

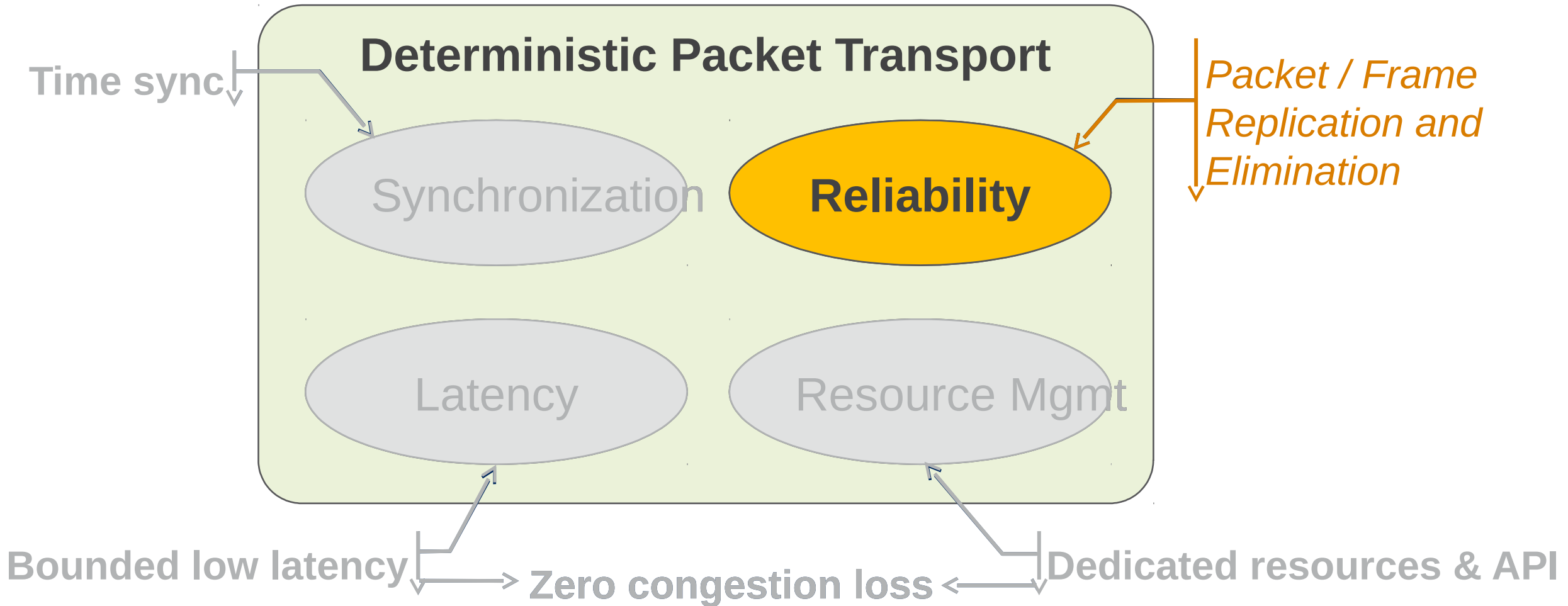
DetNet Data Plane Protection Implementation Report

János Farkas, Balázs Varga, István Moldován

IETF 99

2017-07-20

Reliability on the Spot

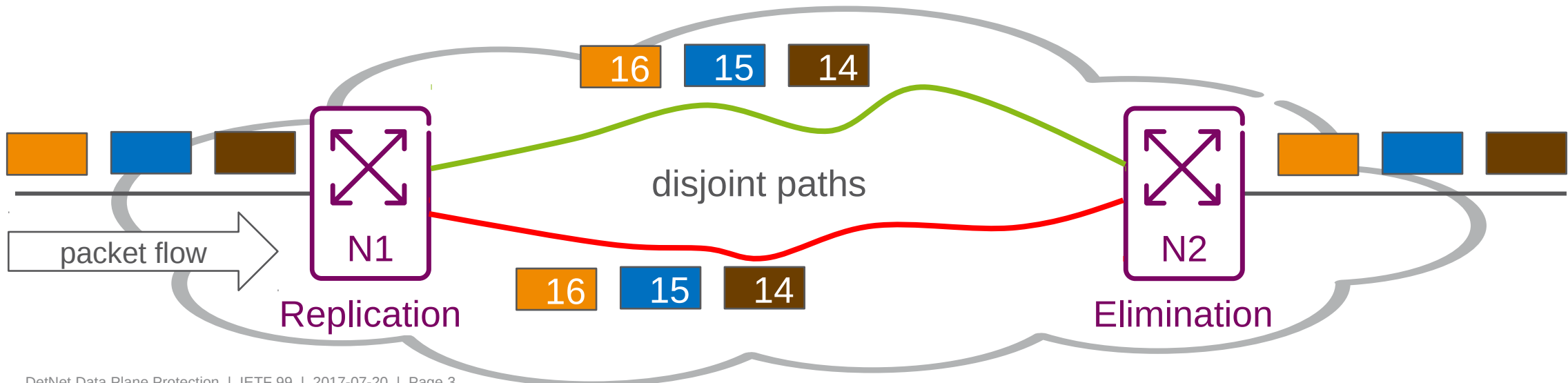


Guaranteed data transport with bounded low latency, low delay variation, and **extremely low loss**



Reliability at Packet Level

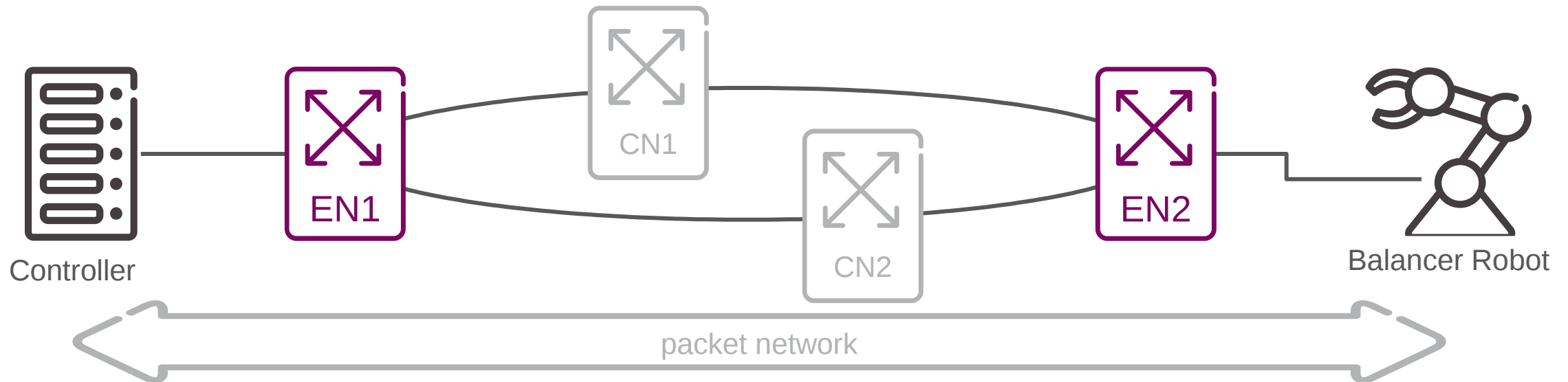
- > **FRER**: Frame Replication and Elimination for Reliability
 - 802.1CB: mechanism, pseudo code, L2 data plane
- > **PREF**: Packet Replication and Elimination Function
 - draft-dt-detnet-dp-sol-01: L3 data plane, i.e., IPv6 and PW over MPLS
- > Per-packet 1+1 (or 1+n) redundancy
- > Send packets on two (or more) disjoint paths, then combine and delete extras



Demo Setup



- › Remote control of a balancing robot
- › Control loop through a packet network
- › PREF / FRER implemented in Ericsson Research software switch on PCs
- › Reference: 50ms protection switching



PW MPLS Data Plane for TSN Service

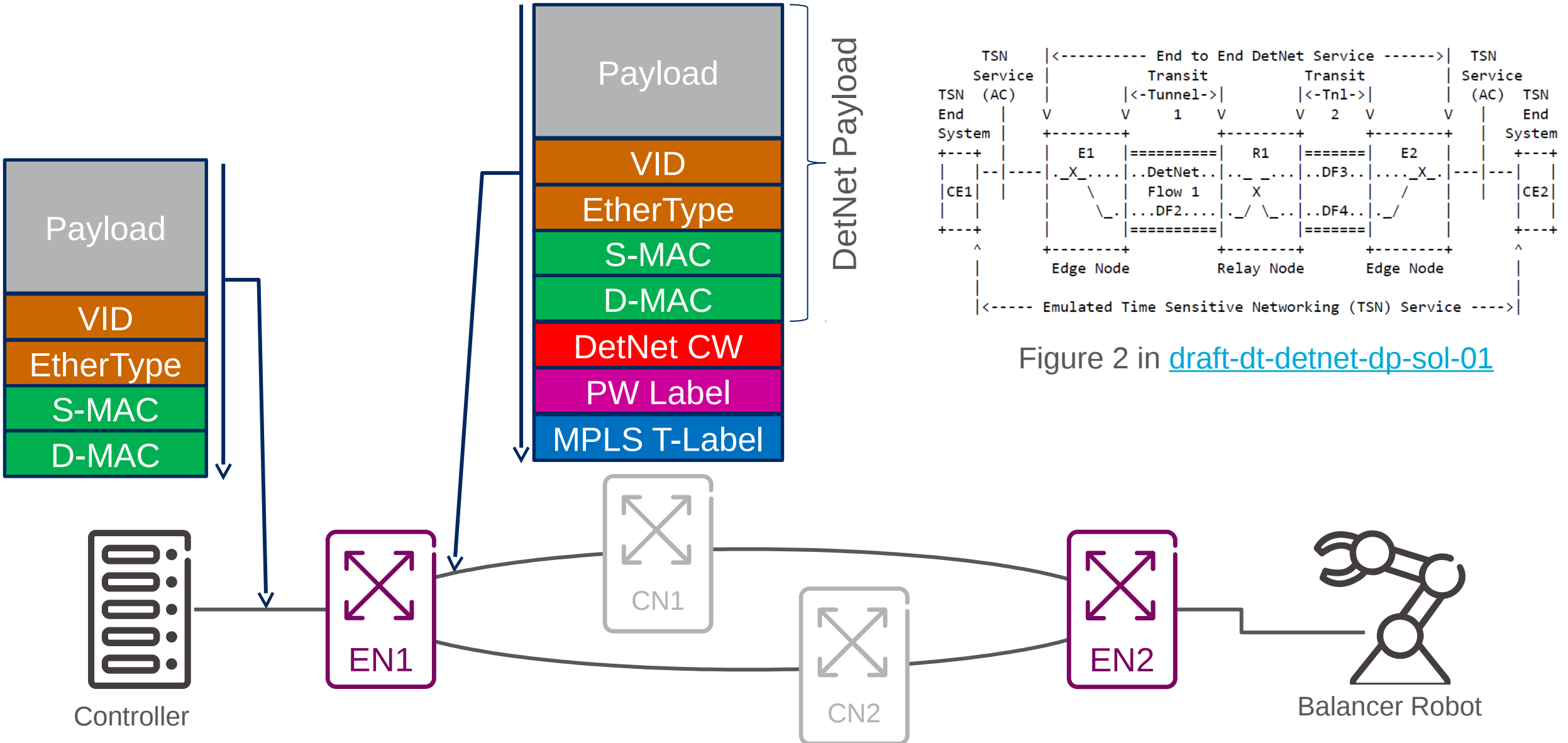


Figure 2 in [draft-dt-detnet-dp-sol-01](#)

PW MPLS Data Plane Example

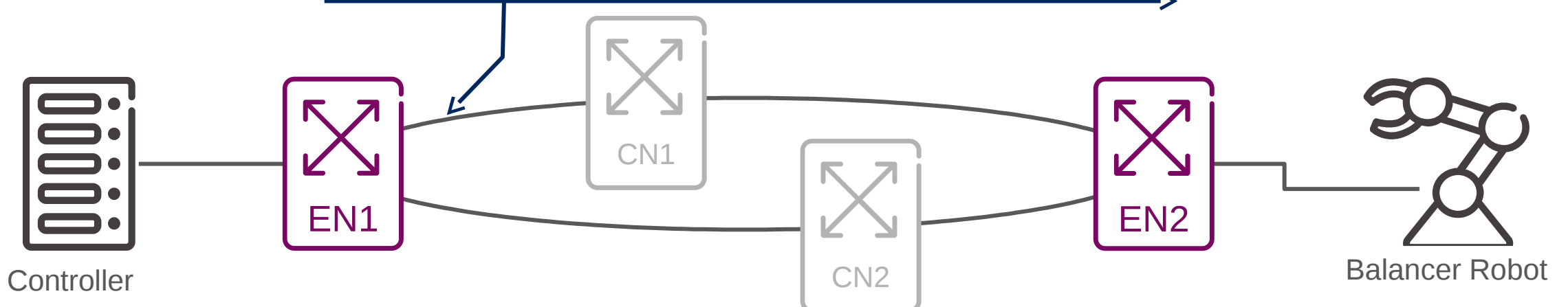


Wireshark:

The screenshot shows a Wireshark capture of network traffic. The main pane displays a list of packets, with packet 23 highlighted. The details pane for packet 23 shows the following structure:

- Frame 23: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
- Ethernet II, Src: IntelCor_bec:c4:9 (00:15:17:be:c4:9), Dst: Syskonne_72:ad:d4 (00:00:5a:72:ad:d4)
- MultiProtocol Label Switching Header, Label: 1001, Exp: 3, S: 0, TTL: 64
 - 0000 0000 0011 1110 1001 = MPLS Label: 1001
 - 0111 = MPLS Experimental Bits: 3
 - 0 = MPLS Bottom Of Label Stack: 0
 - 0100 0000 = MPLS TTL: 64
- MultiProtocol Label Switching Header, Label: 256, Exp: 3, S: 1, TTL: 32
 - 0000 0000 0001 0000 0000 = MPLS Label: 256
 - 0111 = MPLS Experimental Bits: 3
 - 1 = MPLS Bottom Of Label Stack: 1
 - 0010 0000 = MPLS TTL: 32
- PW Ethernet Control Word
 - Sequence Number: 253
- Ethernet II, Src: ae:ab:d0:ad:17:7d (ae:ab:d0:ad:17:7d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 100
- Address Resolution Protocol (request)

The packet bytes pane shows the raw hex and ASCII data for the captured frame.

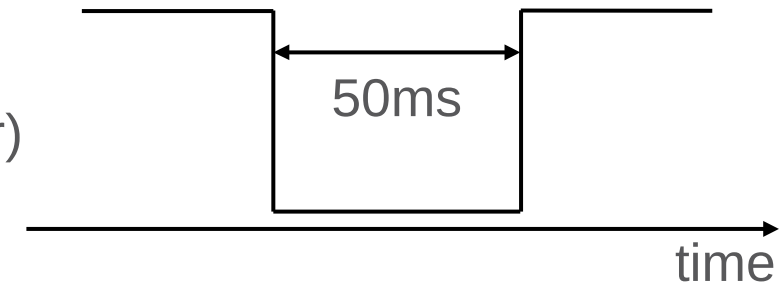


Demo Scenario 1: Link Failure

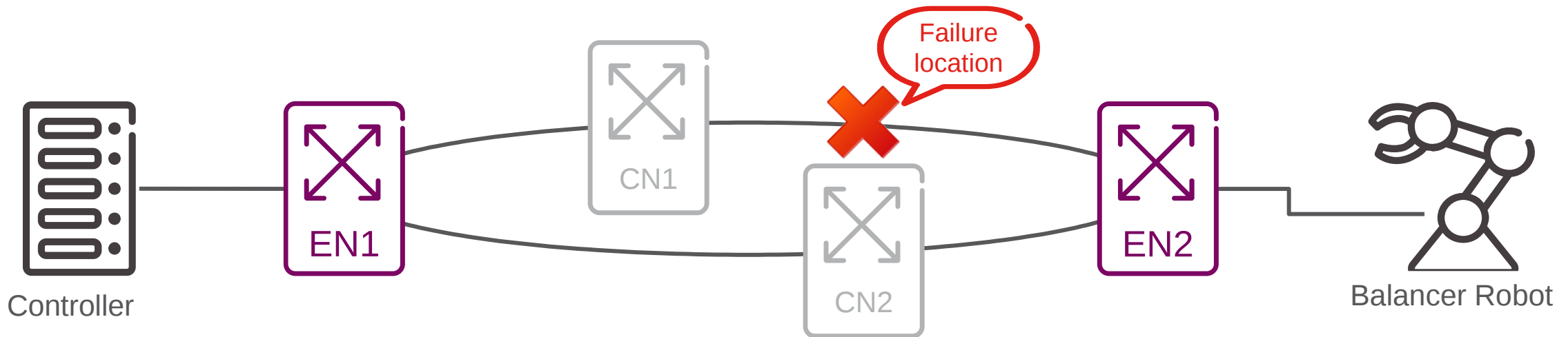


> Protection switching

- Triggered by the failure
- 50ms outage on working path (the protection path is 2ms longer)
- Impacts the application



> PREF / FRER eliminate packet loss caused by outage

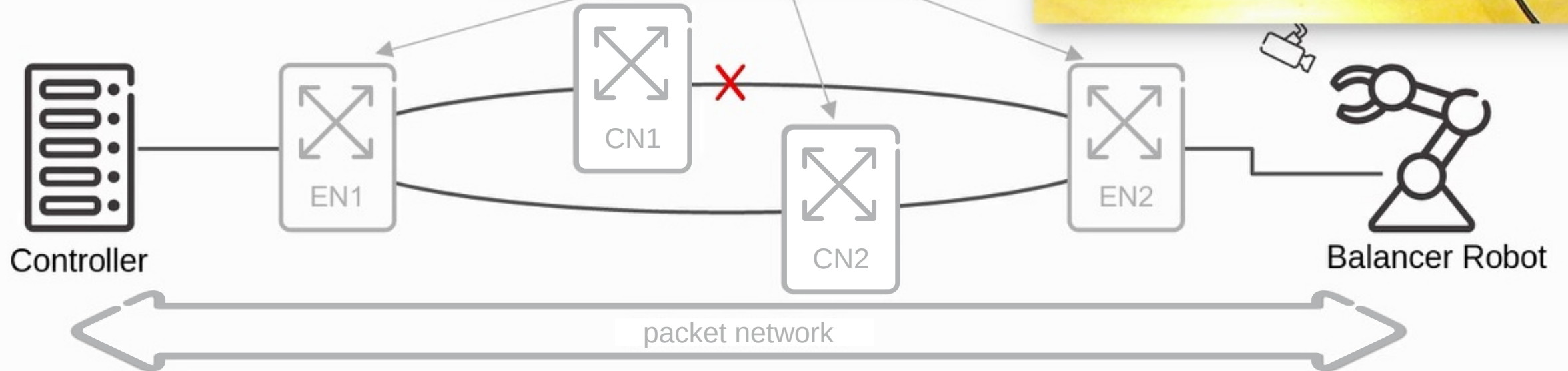
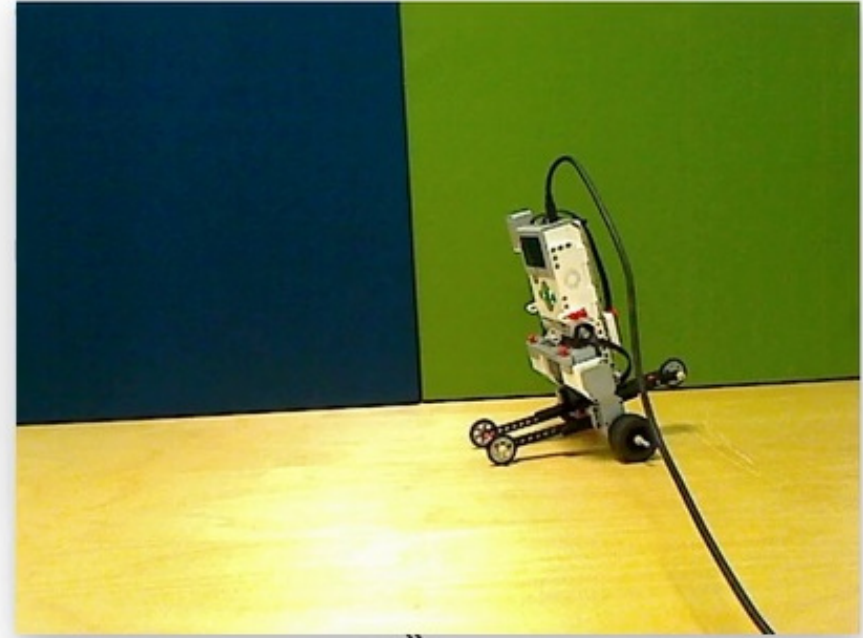


Scenario 1 – Link Failure Protection Switching



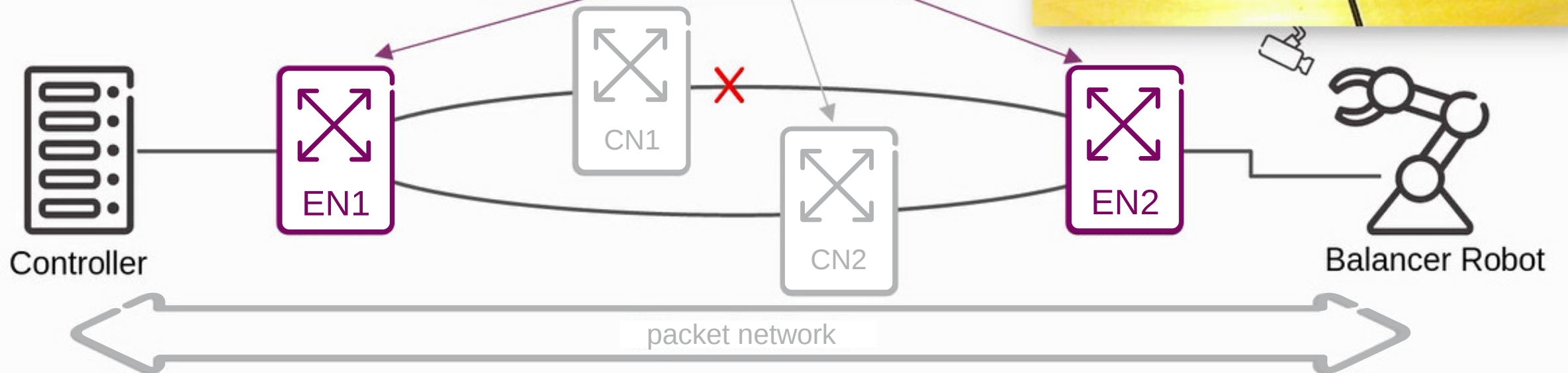
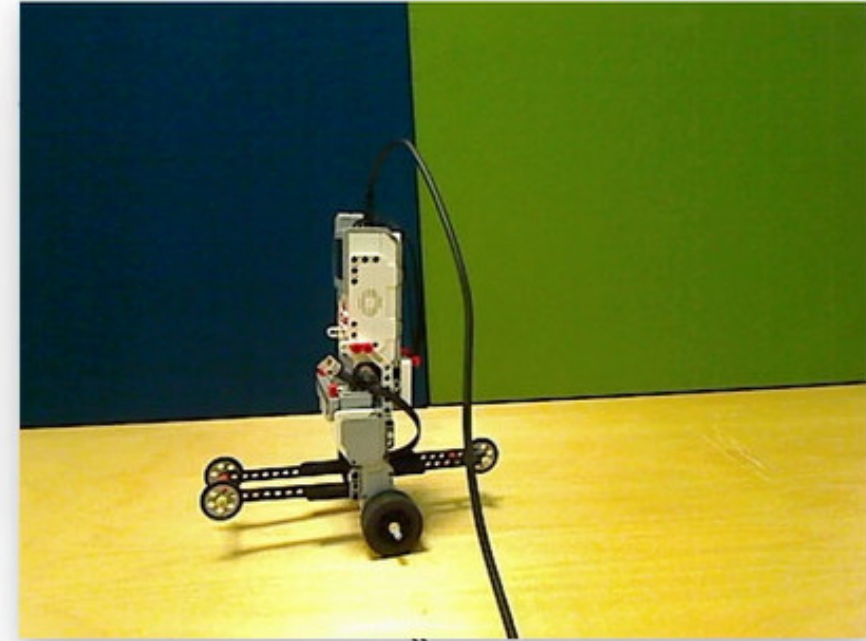
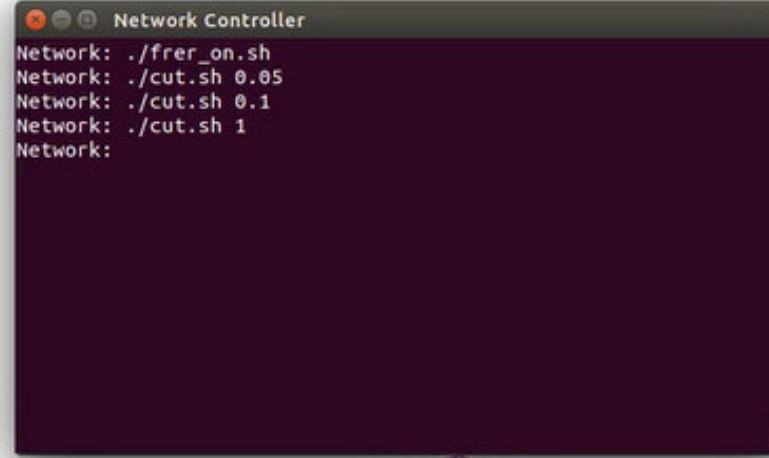
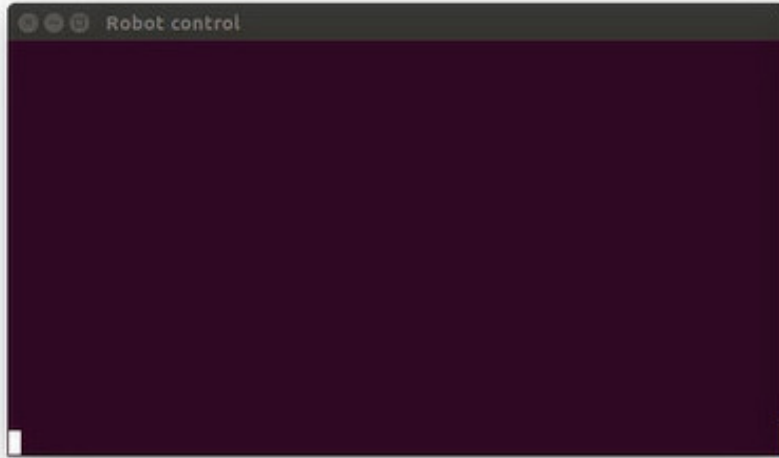
```
Robot control  
  
At 1.26642e+06ms missing 3 messages with seqns from 2104 to 2106
```

```
Network Controller  
Network: ./frer_off.sh  
Network: ./cut.sh 0.05  
Network:
```



Scenario 1 – Link Failure

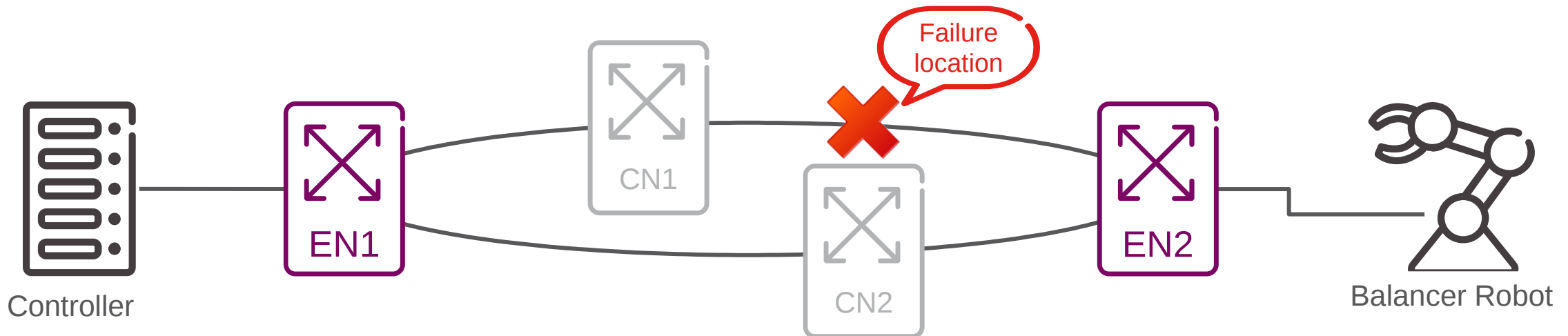
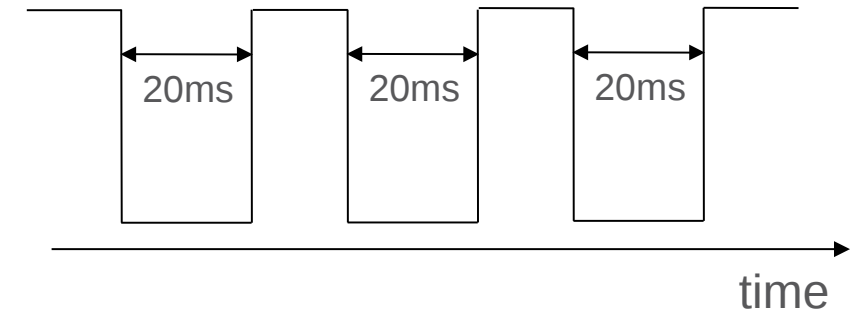
FRER



Demo Scenario 2: Link Flapping



- › Link Flapping
 - Typical L1 problem caused by faulty cable or HW
- › Protection switching does not react
 - Multiple 20ms loss periods impact the application
- › PREF / FRER eliminate packet loss caused by outages

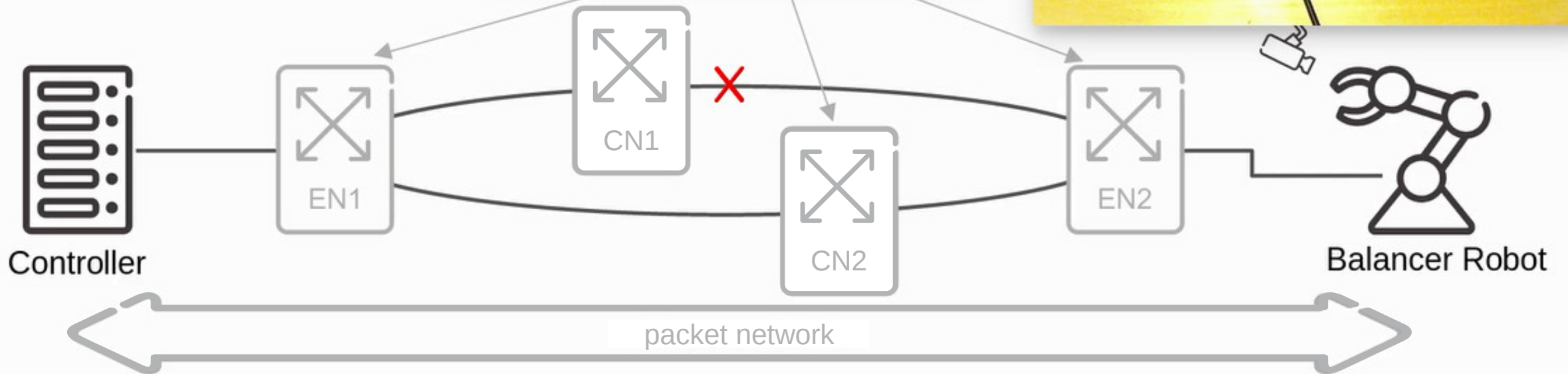
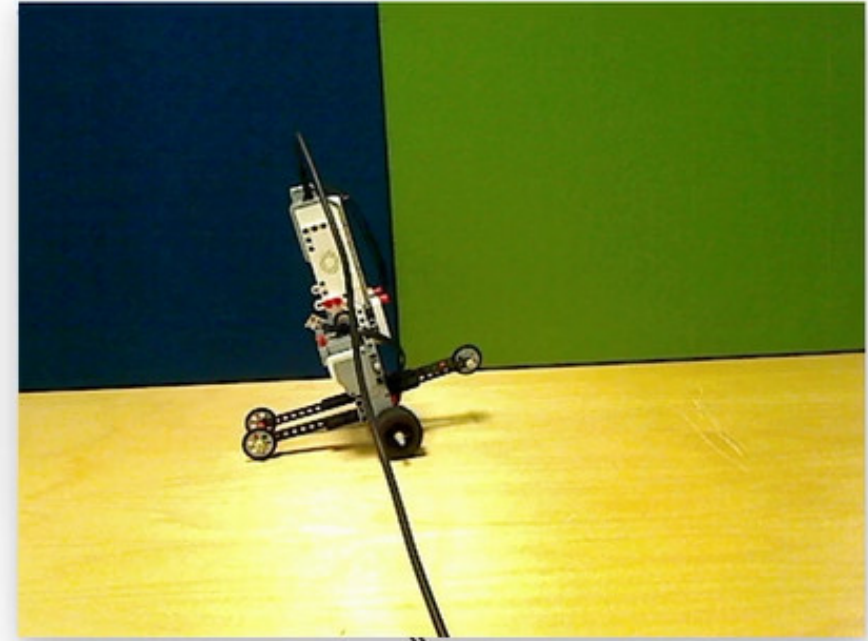


Scenario 2: Link Flapping Protection Switching



```
Robot control
At 515769ms missing 1 messages with seqns from 1756 to 1756
At 515861ms missing 2 messages with seqns from 1758 to 1759
At 515949ms missing 2 messages with seqns from 1761 to 1762
At 516009ms missing 1 messages with seqns from 1764 to 1764
At 516101ms missing 2 messages with seqns from 1766 to 1767
```

```
Network Controller
Network: ./frer_off.sh
Network: ./flapping.sh 0.02 5
Flap 5
Flap 4
Flap 3
Flap 2
Flap 1
Network: █
```



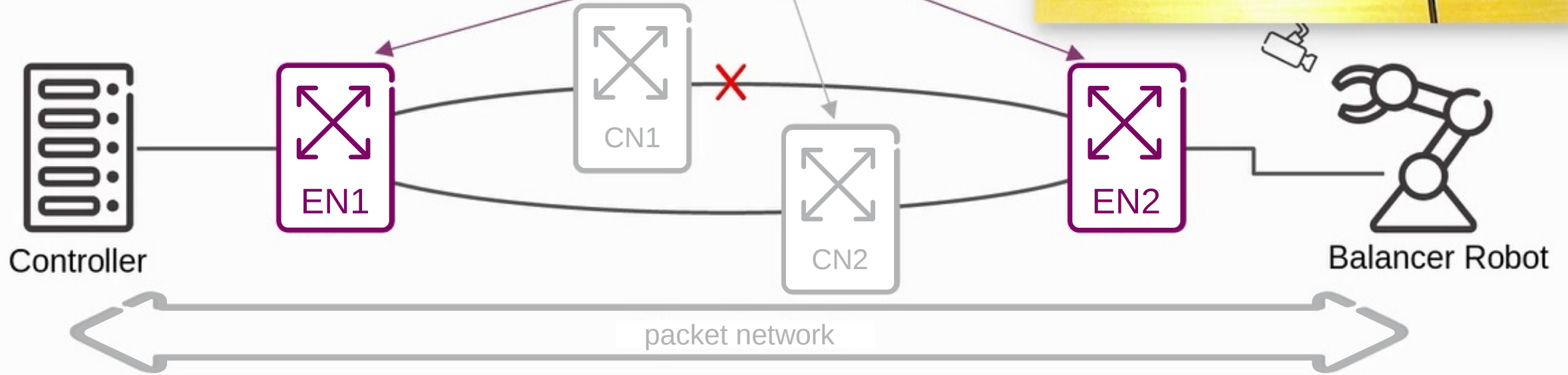
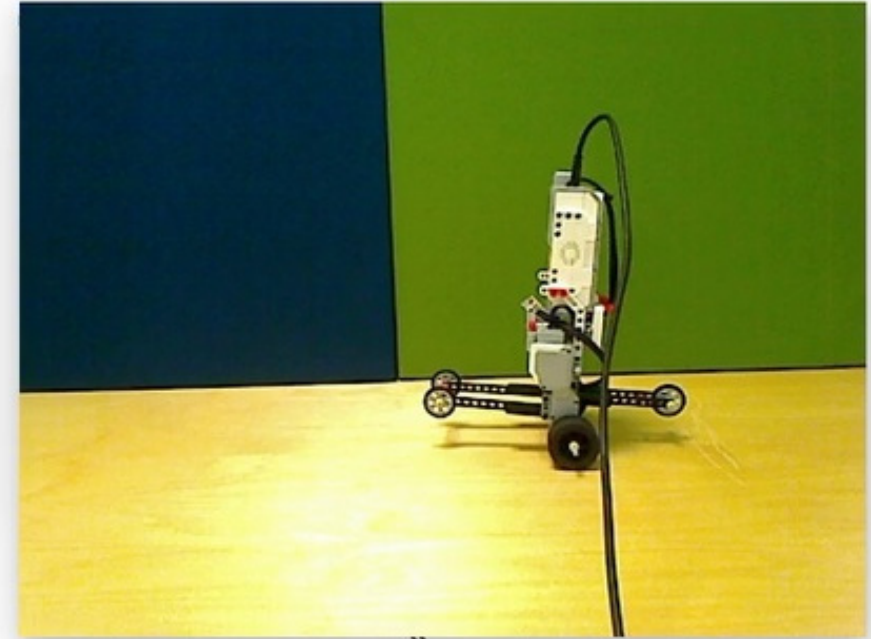
Scenario 2: Link Flapping

FRER



```
Robot control
```

```
Network Controller
Network: ./frer_on.sh
Network: ./flapping.sh 0.02 5
Flap 5
Flap 4
Flap 3
Flap 2
Flap 1
Network: ./flapping.sh 0.05 5
Flap 5
Flap 4
Flap 3
Flap 2
Flap 1
Network: ./flapping.sh 0.1 5
Flap 5
```



Summary



- › PREF provides extremely low packet loss
 - Hitless failover because there is no switchover
 - Protection against failure events that are not even noticed in some environments
- › Implemented the Packet Replication and Elimination Function on PW over MPLS data plane as described in [draft-dt-detnet-dp-sol-01](#)
- › It works fine
- › No caveats found
- › Demo: today during the coffee break starting 15:30 in [Tyrolka, Mezzanine Level](#)



ERICSSON