservations)
  - Mostly unsuccessful, tied to “INTSERV” … per-stream QoS
    - Not scalable, too many options, not tied to capabilities of lower layers
  - Somewhat more successful as a path reservation system for some MPLS systems
    - RSVP-TE (traffic engineered)
- AES (Audio Engineering Society)
  - Incomplete set of standards on how to use RTP and 1588 for transport
- SMPTE (Society of Motion Picture and Television Engineers)
  - Set of standards on how to use RTP for uncompressed video
  - Working on methods to use 1588 for studio timing reference

Putting it all together
TSN Layering



With TSN services we have…

"802 everywhere"

- 802.3, 802.11 and 802.15 links are scaling to all sizes, speeds, costs, power
  - 10/100/1G single pair Ethernet, 25G two pair short range, 100G+, etc, etc
  - Multi-Gigabit wireless, or years-long operation on a coin cell
- Wide area networks with 802 architecture lower layers can now provide "universal service"
  - Existing transaction and bulk transfer user models (the traditional internet)
  - Existing streaming services with MUCH better QoS (no more "buffering …" messages from Netflix, shorter delays for voice/video calls … Skype that really works).
  - New time-based data exchange (the "industrial internet")
- Scaling down to a smaller physical extent …
  - Mobile front haul … "Radio over Ethernet"
  - Industrial monitoring and control systems
  - Stadium-extent phased array speaker systems
  - Echo-free airport announcing systems
  - Time-synchronized server farms with vastly reduced internal traffic delays
- Scaling down to the room or desk …
  - Replacing HDMI, Display Port, FireWire and any other A/V interconnect

Wrap Up

Thank you for your interest in Time Sensitive Networks / IoIT / Deterministic … … … …

Please review the slides from TSNA, WSTS, ISPCS, Deterministic Ethernet Forum, and more

Review Dr Edward Lee's slides on Internet of Intelligent Things
https://www.atis.org/WSTS/papers/3-3-1_UCBerkeley_Lee_LeveragingClocks.pdf
Updated version at: https://goo.gl/r200rl

Review Michael Johas Teener's primer slides on TSN for additional detail
http://www.ispcs.org/2015/files/K2_TimeSensitiveNetworkPrimer_Teener.pdf

Get Involved !!!

Thank You For Your Time

iol.unh.edu

AVnu Testing Service and 1588/Precision Time Protocol

Bob Noseworthy - ren@iol.unh.edu

For more information please refer to

http://www.iol.unh.edu/avnu

http:/www.iol.unh.edu/1588