“Pulse-SDN: Pulse-Coupled Software Defined Networks for Circuit Switched Industrial Control Area Networks”

Reinhard Gentz, Anna Scaglione, Lorenzo Ferrari, and Y.-W. Peter Hong

MOTIVATION

Great potential of Software Defined Control Area Networks (SD-CAN) to better adapt protocols to CAN latency requirements

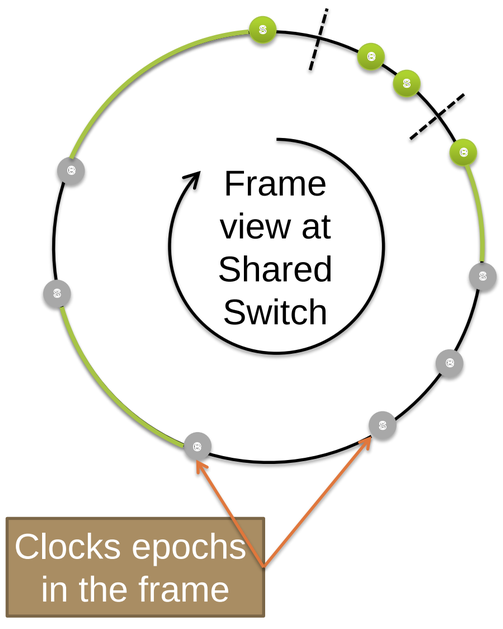
- Circuit switched communications better for CAN than packet switching
- Challenge: complexity of deployment, adaptation, failure recovery

- Idea: Merge in SDN the network and medium access sub-layer solving a multiple access problem over a shared multi-hop path
- The problem is analogous to finding conflict free Time Division Duplex (TDD) schedules in wireless networks

PROPERTIES OF NEW CAN-SDN PROTOCOL

- Minimal end to end latency + no queues at forwarding nodes → instantaneous reception and forwarding of messages
- Fully distributed + simple local computations -> Scalable
- Can co-exist with legacy protocols (with queues) as long as this protocol always has priority (i.e. with IEEE802.1p)
- Traffic flow is controlled by design and is allocated proportionally to demand
- Routes can change dynamically, but not on an active connection
- Resilient to node failures, Self-healing (inspired by biological networks).

BASIC IDEA



- At each node has a “fine” clock measures the time that elapses between one frame transmission and the next
- All nodes schedule is kept via two coarse clock with period = L slots: the “start” slot clock and the “end” driven for each connection
- Each SDN switch has several clocks that mirror the position of all transmitters using a path that crosses that switch

scheduling *synchronization*

$$\Phi_v(t) = s_v(t) + \varphi_v(t) \pmod L$$
$$\Phi'_v(t) = e_v(t) + \varphi_v(t) \pmod L$$

- Each time a clock reaches L it sends a control packet called *beacon*
- Every switch along the path intermediately acknowledges and forwards the beacon to the next switch in the route (like domino effect)
- Both coarse clock and fine clock for communications that are are updated upon receiving a beacon from another communications

Fine Clock update: Every time a beacon is received at time $r_v^{(s)}$ at node v with an estimated travel time of $\hat{t}_{v,c}$, node v updates

$$\hat{\varphi}_c = \varphi_c(r_v^{(s)}) - \frac{\hat{t_{v,c}}}{T}$$

$$\varphi_u(\hat{t}_{c,u}^{(s)+}) = \begin{cases} \varphi_u(\hat{t}_{c,u}^{(s)}), & \Phi_u(\hat{t}_{c,u}^{(s)}) \le \delta_{\text{ref}}, \\ \min\{(1+\alpha)\varphi_u(\hat{t}_{c,u}^{(s)}), 1\}, & \text{else.} \end{cases}$$

The refractory period δ_{ref}, is set based on the noise in Time of arrival of a pulse, i.e. for a 125MHz Ethernet channel and SNR of 30dB is 10ns

Coarse Clock update

- In addition when a node is receiving a start pulse from its successor this node updates its coarse clock:
- Depends on: -The position of the nodes predecessor
  - The nodes demand for communication D_v
  - An transmission guard space δ

$$s_v^{target}(t) = \frac{(D_v+\delta)}{(D_v+2\delta)} e_{pre(v)}(t) + \frac{\delta}{(D_v+2\delta)} s_{suc(v)}(t)$$
$$e_v^{target}(t) = \frac{(D_v+\delta)}{(D_v+2\delta)} s_{suc(v)}(t) + \frac{\delta}{(D_v+2\delta)} e_{pre(v)}(t)$$

SAMPLE NETWORK

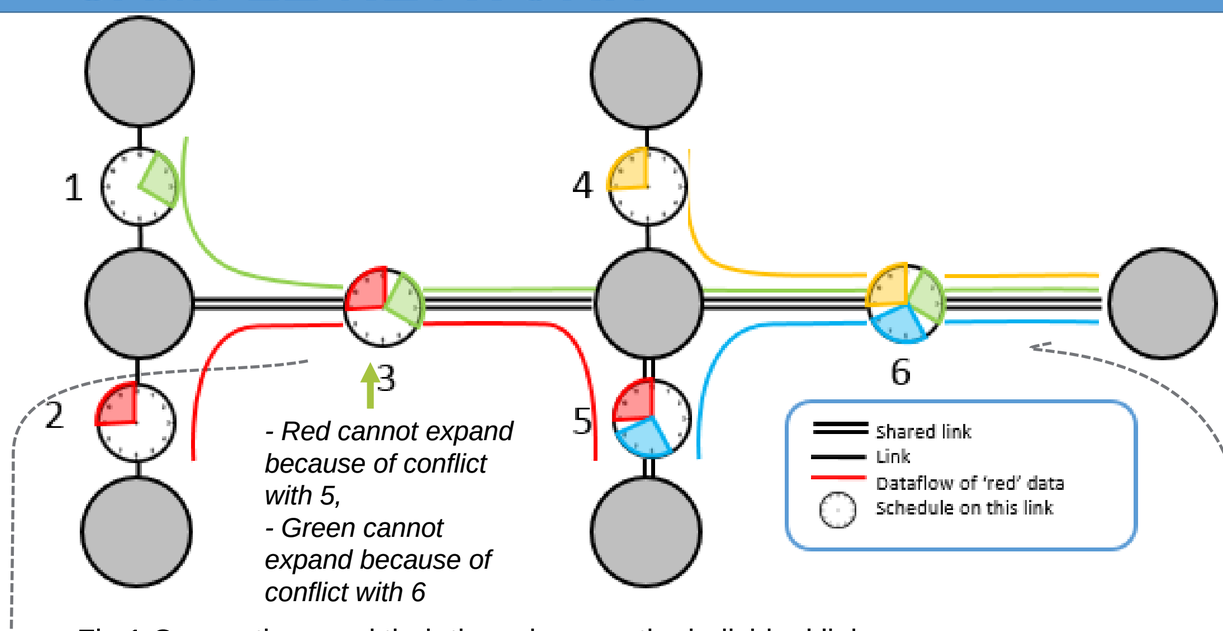


Fig.1 Connections and their time share on the individual links

- Connections are maintained by the data source that is controlling when to send data in the frame and switches mirror source clocks
- Each link is peer to peer, used for a nodes own and forwarded data
- Links that are used my multiple connections create conflicts.
  - our protocol resolves such that there is only one connection active at a time (TDD)
  - A connection always has a two-way path exclusively allocated at a certain interval of the frame → circuit switched communications

SIMULATION RESULTS

- ***Simulation:*** *Topology as in Fig.1, each connection has the same demand 5, delta 1, 120 slots, 20 iterations coupling factor a=0.03, b=0.7*

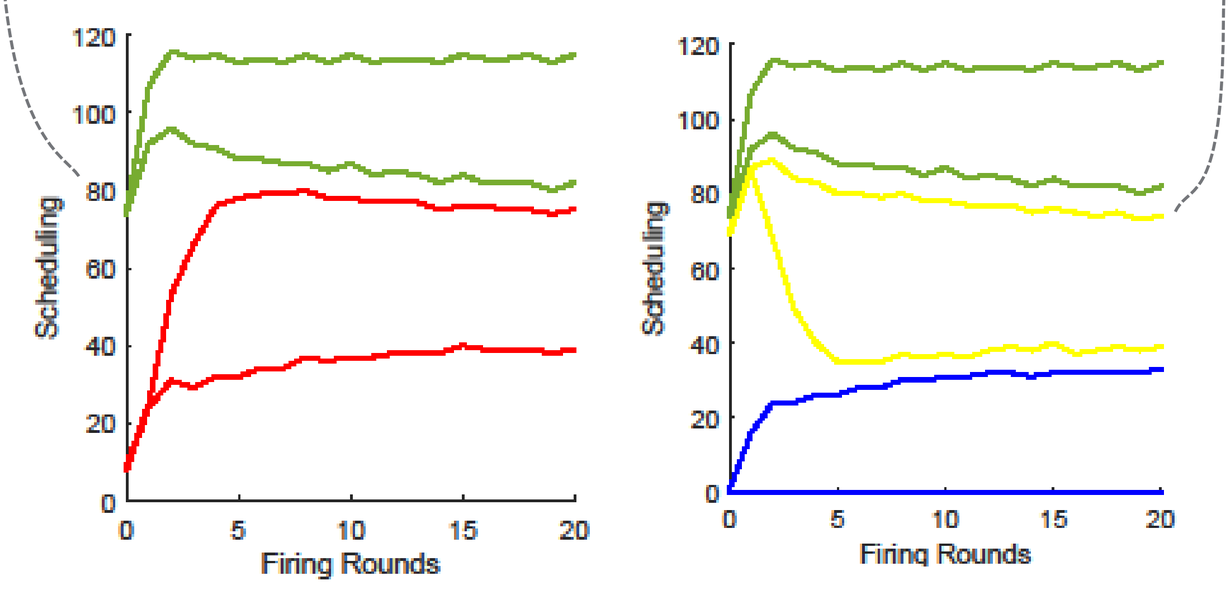


Fig.2 Exemplary scheduling Result of link 3 (left) and link 6 (right) for topology in Fig.1

- Scheduling Results show that the proposed algorithm is converges to the results indicated in Fig. 1.
- We successfully achieve exclusive accesses for each link on each connection
- Each connection gets its fair share of the available communication time per frame (latency = 1 frame)
- The ‘wobbles’ in the graph result due to dithering in the scheduling update that stabilizes the protocol selection of the start and end slots
- Simultaneously all nodes acquire global synchronization (not shown)

BROADER IMPACT/FUTURE EFFORTS

- Impact in all scenarios where latency is important and the traffic patterns are predictable such as industrial control, sensor networks, telephone and video communications.
- Impact at very high speed connections where buffering traffic at each switch is a burden
- Study the protocol on a large network with thousands of connections
- Implement the protocol in an SDN network to prove real world applicability

CYBER RESILIENT ENERGY DELIVERY CONSORTIUM
ASU ARIZONA STATE UNIVERSITY
INFORMATION TRUST INSTITUTE ILLINOIS
SOS SCIENCE OF SECURITY VIRTUAL ORGANIZATION