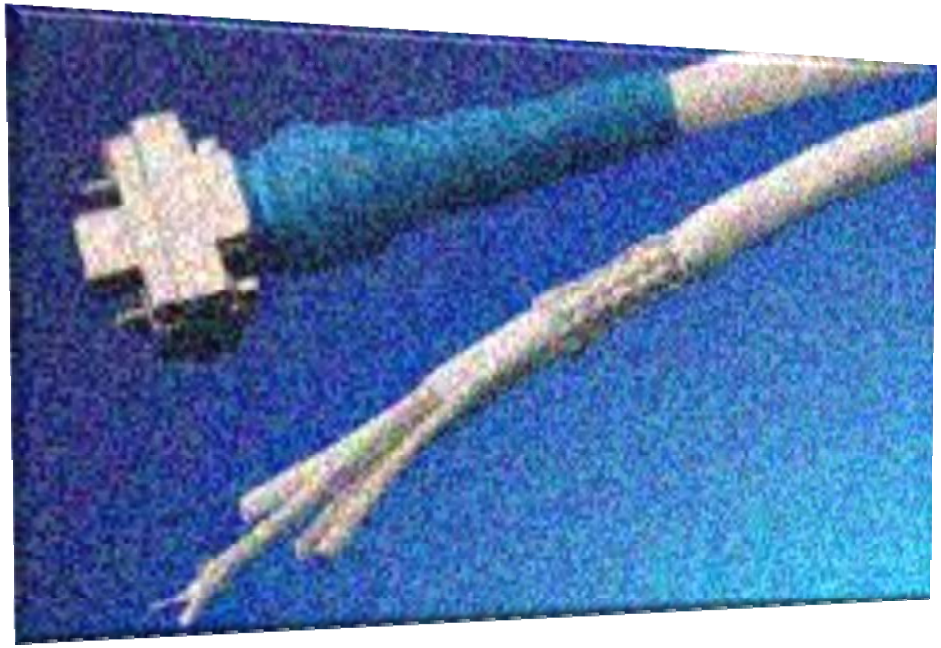


Proposed Technical Solution for Simplex SpW

17th SpaceWire Working Group, 14th December 2011
Noordwijk, Netherlands

Current Status:

- Networking technology for building on-board communications in S/C, used for the interconnection of:
  - Mass-memory
  - OBC
  - Telemetry
  - …
- Designed by ESA and widely used on many ESA, NASA, JAXA, RKA space missions
- The standard specifies point-to-point full duplex links, with flow control mechanism which ensures that no data is lost due to receiver buffer overruns

The Problem:

- Bidirectional flow of data is not always required
- In these cases one direction is only used by the Flow Control mechanism, but:
  - Adds unnecessary mass since half of the wiring is practically unused
  - Adds complexity and cost to the SpW devices

The proposed Solution:

- University of St Petersburg (SUAI) has proposed a solution for a Simplex version of SpW for unidirectional data transfers

Simplex SpW – SUAI Proposal

Simplex Devices:

- Two different ends exist in each link
  - Transmitter, which can only send data and Time-Codes
  - Receiver which can only receive data and Time-Codes

Fields of application:

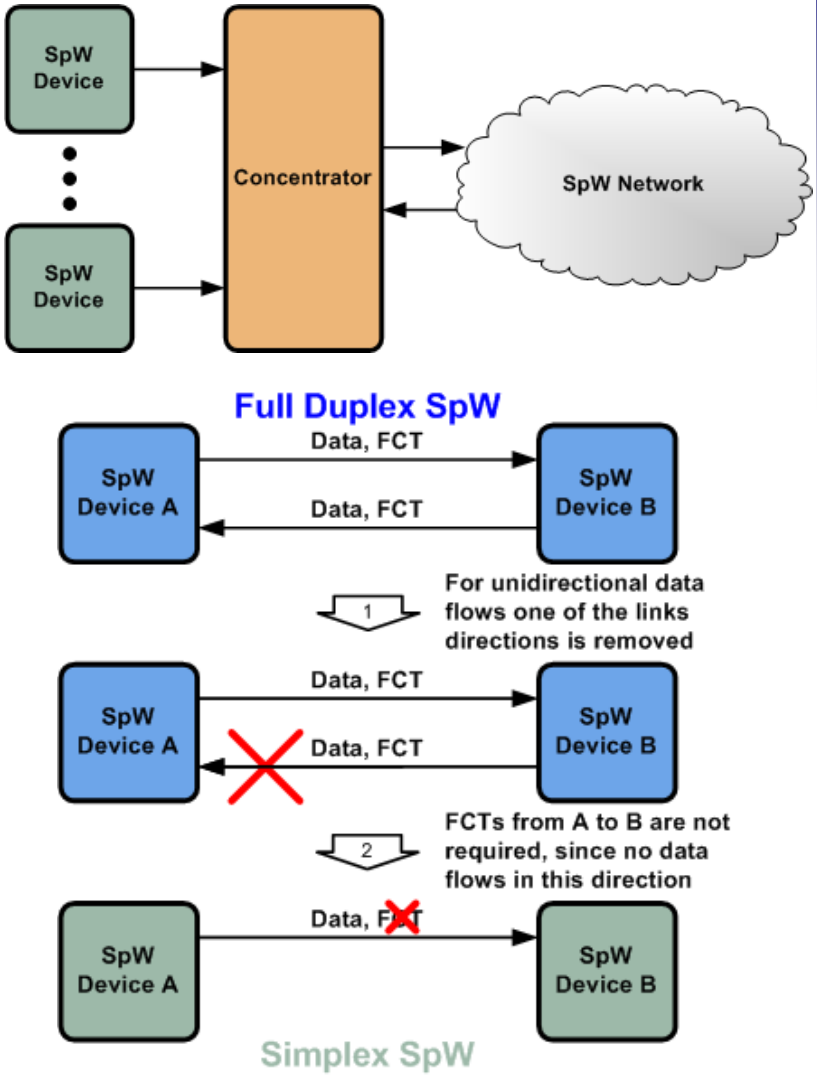
- Connectivity of simple devices
- Potential use in data concentrators

Problems:

- Flow Control is not used at all in Simplex SpW, **but**
  - FCTs are used for Link Initialization; how can the devices know if the link is initialized?
  - FCTs are used to prevent receiver buffer overruns

Proposal:

- Modification of the Link Initialization procedure
- Modification of the data transfer mechanism



Simplex SpW Link Initialization – SUAI Proposal

Link Initialization:

- Tokens flow from the transmitter to the receiver only
  - Only NCHARs, NULLs and Time-Codes are transmitted in the Link
  - The transmitter cannot be aware when a device is connected to the remote end
  - Link initialization can be based on NCHARs, NULLs or Time Codes transmitted from the transmitter to the receiver

☹ **NCHAR based initialization:**

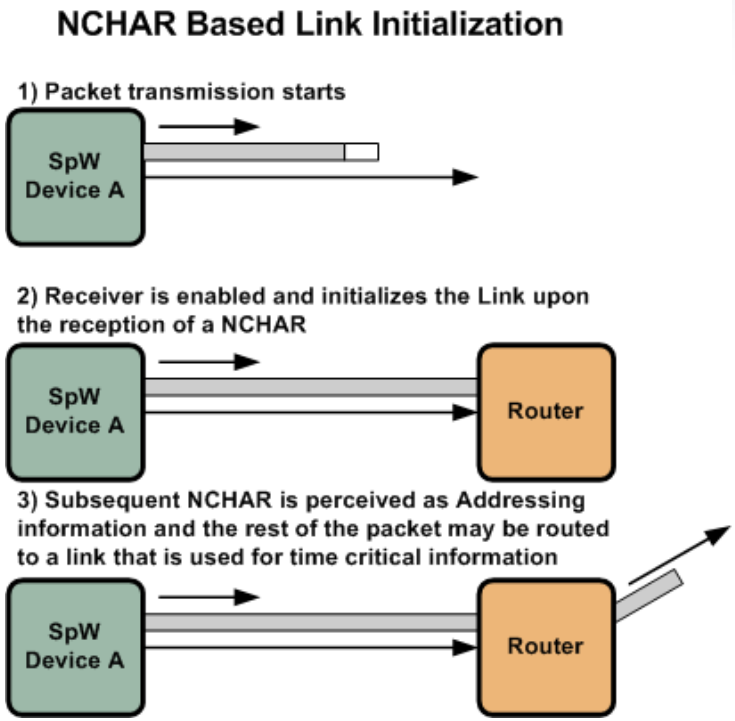
- Receiver Initialization may occur in the middle of a received packet
- The next NCHAR will be perceived as addressing information and propagate to links used for sensitive traffic

😐 **Time-Code based initialization:**

- Simplex will be used for simple devices which are not oriented as Time-Code masters
- Prohibiting Time-Code propagation from Simplex Links, in order to support this method, requires router modifications and adds unnecessary complexity

☺ **NULL based initialization:**

- Does not cause problems to the network operation
- Requires only simple changes at the receiver logic



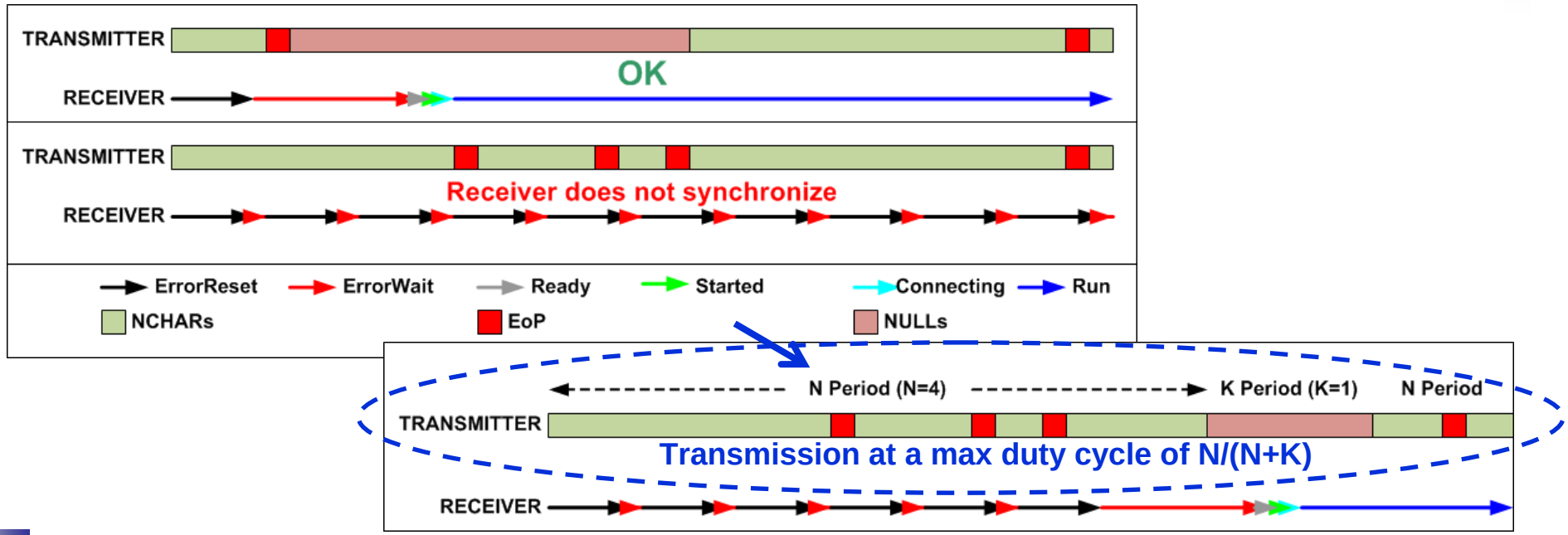
Simplex SpW Data Transfer Mechanism – SUAI Proposal

The Problem:

- The transmitter cannot be aware of the receiver's state
- The receiver may disconnect at some point due to an error
- Transferring back-to-back packets does not allow for NULL insertion and the receiver will not re-syncrhonize

Proposed Solution:

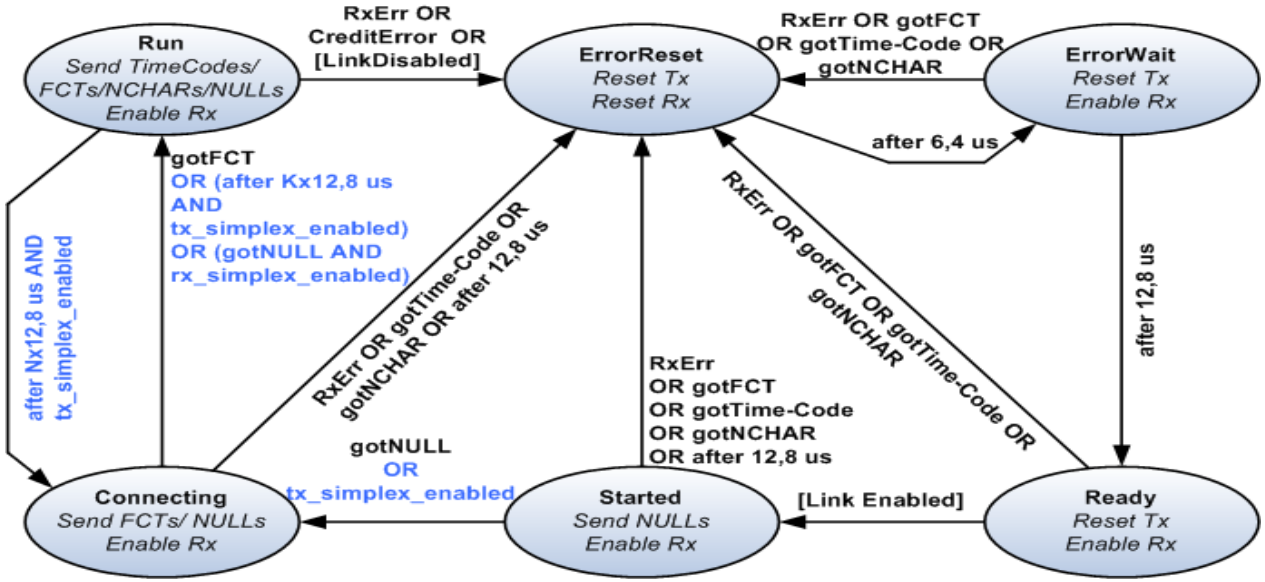
- Transmitter is allowed to transmit for a programmable time period (Nx12,8 us – N Period)
- A period in which only NULLs are transmitted is inserted before the next transmission period, in order to allow the receiver to re-initialize the link after an error (Kx12,8 us – K Period)



Simplex SpW State Machine – SUAI Proposal

Four conditions have been added to the SpW State Machine:

- **Simplex SpW Transmitter:**
  - Transition to from the ***Started*** state to the ***Connecting*** State is unconditional
  - Transition to from ***Connecting*** state to the ***Run*** State is performed after the expiration of the Kx12,8 us period
  - Transition to from ***Run*** state to the ***Connecting*** State is performed after the expiration of the Nx12,8 us period
- **Simplex SpW Receiver:**
  - Transition to from ***Connecting*** state to the ***Run*** State is performed if the ***GotNULL*** signal is asserted



Simplex SpW Data Transfer – SUAI Proposal

The Problem:

- The transmitter cannot be aware of the receiver's state
- The receiver cannot be aware of the transmitter's state
- A receiver may disconnect due to an error while the transmitter is still transferring a packet
- Inserting NULLs between NCHARs within a packet may lead to the same situation as "NCHAR based initialization" if the receiver is in the stage of initialization

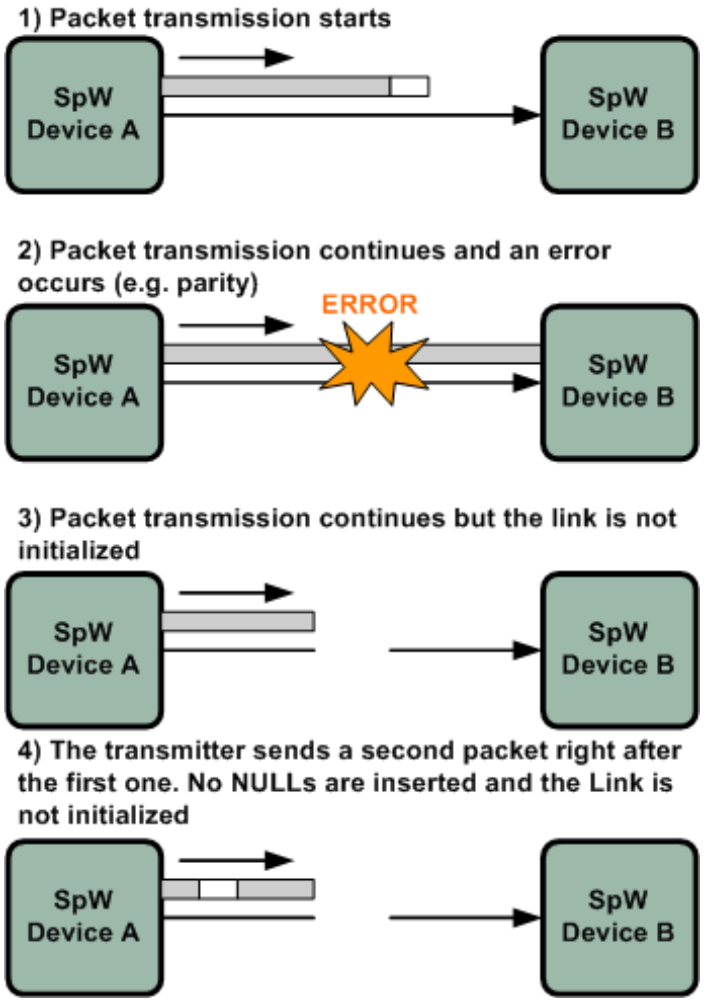
⇨**A mechanism which will allow the receiver to identify start of packets is required**

Proposed modification to the SUAI Proposal:

⇨**NULLs shall not be inserted within a packet (limits max. packet size) or**

⇨**EEP insertion/packet spilling shall be employed at the transmitter if the Tx buffer empties, or**

⇨**<u>FCT based initialization shall be employed</u>**



Simplex Receiver – Transmitter Synchronization

- In SUAI proposal the Simplex SpW state machine has minor changes compared to the Full Duplex SpW state machine
- In Full Duplex SpaceWire a Controller which receives a non-expected character (e.g. NCHAR at the Connecting state) can inform the remote side by causing a disconnect (silence exchange)
- The reason for this is for both ends to pass from the same states at the same time

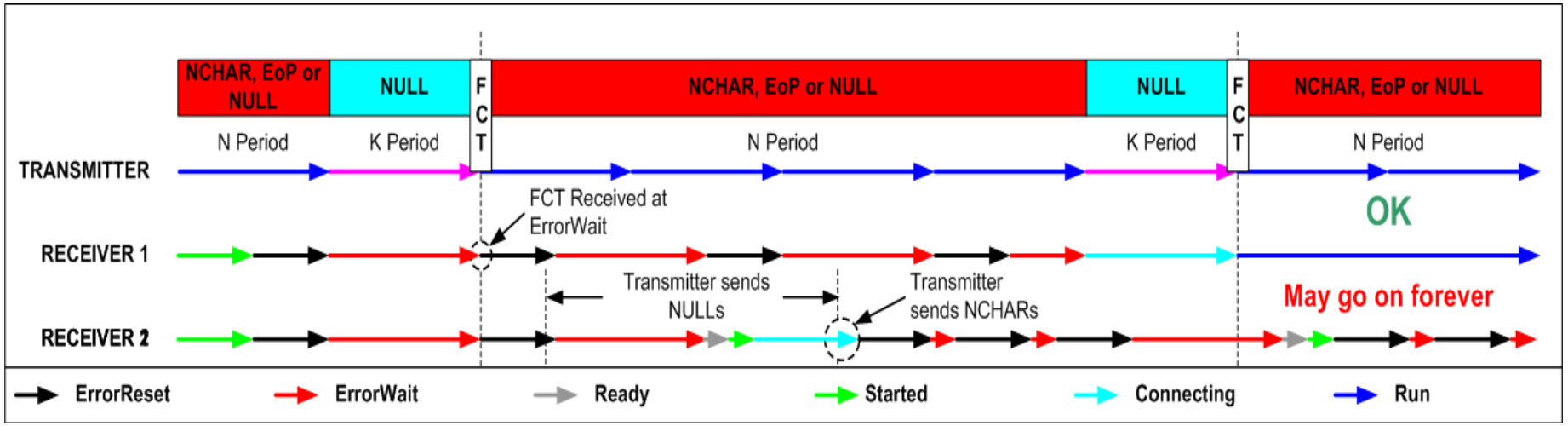
☹ In Simplex the Receiver cannot cause a Disconnect to the transmitter in order to re-synchronize it

☹ The Simplex Receiver may be at any state when the transmitter is sending NCHARs or FCTs

☹ This causes non-deterministic start-up time, but

☺ Only the receiver needs to be synchronized to the transmitter and not vice versa

⇨ **Remove the conditions that enable a Full Duplex Controller to cause disconnect to the remote end**



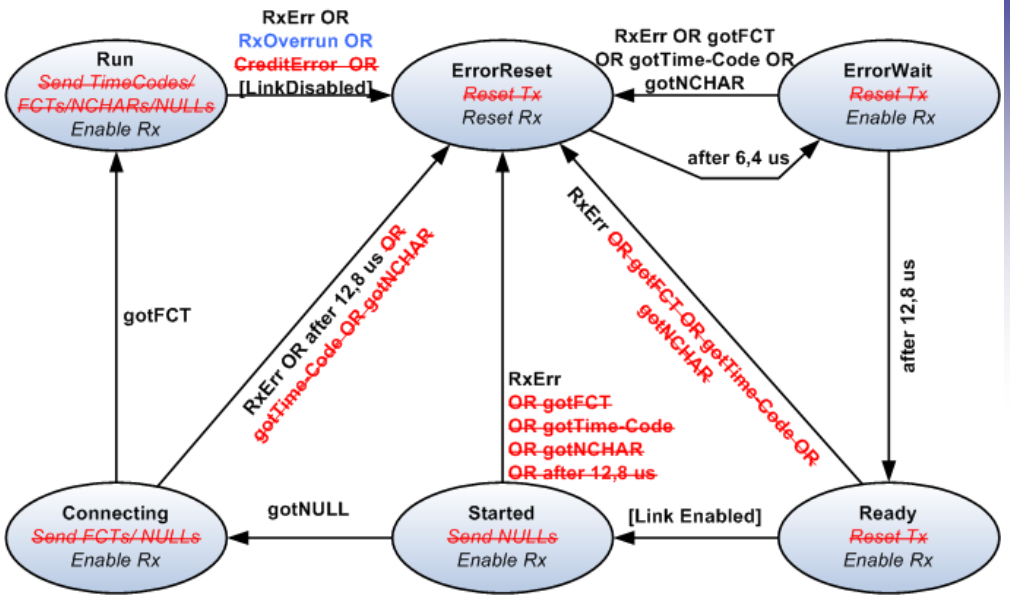
Simplex SpW Receiver functionality – FCT based Initialization

Functional differences:

- The main difference is the absence of the transmit part
- The Simplex SpW state machine logic is much simpler than in Full Duplex SpW

SpW Block Modifications:

- Transmitter part shall be removed
- Link Initialization state machine shall be modified
- Flow Control Logic shall be modified



Link Initialization State Machine Modifications:

- The State Machine shall transition to the “ErrorReset” state in case a overrun occurs at the receive buffer since CreditError cannot be defined for Simplex SpW

Flow Control Block modifications:

- Shall be modified to cause a Buffer Overrun signal based on the number of NCHARs received and the Rx buffer capacity since no FCTs are transmitted

Interface modifications:

- A signal enabling Simplex Mode shall be added for blocks supporting both Full Duplex and Simplex

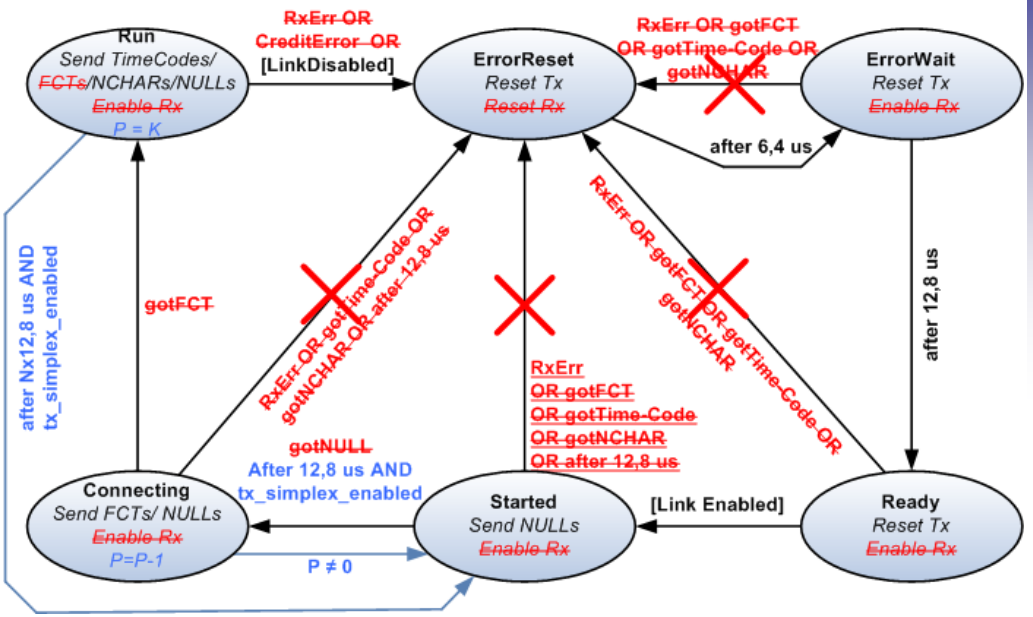
Simplex SpW Transmitter functionality – FCT based Initialization

Functional differences:

- The transmitter is allowed to transmit NCHARs for a period of Nx12,8 us
- The transmitter shall transmit NULLs only for a period of Kx12,8 us and FCTs every 12,8 us

SpW Block Modifications:

- Receiver part shall be removed
- Link Initialization state machine shall be modified
- Transmitter Flow Control Block shall be removed
- An entity implementing the K, N timers shall be added



Link Initialization State Machine Modifications:

- Transitions to ***ErrorReset*** are allowed only from the ***Run*** state
- Transition from the ***Started*** to the ***Connecting*** state occurs after the period of 12,8 us has elapsed
- Transition from the ***Connecting*** to the ***Run*** state occurs after the transmission of K FCTs at a period of 12,8 us
- At the expiration of the Nx12,8 us transmit window the transmitter shall return from the ***Run*** to the ***Started*** state to allow the receiver re-initialize in case of error

Interface modifications:

- A signal enabling Simplex Mode shall be added for blocks supporting both Half Duplex and Simplex

FCT Based Initialization – Pros and Cons

Problem:

- If a packet starts being transmitted late in the N period it will need the next N period(s):
  - A receiver may reconnect during the K periods when the FCTs are transmitted
  - The first NCHAR received during the next N period will be perceived as routing information
  - ⇨ **Same situation as NCHAR based initialization**

Solutions:

- Do not transmit FCTs during the K period if the next NCHAR to be transmitted is not the first NCHAR of a SpW Packet
  - ☑ Decreased latency, packets can be sent at any time
  - ☑ Arbitrarily large packet support
  - ☒ Most likely to lose packets due to receiver buffer overrun
  - ☒ Does not allow Babbling idiot identification by the receiver
  - ☒ Difficult to implement if the SpW Controller's Tx buffer already contains segments from more than one packets
- A SpW Packet shall be transmitted in a single N period (impose maximum packet size)
  - ☒ Max packet size to be transmitted depends on N and on the Link Rate
  - ☒ Late transmission within the N window shall not be allowed or the N period shall be extended – increases complexity
  - ☒ Applications wishing to transmit late on the N period shall wait until the next N period - Increased latency
  - ☑ Less likely to have receiver overruns in heavily loaded networks
  - ☑ Can support identification of Babbling idiots by the receiver

Simplex SpW Performance – SUAI Proposal

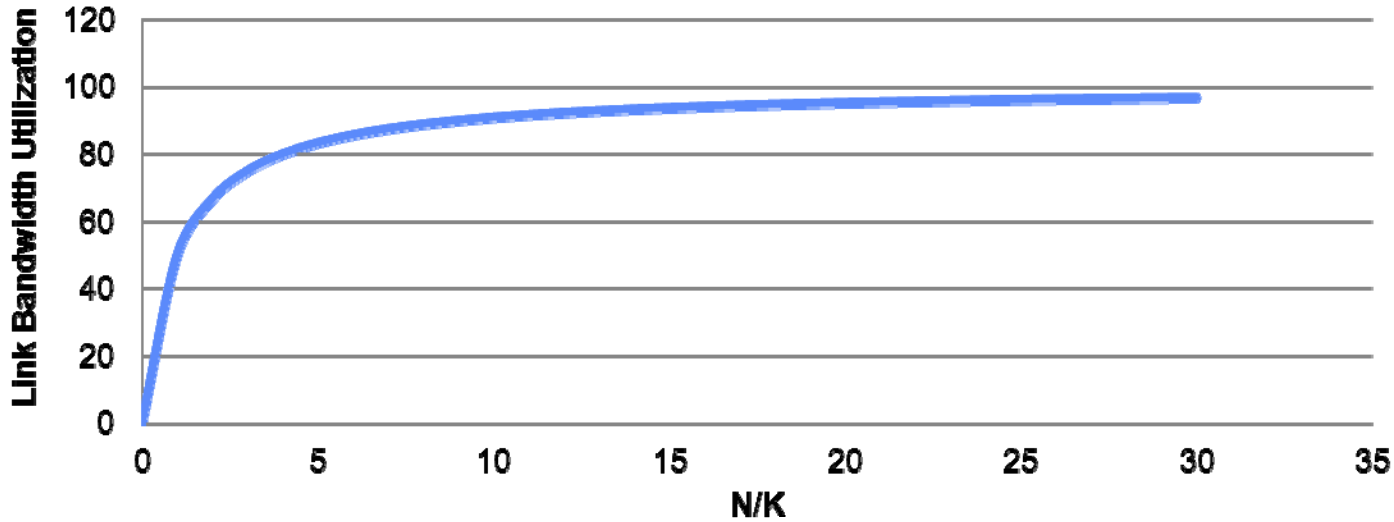
Traffic capacity and potential data loss are opposing driving factors regarding optimization

- For N/K>10 maximum throughput is near the maximum Link Throughput
- Larger transmission windows may result in higher data loss
- Smaller transmission windows result in low bandwidth utilization

⇨**Parameters N, K shall be set per application**

N/K	Simplex SpW throughput
1	50%
5	83,3%
10	90,91%
15	93,75%
20	95,23%
25	96,15%
30	96,77%

Near-full Link throughput for N/K>15



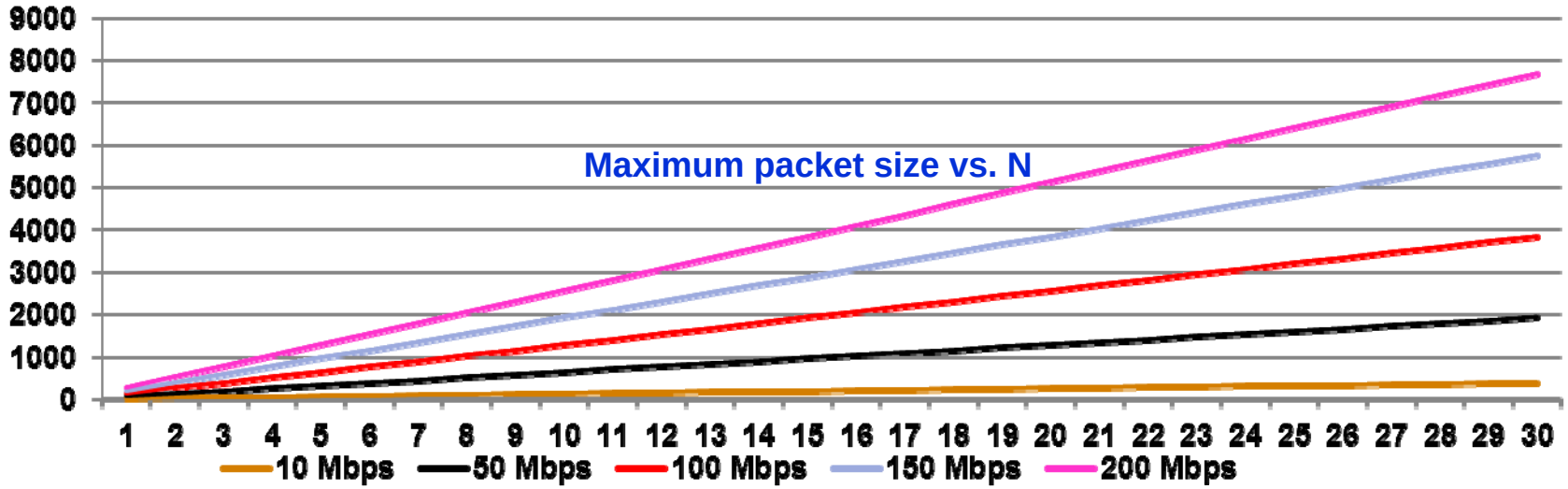
Simplex SpW Performance – FCT Based Initialization (1/2)

Maximum Packet Size as a function of N and Link Speed					
N	**10 Mbps**	**50 Mbps**	**100 Mbps**	**150 Mbps**	**200 Mbps**
1	12 Bytes	64 Bytes	128 Bytes	192 Bytes	256 Bytes
10	128 Bytes	640 Bytes	1280 Bytes	1920 Bytes	2560 Bytes
20	256 Bytes	1280 Bytes	2560 Bytes	3840 Bytes	5120 Bytes
30	384 Bytes	1920 Bytes	3840 Bytes	5760 Bytes	7680 Bytes

BW utilization is good for N > 15, K = 1

⇨ **Resulting Packet size is ok for simple applications even for the initialization Link Rate**

⇨ **For larger packets there is an impact on BW utilization due to segmentation**



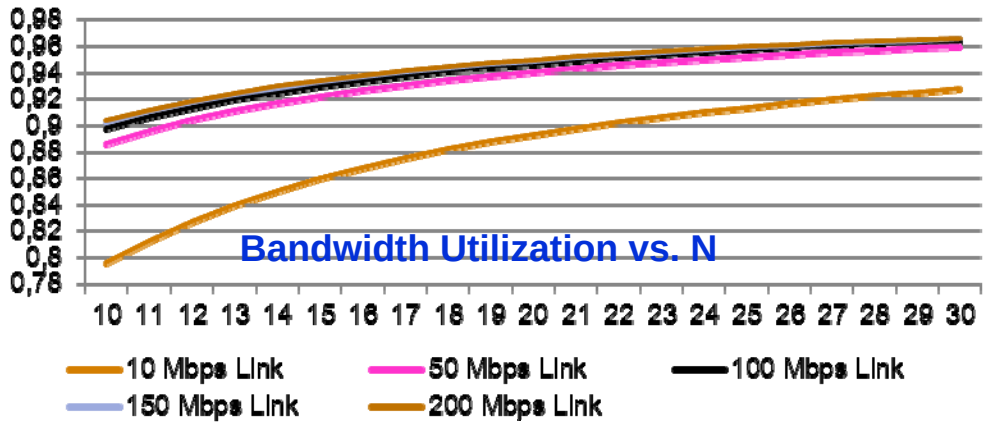
Simplex SpW Performance – FCT Based Initialization (2/2)

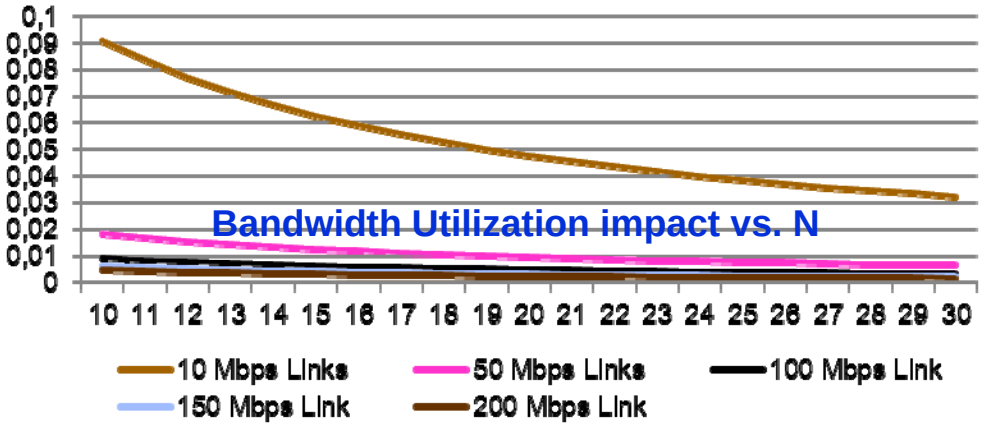
- Application Packets segmented to more than 1 SpW packets if they do not fit within a single N period
- Segmentation introduces overhead (Routing Information, Packet Length, Sequence Number, Control - CRC)
- The figures shows the impact with 16 bytes protocol overhead (e.g. 12 bytes path address, 2 data length, 1 Sequence Number, 1 Control)
- For N=10 the Bandwidth utilization decreases from 90.9% to 79.5% for a 10 Mbps Link
- For N=10 the Bandwidth utilization decreases from 90.9% to 90.34% for a 200 Mbps Link
- For N=30 the Bandwidth utilization decreases from 96.8% to 96.6% for a 200 Mbps Link

⇨ **Increasing N, increases the maximum SpW packet size which fits in a single N period and mitigates the impact of the protocol overhead**

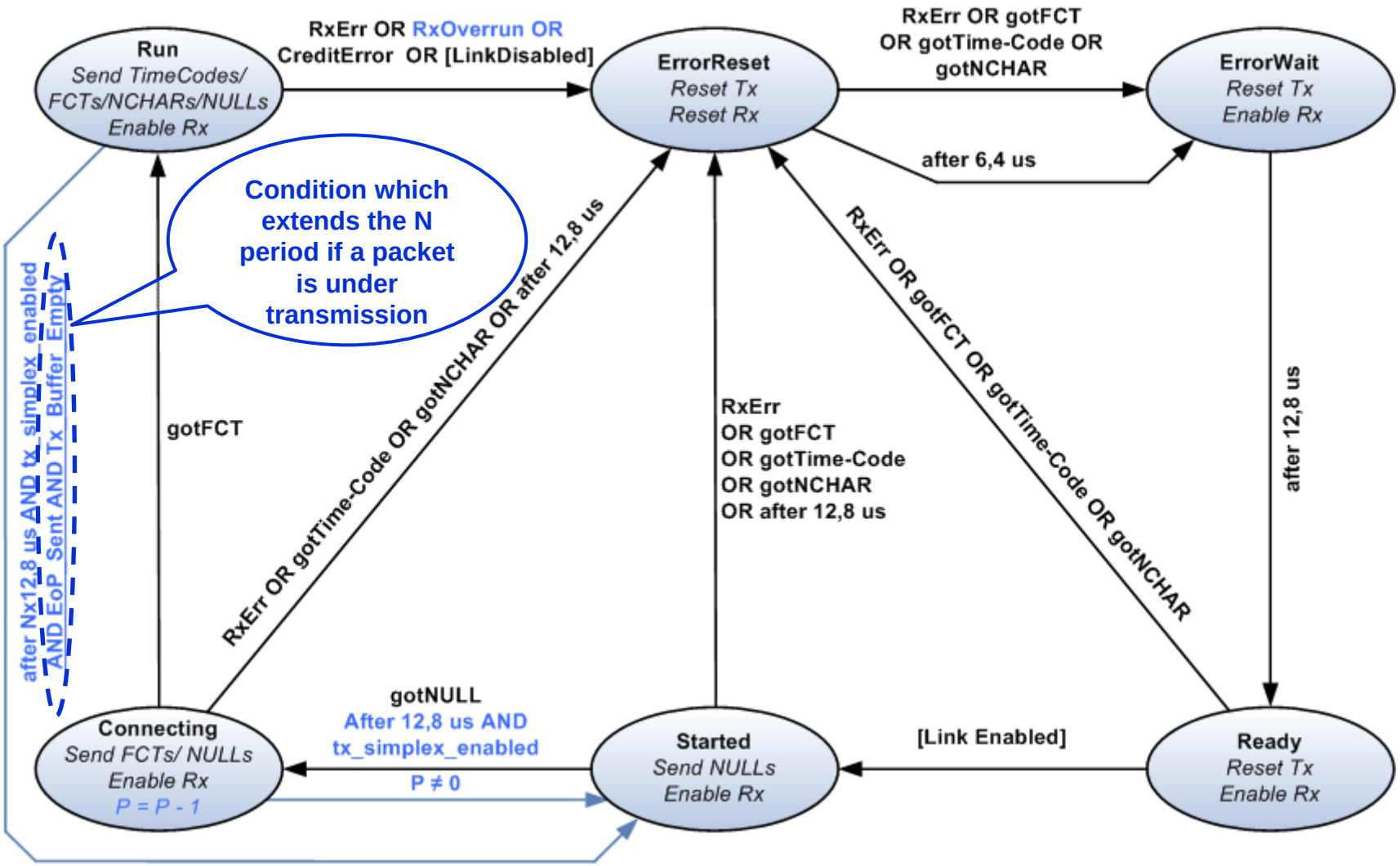
⇨ **Increasing the Link Speed has the same effect**

⇨ **Segmentation impact is not significant**





Overall SpW State Machine - proposal



Simplex SpW Issues

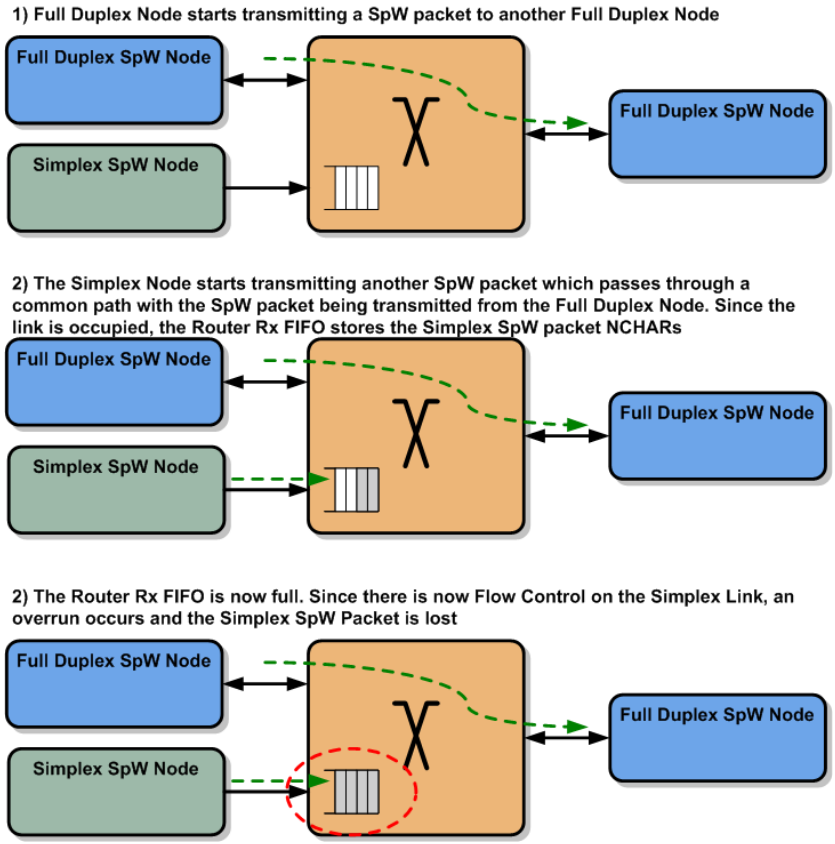
- Assume two Nodes, a Simplex and a Full Duplex who are sending data over a common path
- If the path is free, the entire path “sinks” the data and the Simplex node starts transmitting first then the packet will reach is destination intact
- If the Full Duplex link starts transmission first:
  - the path is reserved until the packet end
  - A packet subsequently transmitted by the Simplex Node will be stored in the Router Rx buffer
  - At some point the buffer will be full and the transmission of a subsequent NCHAR will result in Rx buffer overrun at the Router

⇨ **Packet loss probability increases as the SpW packet size increases**

⇨ **Packet loss probability increases as the average traffic increases**

⇨ **Better probabilities for packets smaller than the Rx buffer size or with the Rx buffers of larger capacity**

⇨ **The situation becomes more difficult in complex topology networks with many Simplex nodes**



⇨ **For K = 1, a receiver may need a many N-K cycles to synchronize since the transition from the *ErrorReset* state to the *Connecting* state takes 12,8us+6,4us = 19,2 us which is larger than the K period**

⇨ **<u>Problems</u> are inherent to Simplex (<u>exist for both the SUAI and the FCT Based Initialization</u> solutions)**

Conclusions

☺ **Advantages:**

- Wiring is reduced by half resulting in weight and cost savings
- Router/Node Link Controllers' Gate count and subsequently cost is reduced
- Power consumption is reduced by at least 50 % due to the absence of one direction

☹ **Drawbacks:**

- There is no guarantee that a packet has been received at the remote end, Retry/Cold Redundancy/FDIR cannot be supported and therefore Simplex SpW is not appropriate for critical data
- Not recommended for large application packets
- It cannot be used in scheduled networks (SpW-D, SpW-T, …)
- It cannot support device configuration/Network Discovery/PnP
- It cannot support time-stamped data through Time Distribution
- The lack of flow control may cause permanent data loss for large packets if a Router's outgoing port (either Simplex, Full Duplex or Half Duplex) has lower effective rate than the incoming Simplex port
- The lack of flow control increases concentrators buffering requirements thus cancelling its low cost advantage

Fields of application:

- Hardware configured SpW devices
- Applications with small packet sizes (e.g. sensors) and large inter-packet gaps
- Concentrators which receive data from simplex links and propagate them to the backbone network through half/full-duplex SpW
- Low cost missions (e.g. nano sats) without any QoS requirements