SDN-BASED CONFIGURATION SOLUTION FOR IEEE 802.1 TIME SENSITIVE NETWORKING (TSN)

SIWAR BEN HADJ SAID, QUANG HUY TRUONG, AND MICHAEL BOC

RTN 2018, 03 July 2018, Barcelone | Michael BOC

CONTEXT

- **Switch to IEEE standard Ethernet in Industrial and automotive domains**
  - Killer applications (or at least advanced prototype of)
    - Autonomous driving, Artificial intelligence, Cloud, 3D reconstruction, augmented reality
    - Monitoring for zero-defect manufacturing, learning robots
  - High bandwidth (100Mbit/s, 1Gbit/s, 10Gbit/s …)
    - Compared to LIN/CAN

  ➔ Requirement: protect existing assets traffics from new peripherals traffics

- **Potential solution: IEEE 802.1 Time-Sensitive Networks**
  - Layer-2 mechanisms to segregate traffics and to manage their latency
    - Time synchronization (AS), Traffic shaping (Qav), scheduling (Qbv), Frame Preemption (Qbu) to name a few
    - Different levels of traffic forwarding management according to the priority level
      - Latency protection services for “critical” flows, best effort for other flows
  - Can provide performance guarantee

We have a winner! nuh?

CHALLENGES

- **Network engineering & Configuration**
  - Performance guarantee implies constraints on the topology, PHY and MAC
    - E.g., up to 7 hops deep, symmetric propagation delay, full-duplex, etc.
  - Deep knowledge on characteristics of traffics
    - To setup guard bands (max packet size), shaping parameters, gates cycle, etc.

- **Traffic Engineering**
  - Shift of paradigm: the packet must arrive before the deadline and not as fast as possible
  - Best effort (BE) traffic may not be as BE as one think
    - Video traffic (not critical) may be useless if too much latency
    - File transfer (not critical, e.g., firmware update) may have some restrictions on the minimum bandwidth
  - “Critical” flows may support high latency

CONFIGURATION PROCESS

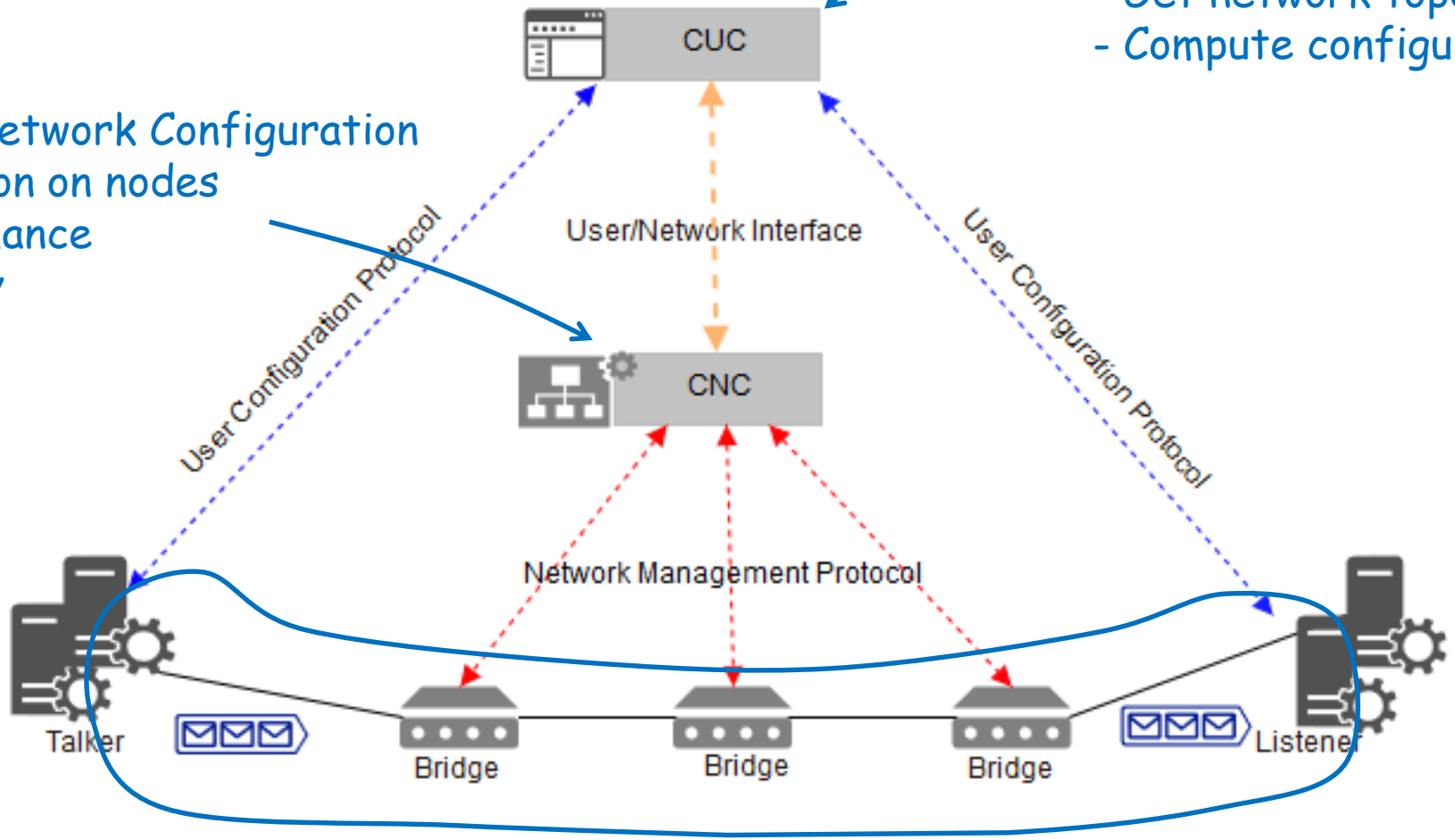
- **TSN network configuration**
  - Simulator/Emulator/Network calculus to find the right configurations
    - Time-consuming, run each time a new traffic is proposed
  - Manual configuration
    - SSH/FTP/SD Card to each node to update firmware
    - Slow process that may prevent dynamism in network topology/evolution
  - OPC-UA
    - How to learn/adapt traffic patterns to fit in the budget
    - Pertinent on endpoints but not on bridges

- **Automation of network configuration**
  - Ongoing standardization efforts to define the building blocks
  - IEEE P802.1Qcc (Stream reservation (SR) protocol and enhancements)
    - The fully centralized architecture has some tractions
  - IETF Deterministic Networks (DetNet) Working Group
    - The fully centralized architecture is preferred

QCC/DETNET FULLY CENTRALIZED ARCHITECTURE

CUC=Centralized User Configuration
- Check traffic requirements
- Get network topology
- Compute configurations

CNC=Centralized Network Configuration
- Push configuration on nodes
- Monitor performance
- Monitor topology



IEEE 802.1TSN dataplane

SOFTWARE-DEFINED NETWORKING FOR TSN NETWORKS CONFIGURATION

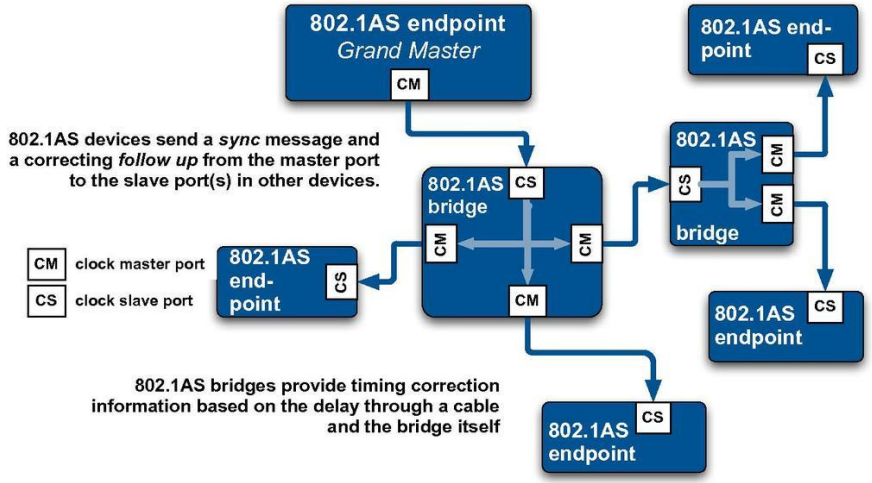
- **SDN maps with the fully centralized IEEE 802.1Qcc model**
  - SDN Controller ↔ Centralized Network Configuration (CNC) entity
    - Support a plurality of network management protocols (SNMP, NETCONF, etc.)
    - Network topology discovery embedded
  - Management the network configuration and forwarding of all nodes
    - VLANs, redirection, queues, shaping, ports configuration, etc.
  - Is pertinent for critical environment
    - Active monitoring to detect deviations and recovery mechanisms to protect “critical” flows

- **Support of SDN “services” to push configuration on network nodes**
  - Processing of traffic characteristics and requirements to configure the network
    → Equivalent to a Centralized User Configuration (CUC) entity
  - Packet interception/injection mechanisms
    - Change protocols behavior without modifying their implementation

OK so lets focus on one standard: Time Synchronization (802.1AS)

IEEE 802.1AS – TIME SYNCHRONIZATION

- **Time Synchronization (IEEE 802.1AS)**
  - Required for Traffic Shaping, Gates scheduling, and application-level clock reconstruction
  - Distribute a clock to all switches and end-points in the domain
    - AS assumes 1 domain; AS-Rev will support more domains
  - Best Master Clock Algorithm (BMCA) to elect the Grand Master
  - Construction of a « clock spanning tree » from a « Grand Master » to slaves
  - Guarantee on end-to-end max latency is dependent of:
    - Number of hops
    - Reliability and accuracy of residence time computation
    - Accuracy of link propagation delay



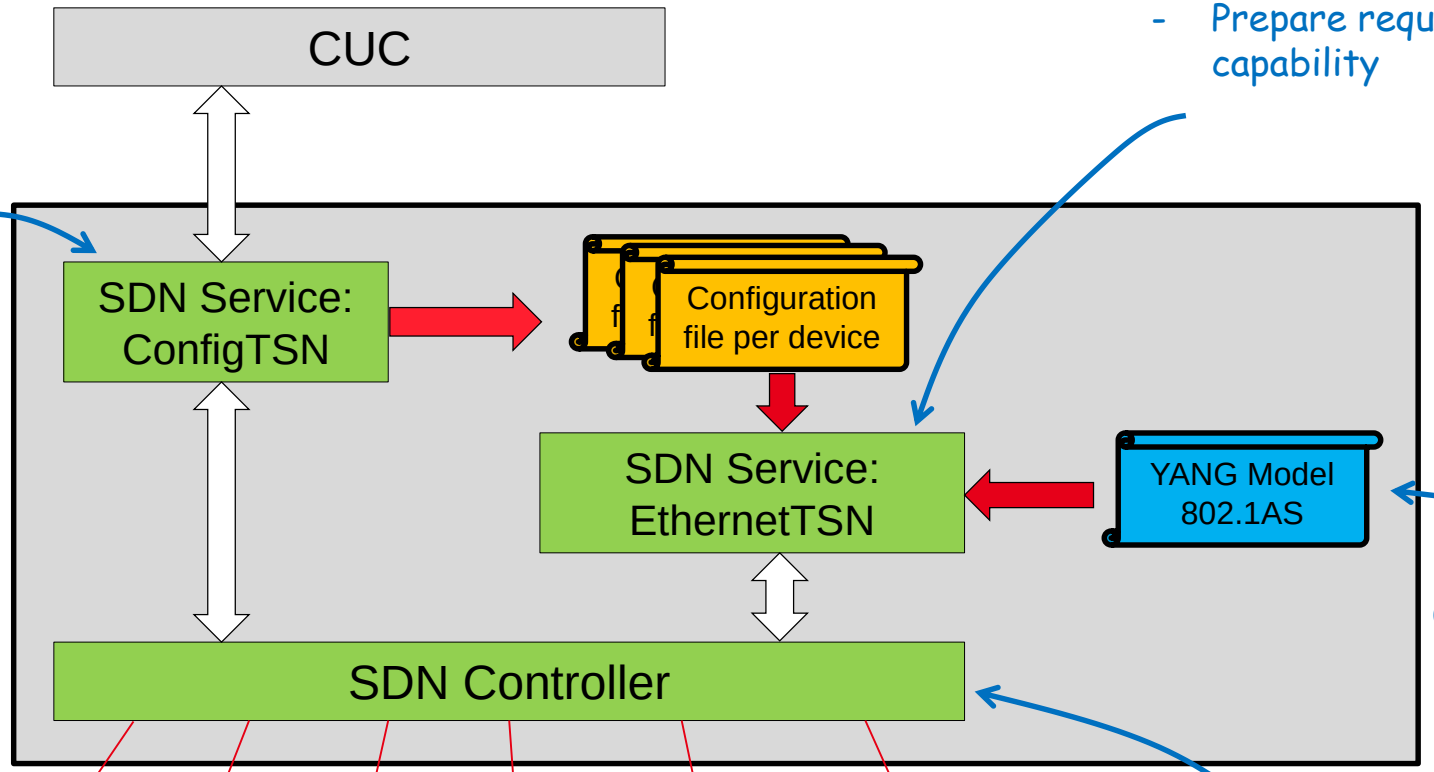
ON THE FIELD CHALLENGES

- **Its not always possible to use BMCA (AVnu)**
  - Convergence time too slow (in the automotive domain)
  - How to manage system reliability and security?
  - Without BMCA ➔ Configure all parameters manually more than 15 parameters per bridge
    - Priority1, Priority2
    - PortRole (Master/Slave)
    - DelayAsymmetry
    - Sync messages frequency shall be the same on all nodes
    - …
- **Adding a new device may increase the number of hops beyond what can be guaranteed**
  - Network engineering to rebalance the tree
  - Change of traffic shaping and scheduling to cope with this new topology

PROPOSED ARCHITECTURE

EthernetTSN:
- Match configuration with YANG model
- Add default values if required
- Prepare request according to device capability

ConfigTSN:
- Filter input from CUC
- Validate configuration
- Fill the blanks
- Generate config per device



draft-benhadjsaid-detnet-gptp-yang-00

Southbound protocols
-NEON
-OpenFlow
-NETCONF
-etc.
→ Specific to each node

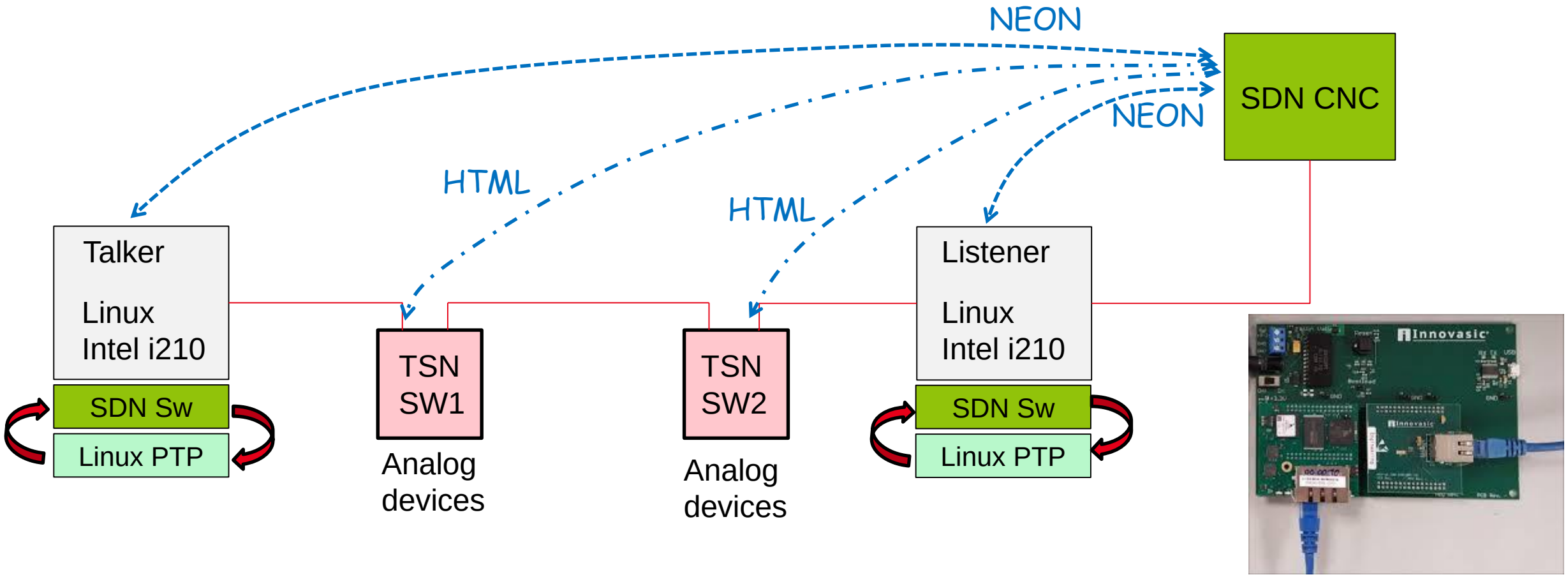
SDN Controller:
- Active monitoring of topology
- Get interfaces configuration and state
- Get performance indicators
- Push request to devices

EXAMPLE

CUC	ConfigTSN	EthernetTSN	SDN Controller

AS Grand Master is Node 1	-Get optimal clock tree
	-Get internal nodes Ids
	
	Per device configuration:
	isGM=1
	Master port=1
	PortRole1=Master
	Priority1=147
	Priority2=250
		-Match configuration file with YANG model
		-Fill the configuration blanks
		
		Per device request:
		{
		Method: TTTechConfigAS
		isGM: 1
		Port1: {
		Role: Master
		DelayAsymmetry: 3ns
		}
		Port2: {
		Role: Master
		DelayAsymmetry:1ns
		}
		Priority1: 147
		Priority2: 250
		InitialLogPdelayReqInterval: 125us
		…
		}
			Send Request over the southbound
			-NETCONF
			-NEON
			-…
			To SDN Sw on devices
			- Interpret parameters to generate plateform dependent configuration

VALIDATION ON TESTBED



SUMMARY AND ON-GOING WORK

- **Why SDN is pertinent to handle and manage TSN mechanisms**
  - Huge number of parameters
  - Reliability of implementations in questions
  - (Temporary) Safety and recovery mechanisms
- **Building blocks to handle Time Synchronization at the CNC level**
  - Set of SDN services to ensure correct configuration
  - YANG Model to be pertinent per device manufacturer

- **Find a most suitable TSN switch**
  - HTML is not efficient/reliable enough
  - Some manufacturers proposed to send configuration by email to generate a new firmware
- **Closing the implementation of ConfigTSN**
  - Generate new configuration if topology change
  - Computation of a new clock tree and synchronized reconfiguration of the infrastructure
  - Find the right data model for communication with the CUC
- **Detect deviations**
  - Safety requires to be able to detect problems

Thank You

Commissariat à l'énergie atomique et aux énergies alternatives
Institut List | CEA SACLAY NANO-INNOV | BAT. 861 – PC142
91191 Gif-sur-Yvette Cedex - FRANCE
www-list.cea.fr

Établissement public à caractère industriel et commercial | RCS Paris B 775 685 019