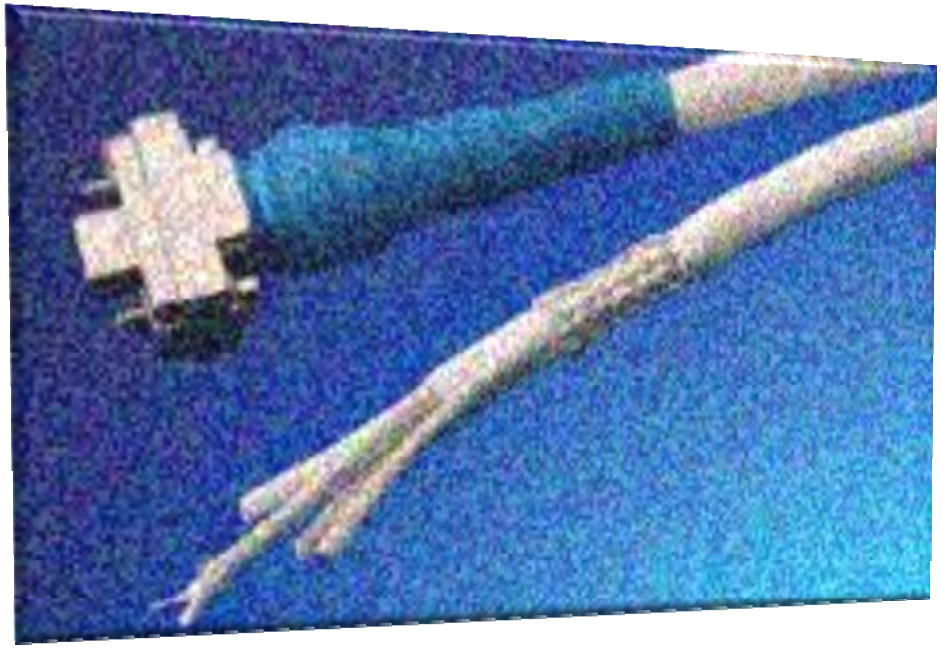


Technical Solution for Interrupts Distribution over SpW

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 and provides minimal latency characters (time-codes) for timing information

The Problem:

- Current SpW standard does not provide a mechanism for propagation of critical events (e.g. alarms)
- With the current standard if an a Node shall notify another node for the occurrence of a critical event a SpW packet shall be sent but:
  - Links between Nodes/Routers may be temporarily blocked
  - The event may be delivered with unacceptable delay

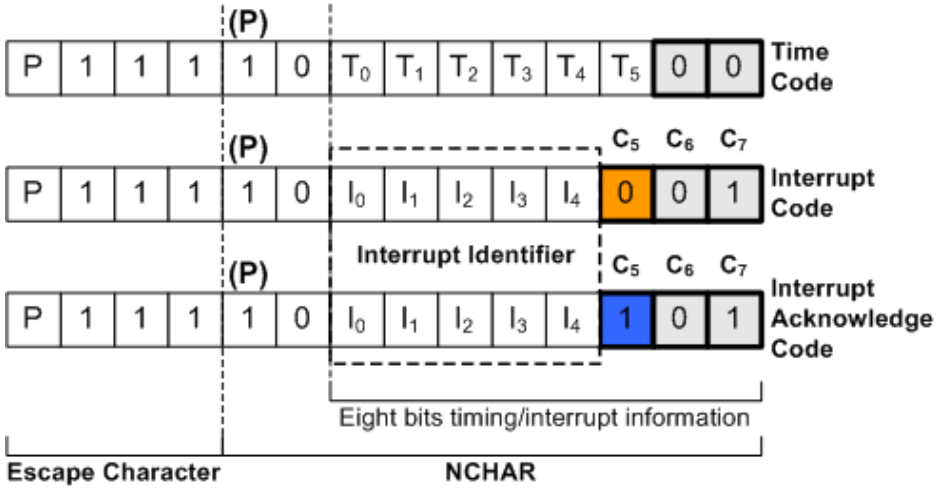
The University of St Petersburg (SUAI) proposed Solution:

- SUAI has proposed a solution for distribution of minimized latency interrupts and interrupt acknowledgements over SpW based on unassigned time-code characters

Time-Codes & Interrupts

The Time Code Characters:

- Time Code is a minimal latency character broadcasted throughout the entire SpW network
- Six bits of time information are held in the least significant six bits of the Time-Code (T0-T5)
- Two bits (T6, T7), assigned to “00”, contain control flags Time-Code
- The rest three T6, T7 combinations are reserved for future use



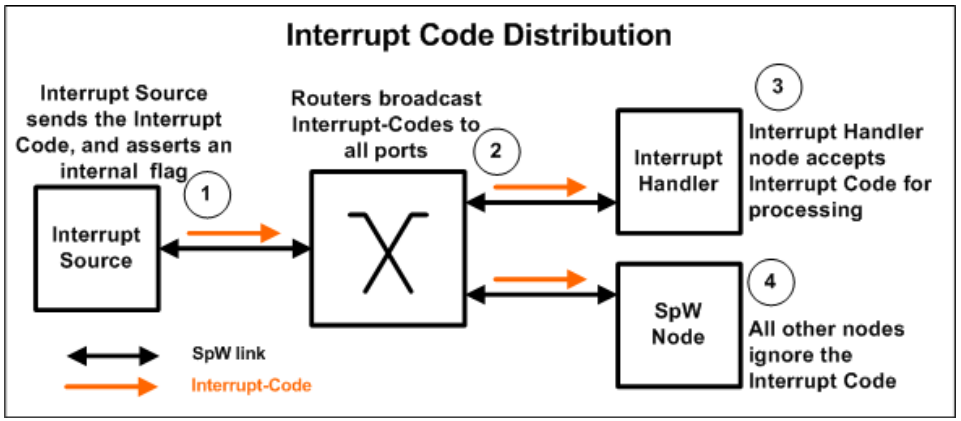
The SUAI Proposal:

- Define **Interrupt Codes**, which is a signal representing a request to handle an event of high priority
- Define **Interrupt Acknowledge Codes** which acknowledge Interrupt Code acceptance for processing by a handler
- Use the time-codes propagation mechanism to distribute interrupts/acknowledgements
- This ensures minimal propagation latency and propagation through blocked links
- Use one of the reserved T6, T7 combinations to define interrupts/acknowledgements
- Use on bit (C_5) to distinguish between Interrupt Code and Interrupt Acknowledge Code
- Use a five bits **interrupt identifier** (I_0-I_4) to define 32 Interrupt Codes and 32 Interrupt Acknowledgement Codes

Interrupt Codes & Interrupt Acknowledge Codes

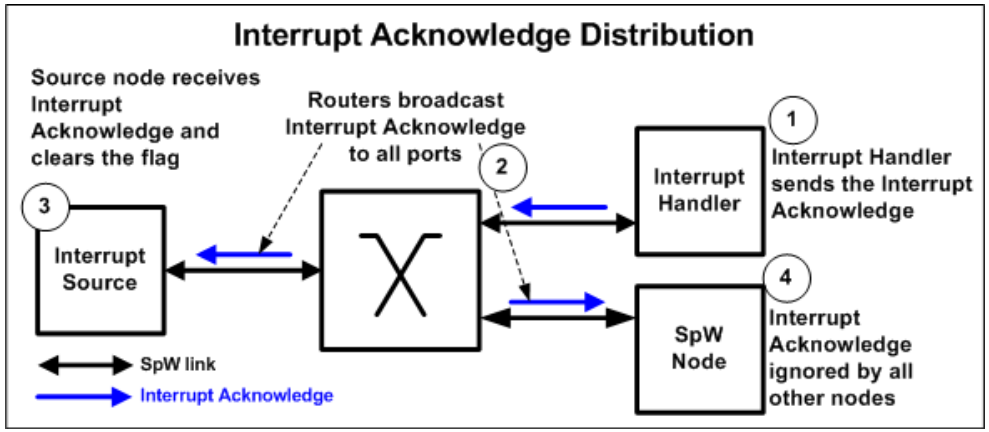
Interrupt Code:

- Issued by a Node, called Interrupt Source, to indicate a critical event
- Broadcasted by Routers to all links except for the link in which the Interrupt was received
- Accepted for handling at a Node, called Interrupt Handler, which will handle the request for the specific interrupt identifier
- All other Nodes ignore the Interrupt Code



Interrupt Acknowledge Code:

- Sent by the Node specified as the Interrupt Handler for the specific interrupt identifier and contains the same five-bit interrupt source identifier
- Represents a confirmation that the Interrupt Code has been accepted by the Interrupt Handler for processing
- Broadcasted by the Routers to all links except for the link in which the Acknowledge was received
- Acknowledge is received to the Interrupt Source notifying that the Interrupt has been accepted for processing by the Handler
- Acknowledge is received by all other Nodes in the network and it is ignored



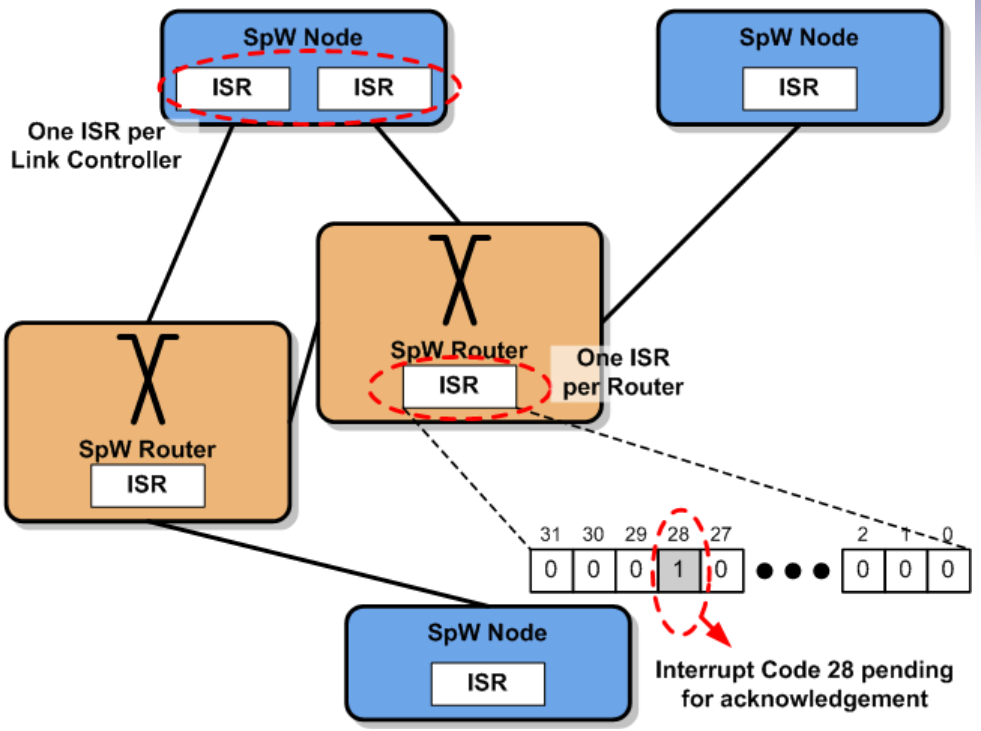
Interrupt Status Register

The Interrupt Status Register

- Each Link Controller of a Node and each Router contains one 32-bit Interrupt Status Register (ISR)
- Each ISR bit corresponds to one of 32 possible interrupt identifiers
- An ISR bit is set to ‘1’ upon the transmission or reception of an Interrupt Code with the corresponding Interrupt Source Identifier
- An ISR bit is cleared to ‘0’ upon the transmission or reception of an Interrupt Acknowledge Code with the corresponding Interrupt Source Identifier

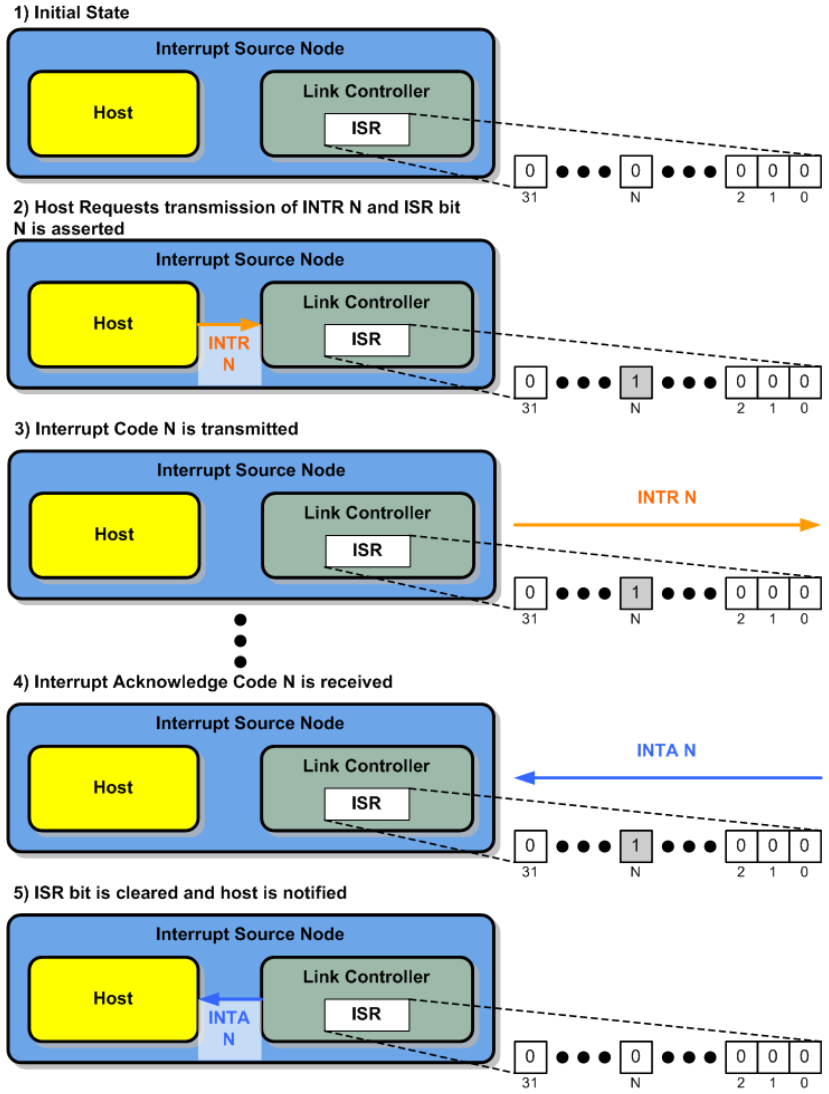
Interrupt Status Register Functionality

- Stores information about Interrupt Codes pending for Acknowledgement
- Initiates a time-out for each Interrupt Code broadcasted in the network
- Prevents repeated transmission of the same Interrupt Code or same Interrupt Acknowledgement in circular networks



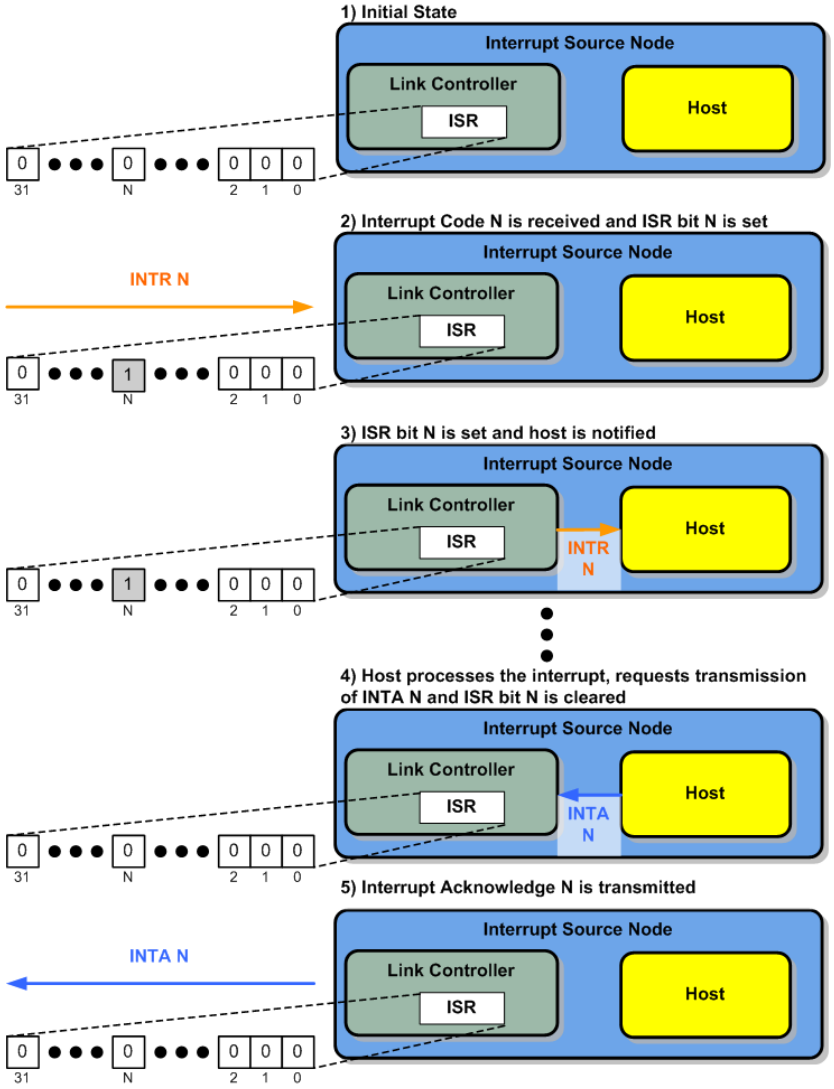
Interrupt Source Node

- The host indicates a significant event by issuing a request for transmission of an Interrupt Code
- Upon the reception of the request the Link Controller:
  - asserts the respective ISR bit
  - starts a time-out timer waiting for the Interrupt Code to be Acknowledged
  - transmits the Interrupt Code if the link is in the RUN state
- Upon the reception of an Interrupt Acknowledge the Link Controller:
  - clears the respective ISR bit
  - cancels the time-out timer
  - indicates to the Host that an Interrupt Acknowledge has been received, passing its identifier
- Conditions:
  - upon a host request, the Interrupt Code is sent **ONLY** if the corresponding ISR bit is '0'
  - upon an Interrupt Acknowledge reception, the host is notified **ONLY** if the corresponding ISR bit is '1'



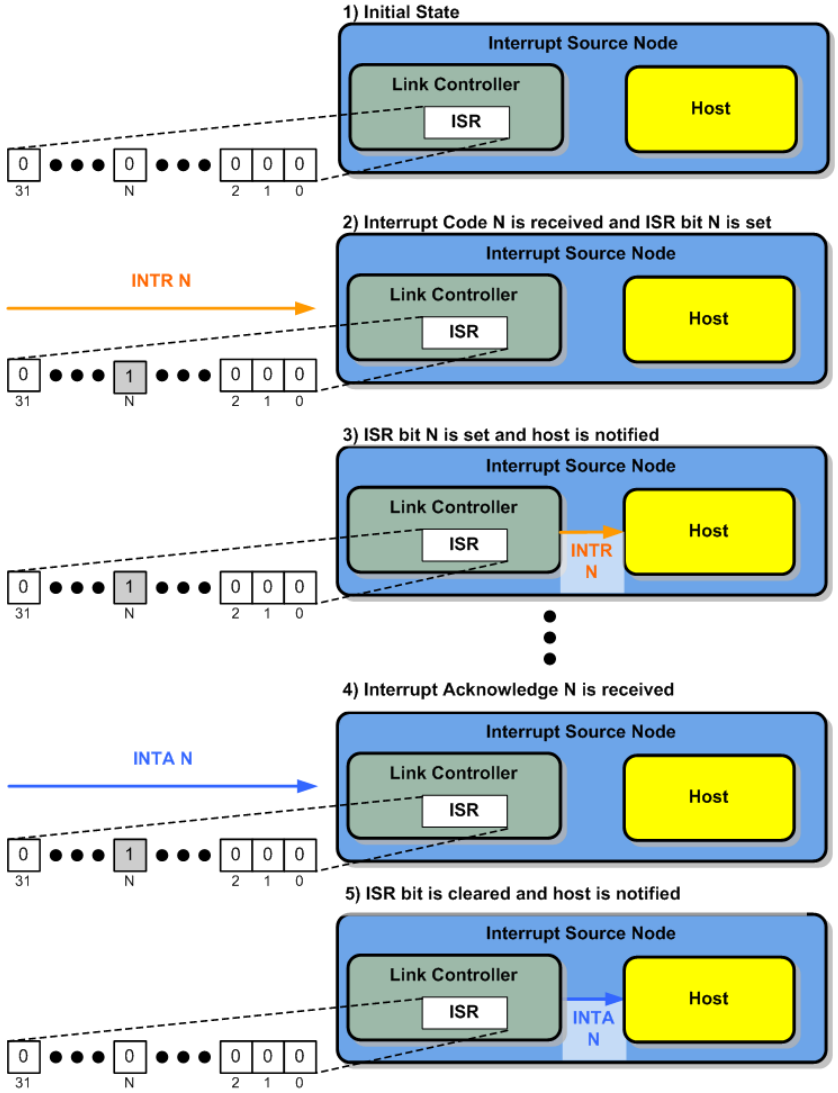
Target Node - Interrupt Handler Node

- Upon the reception of the Interrupt Code the Interrupt Handler Node:
  - asserts the respective ISR bit
  - starts a time-out timer waiting for the Interrupt Code to be Acknowledged
  - indicates to the host that an Interrupt Code has been received and passes its identifier
- The host indicates that the Interrupt shall be acknowledged
- Upon the reception of the indication the Link Controller:
  - clears the respective ISR bit
  - cancels the time-out timer
  - transmits the Interrupt Acknowledge if the link is in the RUN state
- Conditions:
  - Upon the reception of an Interrupt Code the host is notified **ONLY** if the corresponding ISR bit is '0'



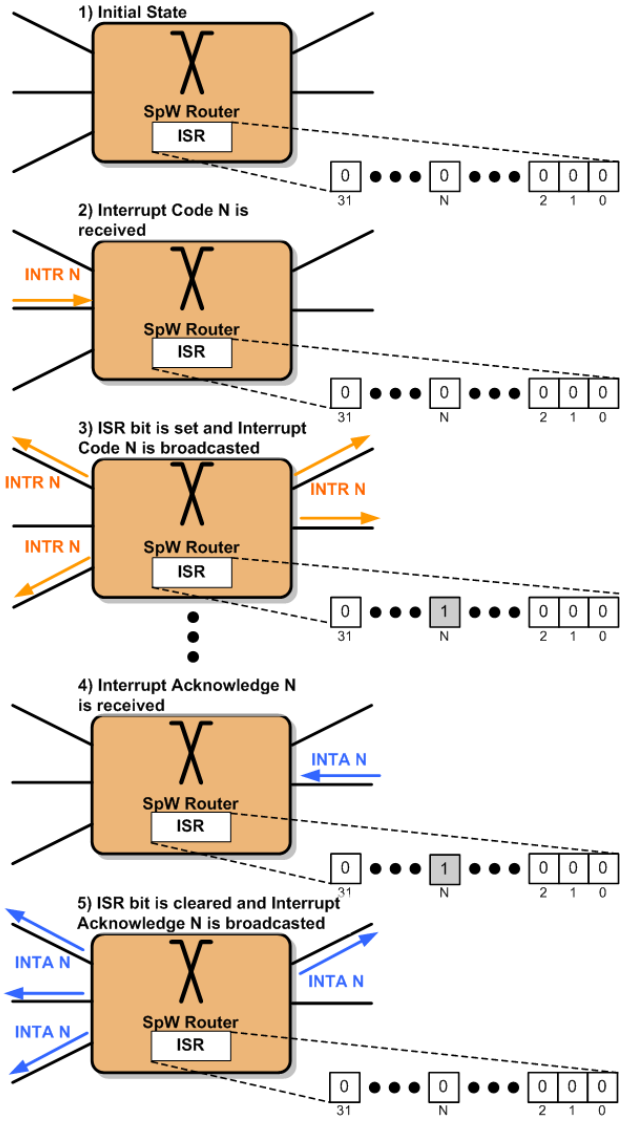
Target Node – Non Interrupt Handler Node

- Upon the reception of the Interrupt Code the Interrupt Handler Node:
  - asserts the respective ISR bit
  - starts a time-out timer waiting for the Interrupt Code to be Acknowledged
  - indicates to the host that an Interrupt Code has been received and passes its identifier
- ~~The host indicates that the interrupt shall be acknowledged~~
- ~~Upon the reception of the indication the Link Controller:~~
  - ~~clears the respective ISR bit~~
  - ~~cancels the time-out timer~~
  - ~~transmits the Interrupt Acknowledge if the link is in the RUN state~~
- Upon the reception of an Interrupt Acknowledge the Link Controller:
  - Clears the respective ISR bit
  - Cancels the time-out timer
  - Indicates to the Host that an Interrupt Acknowledge has been received, passing its identifier



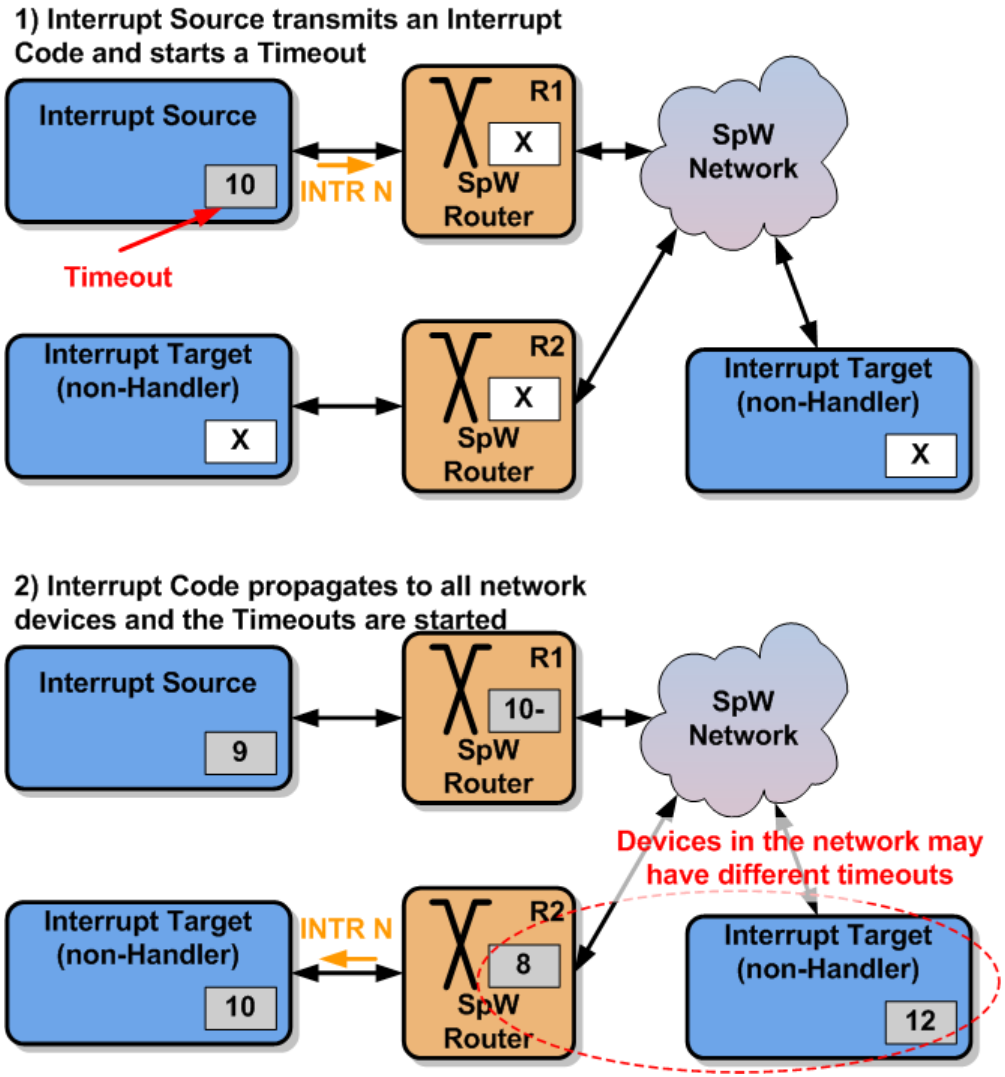
Router

- Upon the **reception** of the Interrupt Code the Router:
  - asserts the respective ISR bit
  - starts a time-out timer waiting for the Interrupt Code to be Acknowledged
  - broadcasts the Interrupt Code to all ports except for the port the Interrupt Code was received
- Upon the reception of an Interrupt Acknowledge the Router:
  - clears the respective ISR bit
  - cancels the time-out timer
  - broadcasts the Interrupt Acknowledge to all ports except for the port the Interrupt Code was received
- Possible to receive simultaneous Interrupt Codes/Acknowledgements:
  - one FIFO for the Interrupt Codes and one for the acknowledgement shall be implemented
  - Priority mechanism is TBD or optional but
  - this may result in non-consistent timeouts throughout the network (in the range of us)



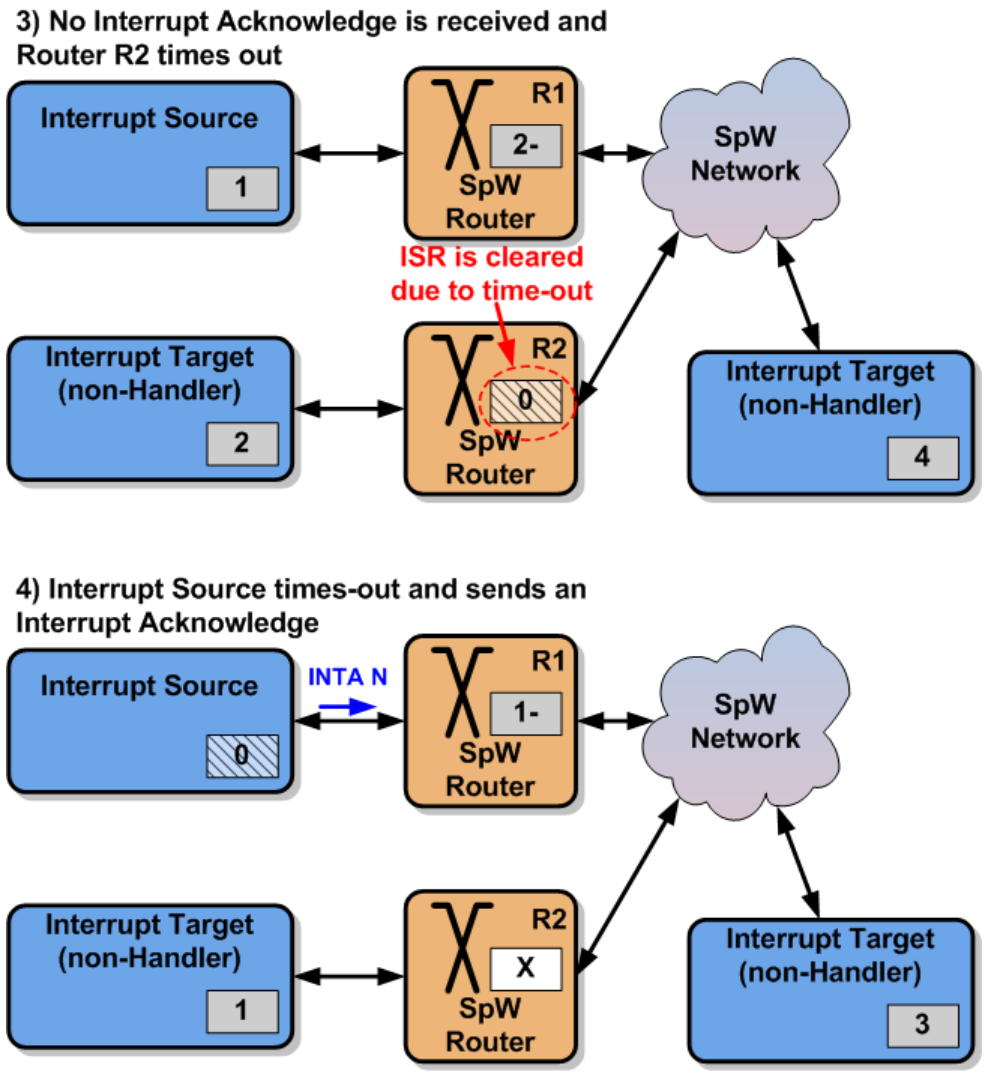
Timeouts (1/3)

- An Interrupt Code is transmitted
- A timeout is initiated at the Interrupt Source
- The Interrupt Code is propagated throughout the entire network
  - the respective ISR bit is set in all devices
  - A timeout for the specific interrupt identifier is initiated in all devices
- Devices may have different timeout values, or their oscillators may drift or the timeout timers may not have the same resolution
  - It cannot be guaranteed that the timeouts will be consistent throughout the entire network
  - The timeout of a device may expire before the others
  - Interrupt Acknowledge that will be subsequently sent by the Handler, or the Source may not propagate through the entire network



Timeouts (2/3)

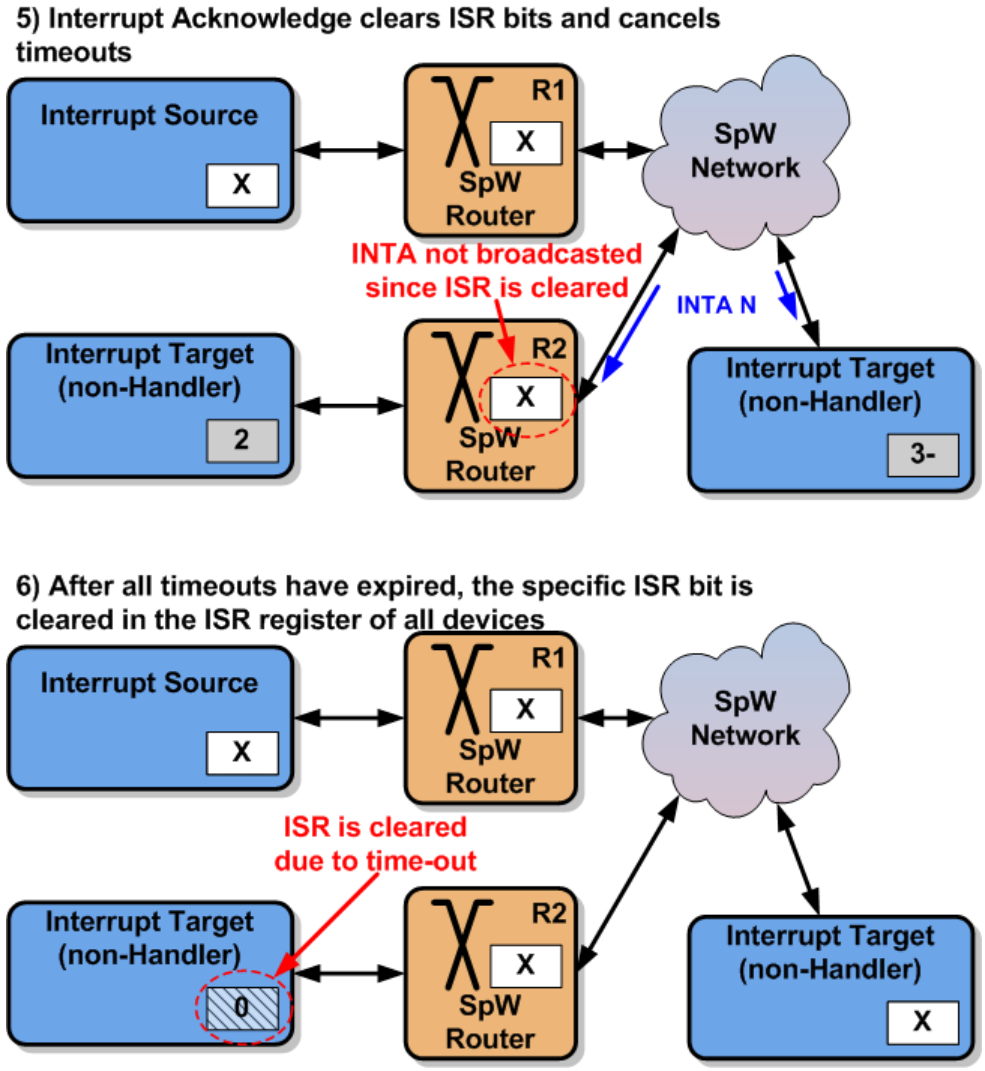
- In case the Interrupt Handler does not send the Interrupt Acknowledge within time
- A network device will timeout first and
- will clear the ISR bit which corresponds to the expired timeout
- This device may be a router which means that subsequently received Interrupt Acknowledge for this specific identifier will not be broadcasted by the router
- The Interrupt source may timeout after the time-out of the router and
- It Clears its ISR bit and sends an Interrupt Acknowledge corresponding to the Interrupt Identifier whose timeout expired



Timeouts (3/3)

- The Interrupt Acknowledge will propagate in the network until it reaches a router in which the timeout for the specific interrupt identifier has expired
- Devices in which the Interrupt Acknowledge is not received will clear the respective ISR upon the expiration of their internal timeout

⇨ **All devices implement the ISR and timeout mechanism**



Operation in Circular Networks

- An Interrupt Code is transmitted by the Source and reaches the Handler in two hops
- The Interrupt Handler “immediately” sends back the Acknowledge and the Interrupt Service Routine **executes for the first time**
- The Acknowledge reaches the farthest router where it is ignored since the respective ISR bit is not yet set
- The Interrupt Code reaches the last router and is propagated to the Handler again

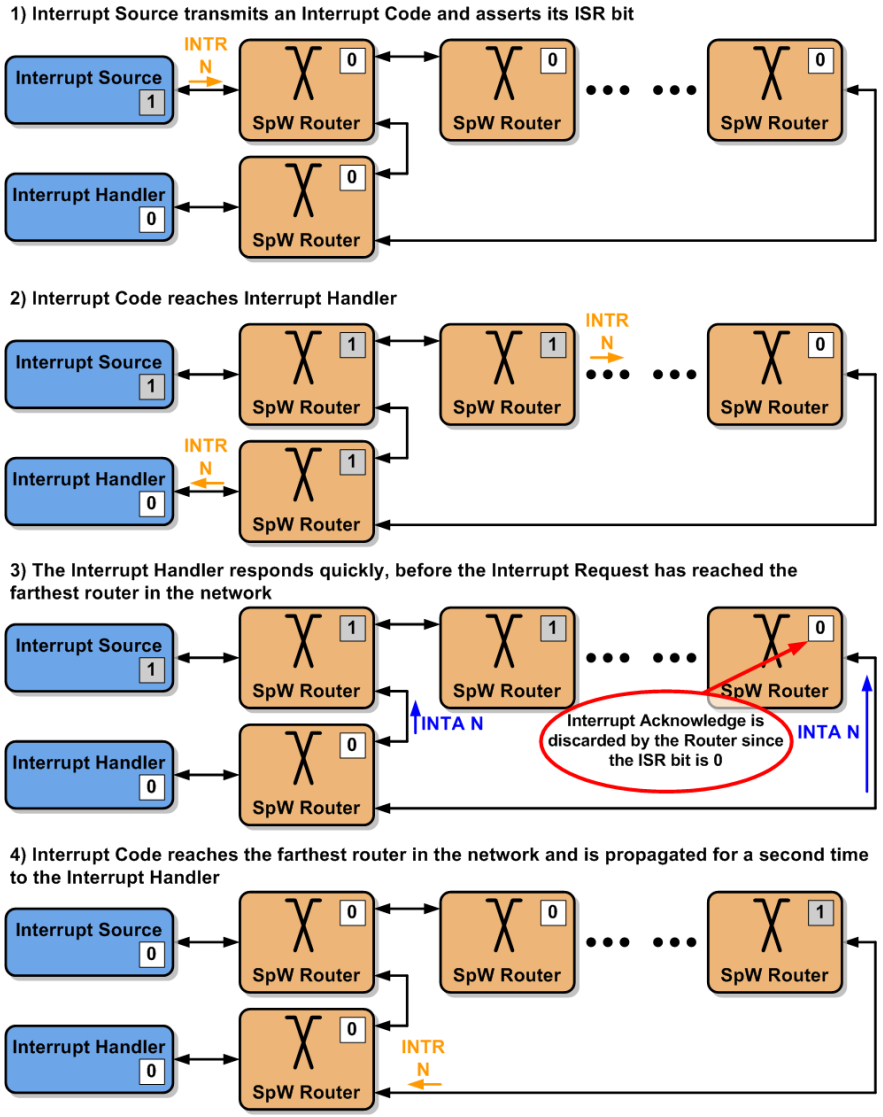
⇨ **The Interrupt Service Routine is executed again**

⇨ **Same case if the Source has redundant links**

⇨ **May be fatal if the Handler controls an actuator**

⇨ **Similar case if a second Interrupt Code is sent while the Acknowledge is still being propagated**

- **Proposed additions:**
  - Specify a minimum for the Interrupt Handler time response related to the network diameter (SUAI)
  - Specify a minimum time for handling the same interrupt request



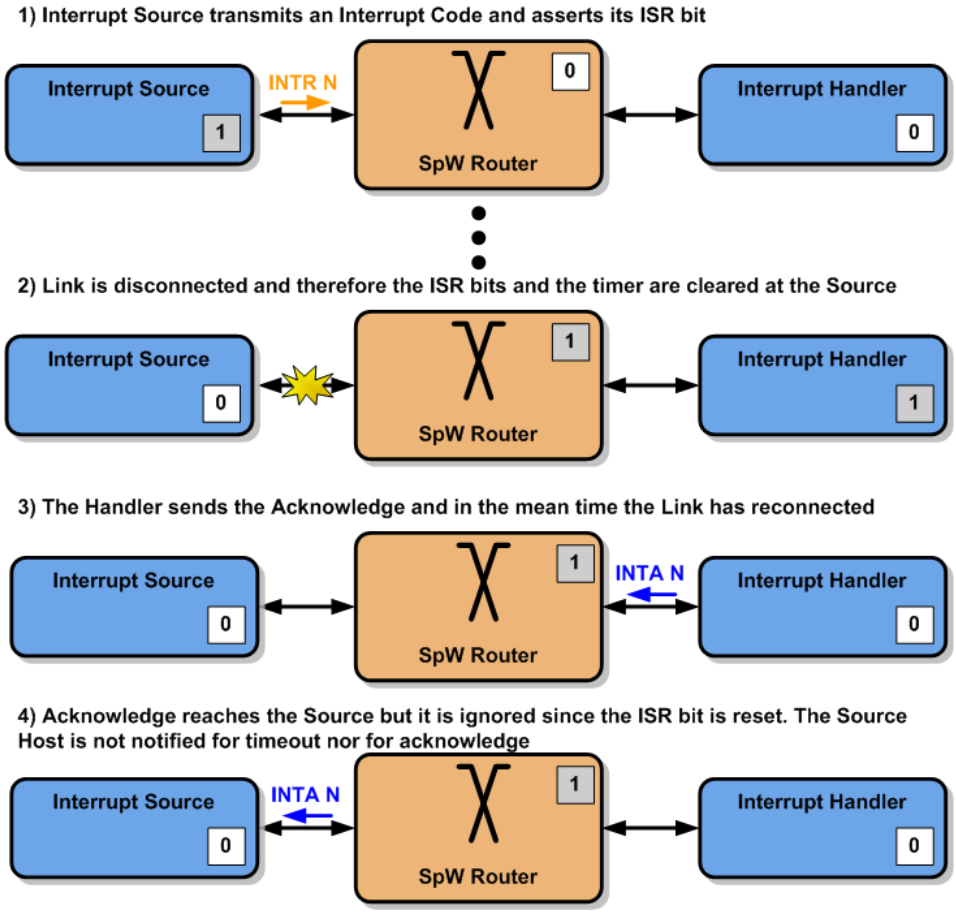
Reset on Disconnect

- **8.13.1 w: After a disconnect-reconnect the ISR bits shall be set to zero**
- **8.13.3 g: After a disconnect-reconnect the ISR timers shall be set to zero**
- The Interrupt Source transmits an Interrupt Code
- Interrupt Source's link disconnects and reconnects before receiving the Acknowledge
- The ISR and the ISR timers are reset due to the disconnect
- The Interrupt Handler sends the Interrupt Acknowledge, which reaches the Source but it is ignored since the ISR bit is cleared

⇨ **The Source Host does not receive the Acknowledge indication nor is it notified for timeout. Infinite timeout at the host!**

⇨ **Similar case if the Acknowledge is sent while the Source link is in disconnect**

- **Proposed modification:**
  - ISR bits and timers shall not be affected by a disconnect



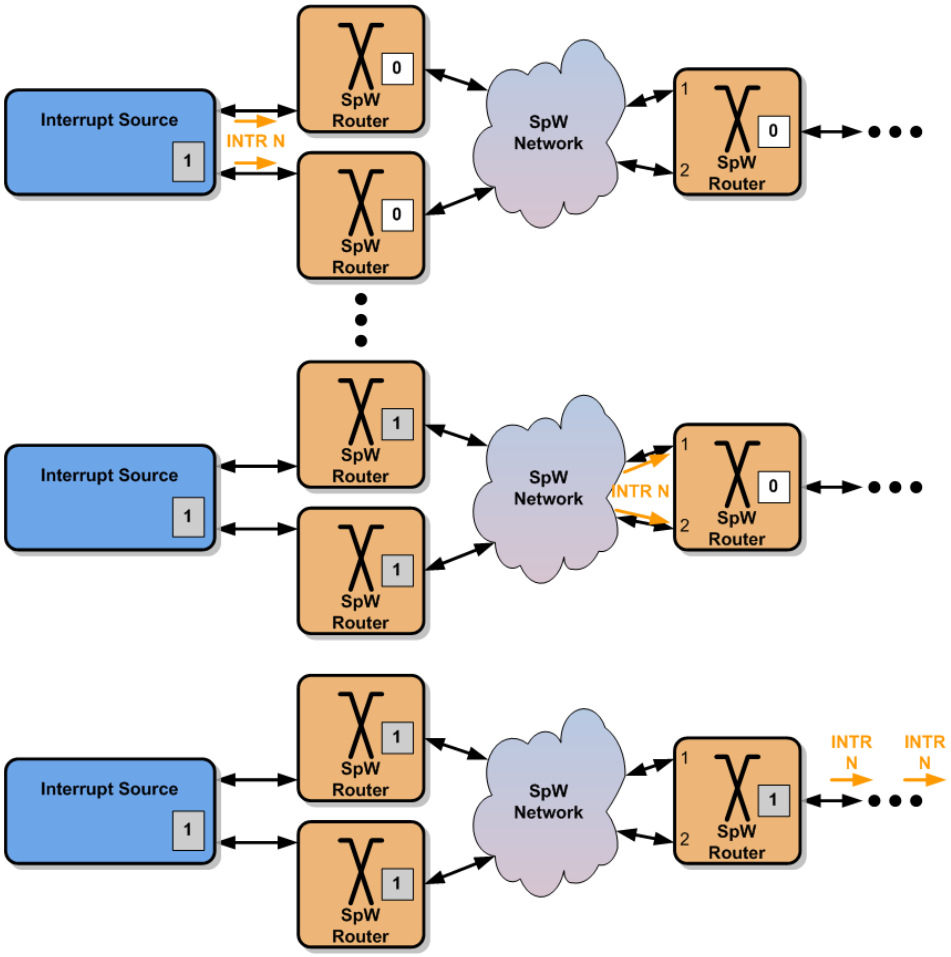
ISR Handling

- A node with redundant links transmits an Interrupt Code
- The Interrupt Code arrives through two paths at a single router
- Upon Interrupt's arrival at port 1 the ISR is checked and found to be 0
- Upon Interrupt's arrival at port 2 the ISR is checked and found to be 0
- An Interrupt Code is queued due to the Interrupt received through port 1 and the ISR bit is set
- An Interrupt Code is queued due to the Interrupt received through port 2 and the ISR bit is set again

⇨ **Two Interrupt Codes are propagated through the link to the next hop**

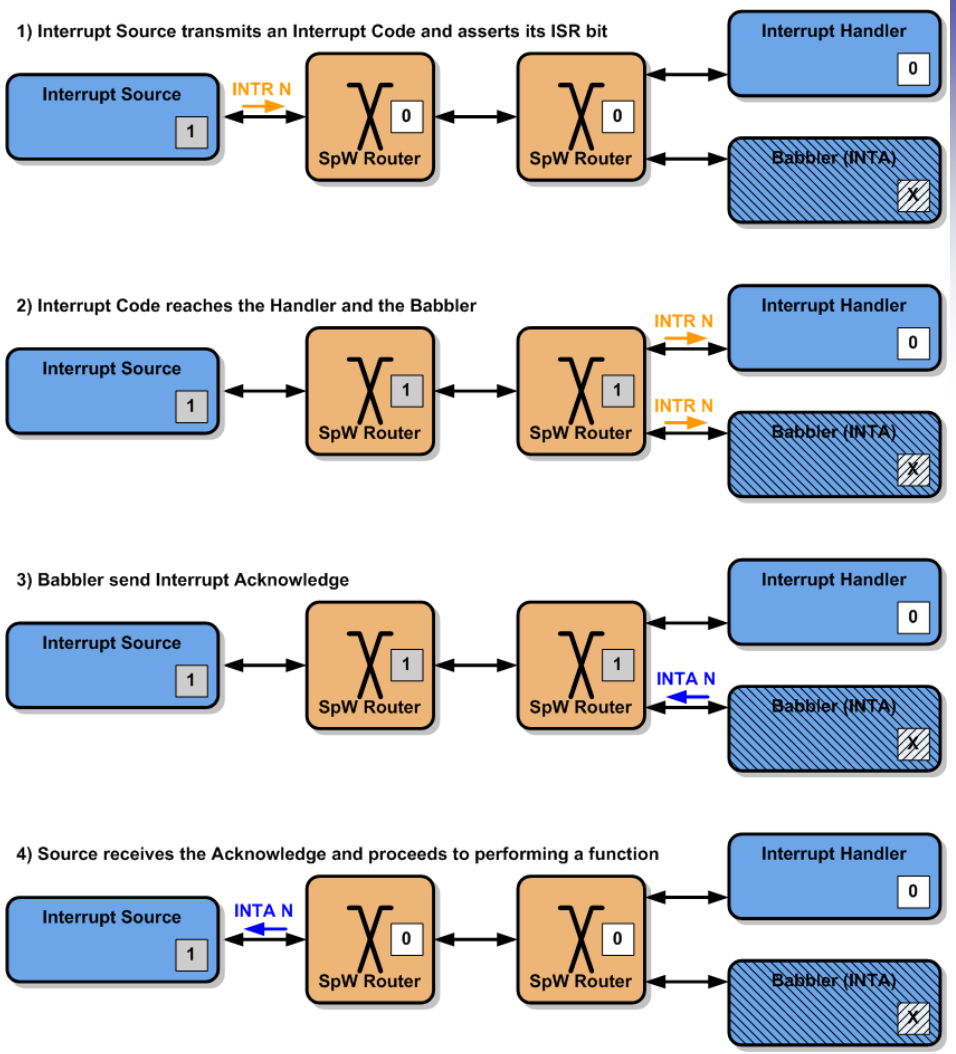
⇨ **May be logged as an error and initiate FDIR**

- **<u>Proposed clarification:</u>**
  - The routers shall check and set/reset ISR bits through **ATOMIC** operations



Babbling idiot

- A faulty node in the network responds to all (or certain) Interrupt Code(s)
- An Interrupt Code is sent by the Interrupt Source
- The Interrupt Code is delivered to both the Interrupt Handler and the faulty node
- The faulty node responds (erroneously) with an Interrupt Acknowledge
- ⇨ **The Acknowledge is delivered to the Source and "grants" the execution of further actions**
- ⇨ **Interrupt is erroneously acknowledged if Handler is disconnected or off**
- ⇨ **Similar case if a Babbler sends Interrupt Codes**
- **Proposed solution:**
  - Protection can be provided by **edge** routers
  - Assign allowed/expected Interrupt Code(s) and Acknowledge(s) per edge Router port
  - Consecutive range of allowed Interrupt Code(s) and Acknowledge(s) per port eases implementation (INFORMATIVE)
  - Provide error information in routers for FDIR (remote FDIR through PnP)?



Summary of modifications & additions to the SUAI proposal

- If the link is not in the RUN state upon host's request for Interrupt Code transmission, the Link Controller shall notify the host and ignore the request
- At the Interrupt Source, a timer shall start upon transmission of the Interrupt Code through the SpW link (imposes extensions to the SpW CODEC)
- There is one Interrupt Mask register per node which defines for which Interrupt Identifiers the node is the Interrupt Handler (INFORMATIVE?)
- Specify a minimum time for the Interrupt Handler response related to the network diameter (SUAI) – the largest diameter may correspond to a circular connection
- Specify a minimum time for the transmission of an Interrupt Code after reception of an Interrupt Acknowledge with the same Interrupt Identifier
- Implement Cold Redundancy at both the Source in circular networks where possible (INFORMATIVE)
- ISR bits and timers shall not be affected by a disconnect
- The routers shall check and set/reset ISR bits through **ATOMIC** operations
- Protection can be provided by **edge** routers
- Assign allowed/expected Interrupt Code(s) and Acknowledge(s) per edge Router port
- Consecutive range of allowed Interrupt Code(s) and Acknowledge(s) per port eases implementation (INFORMATIVE)
- Provide error information in routers for FDIR (e.g. in order to support remote node deactivation through PnP)?

Conclusions

☺ **Advantages:**

- The proposed interrupt mechanism provides minimal latency and its performance is not affected by blocked links
- It does not require (or requires minimal) modifications on the SpW CODEC since the mechanism uses SpW CODEC's Time-Code I/F

☹ **Drawbacks:**

- Interrupt Codes/Acknowledgements may insert more jitter to a Time Code than a NCHAR (14 bits instead of 10)
- Limitation for up to 32 Interrupt Code/Acknowledge pairs whereas up to 224 nodes may be present in the network
- Performance on appearance of simultaneous interrupts is not predictable
- Router design challenges due to the possibility for multiple simultaneous Interrupt Codes/Acknowledges
- Although compatible with the SpW standard the proposal is incompatible with certain existing implementations (GR SpW-RTC, RUAG COLE ASIC)
- Does not allow the Interrupt Source to be Interrupt Handler for the same Interrupt Identifier. Could it be required by IMA/TSP partitioning?
- Failsafe requirement adds complexity to the Router:
  - Faulty node protection by edge routers increases router complexity. Without taking into account TMR/EDAC/scrubbing, up to 32+32 bits per port are required
  - FDIR statistics by edge routers increase router complexity even more. Without taking into account TMR/EDAC/scrubbing, for N Interrupt/Acknowledge pairs, 2xN bits per port are required to count up to 2^N Interrupt Code and Interrupt Acknowledge errors per port

Back Up Slides

Extensions for more than 32 Interrupts (1/2)

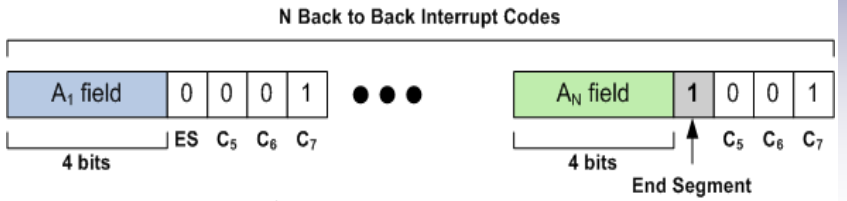
- **Use more than one T7, T6 combinations:**
  - ☑ Minimum latency Broadcasted Interrupts/Acknowledgments
  - ☒ Uses all broadcasted characters and does not allow future extensions
  - ☒ Has an upper limit of 96 Interrupts
- **Virtual Networks (4Links):**
  - SpW CODEC performs a “context switch” upon the reception of the Signaling Code (Reserved Time Code character)
  - ☑ Interrupt vectors formed by SpW packets
  - ☑ Unlimited number of Interrupts
  - ☑ Possible to define NACK packet
  - ☒ Violates the SpW 1.0 standard since the Signaling Codes are not broadcasted?
  - ☒ Flow Control may result in blocked links which is not desirable for Interrupts/Acknowledgements
  - ☒ Introduces latency/jitter to application packets
  - ☒ No broadcast/multicast support
  - ☒ Duplicate Interrupt reception in circular networks – requires redundancy filter (same with application packets)
  - ☒ Requires CODEC modifications in order to synchronize Signaling Codes and NCHARs transmission
- **Back to Back Interrupt Codes:**
  - More than one broadcasted characters used to form the Interrupt vector and Segmentation/Reassembly at the receiving end
  - ☑ Support for up to 256 Interrupts
  - ☑ Broadcasted Interrupts/Acknowledgements
  - ☑ Possible to define NACK packet by adding an extra control bit and reducing the number of Interrupts **(Do we need more?)**
  - ☑ Can be extended to support more interrupts, with more than two consecutive Codes, with the addition of a toggle bit (for lost code detection) at the expense of additional latency
  - ☒ Violates the SpW standard since the Signaling Codes are not broadcasted individually
  - ☒ Introduces latency/jitter to application packets

Extensions for more than 32 Interrupts (2/2)

- **Signaling Code - NCHAR - Signaling Code:**
  - NCHAR interleaved between two Signaling Codes
  - Similar to 4Links Virtual Network - SpW CODEC performs a “context switch” upon the reception of Signaling Codes
  - ☑ Broadcast interrupt support for up to 256 Interrupts
  - ☑ Possible to define NACK packet
  - ☒ Introduces latency/jitter to application packets
  - ☒ Requires modifications to the SpW CODEC transmitter in order to synchronize transmission of Signaling Codes and NCHARs
  - ☒ Requires modifications to the SpW CODEC receiver in order not to use the credit logic when in “Interrupt context”
- **Dedicated SpW Packet:**
  - Insert a (bounded size) SpW Packet instead of a single NCHAR – extension of the previous proposal
  - ☑ Unlimited number of interrupts
  - ☑ Support for Broadcasted and/or Routed Interrupts (but broadcast violates SpW 1.0 standard)
  - ☑ Possible to define NACK packet
  - ☒ Introduces Latency/jitter to application packets
  - ☒ Requires modifications to the SpW CODEC transmitter in order to synchronize transmission of broadcasted characters and NCHARs
  - ☒ Requires modifications to the SpW CODEC receiver in order not to use the credit logic during the context switch
  - ☒ Additional resource utilization – e.g. For a N hops network, N + 4 Bytes of Interrupt information (2 for Interrupt Identifier, 1 for Control, 1 for CRC) are required per router port due to lack of Flow Control. For a 12 hops network and 8 ports on the Router, this means 128 Bytes for Interrupt Codes and 128 Bytes for Acknowledgements are required in order to handle simultaneous arrival on all ports

Back to Back Interrupts

- Functionality similar to the SUAI proposal (ISR, timeouts …)
- Two (or more) consecutive Signaling Codes are transmitted by the Source
  - Sent one after the other with a microseconds-range gap
  - No other Signaling Code can be interleaved
  - NULLs, FCTs, NCHARs and Time Codes are allowed to be inserted between the consecutive Interrupt Codes
  - Each Signaling Code contains 4 bits of the interrupt vector
  - The “End Segment” field determines it the first or last segment is being received
  - More than 256 interrupts can be supported with more back to back codes and the addition of an extra control bit (e.g. toggle)



Up to 2^{4xN} Interrupts can be supported
For N = 2, 256 Interrupts are supported

Link Controller Functionality:

- Upon the reception of an Signalling Code the Link Controller it checks if the End Segment bit is 0 and if it is not it is rejected
- If the End Segment bit is 0 a time-out timer (microseconds range) is started until the reception of the second Signalling Code
- If the time-out expires and the second Signalling Code has not been received the Link Controller discards the first segment
- When the second Signalling Code is received the Link Controller checks if the End Segment bit is 1 and if it is not, it rejects this segment and discards the first segment
- If both segments are received correctly the Link Controller asserts a signal and passes the Interrupt Identifier information and the type (Interrupt Code or Acknowledgement)

Signalling Code - NCHAR – Signalling Code

- The Interrupt request is signalled by a sequence of a signalling code and NCHAR and another signalling code in which the NCHAR carries the Interrupt Identifier
- The sequence is interleaved within a SpW Packet Transmission
- Time Codes may be interleaved
- The receiver Acknowledges reception of the each signalling code in order not to corrupt the interrupted packet transmission in case of signalling character loss **(can this happen?)**
  - In case the first signalling code is not acknowledged within the time out period (microseconds range) the transmitter may retry and if all retries fail the host application is informed
  - In case the transmission of the last signalling code is not acknowledged, the host application is informed (in this case normal packet transmission may never be resumed!)
  - ☑ Ensures correct operation in the case that one of the signalling characters is lost
  - ☒ Introduces significant latency per link and end-to-end and inserts jitter to the application packets
  - ☒ In a Router acknowledgements shall be received by all ports in order to proceed sending another, pending, interrupt character and thus additional delays are introduced
  - ☒ The last acknowledge may be lost and interrupted packet transmission will not resume
- The ideal is similar with the SpW Virtual Networks proposed by 4Links with the following differences:
  - There are no FCTs since their presence may result in blocked links
  - There is no per-port buffer in the Routers but one buffer for all Interrupts Codes and one for Acknowledgements
  - The signalling codes shall be acknowledged before proceeding to sending NCHARs
- Can be extended to carry entire SpW packet of bounded size instead of a single NCHAR in order to define SpW Packet based Interrupt/Acknowledge protocol and can support Broadcasted or Routed Interrupts
- Can be modified to operate as Signaling Code – NCHAR - EoP

Extending the number of supported Interrupts - Comparison

Proposal	T7, T6 bits	Back to Back Interrupts	Virtual Networks	Signaling Code NCHAR Signaling Code
Latency	Excellent	Excellent	Excellent/Very Good	Very Good
SpW Link Blocking	No	No	Yes	No
Max supported Interrupts	32 – 96	Up to 256	Unlimited	At least 256
Acknowledgement	No	No	No	Yes, if handshake is employed
Bounded message size	Yes	Yes	Not specified	Yes
Broadcasted Interrupts	Yes	Yes	No	Yes
Routed Interrupts	No	No	Yes	Extended version only
NACK packet	No	Possible	Yes	Yes
Compatibility with SpW Spec	Yes	No	No	No
Compatibility with existing devices	No	No	No	No
Jitter insertion to applications	Minimum	Minimum	Medium	Medium
Single Interrupt reception in redundant/circular configurations	Yes	Yes	Needs redundancy filter	Yes
Failure behavior & Handling	Good	Good	TBD	TBD
SpW CODEC modifications	No	Minimum	Significant	Significant
Required Memory Resources	Low	Low	Medium	Low - Medium

Implementation Issues - Simultaneous Interrupts

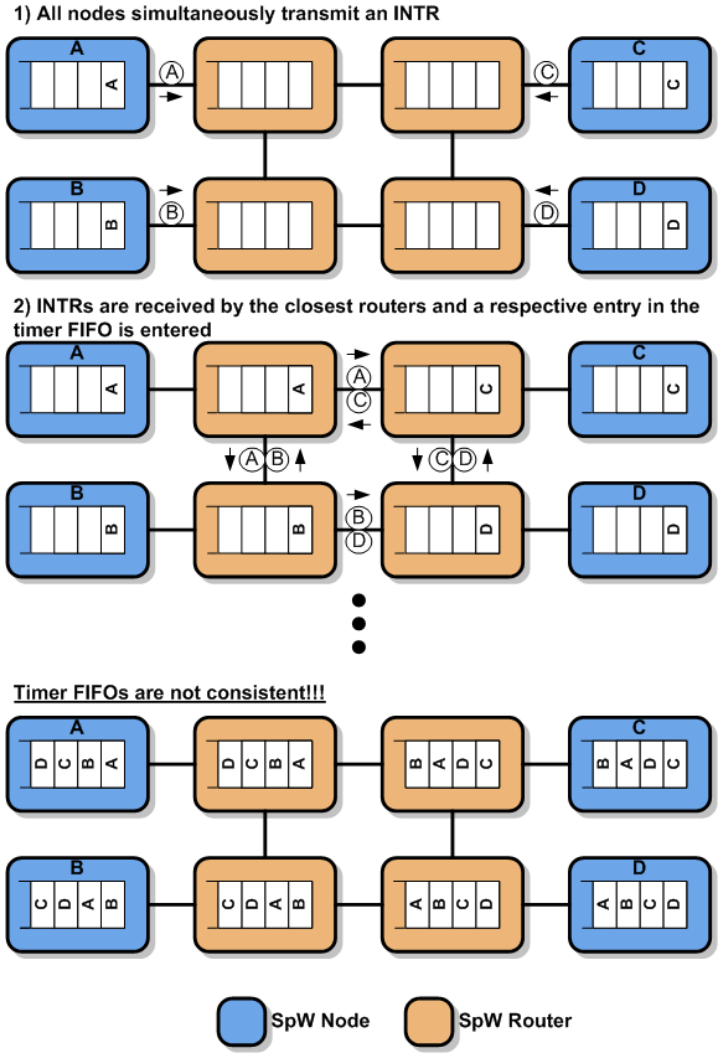
- The intent of the SUAI proposal is to create **network-wide** distributed and **synchronized** ISRs/Timeouts
- ISR Timers shall be consistent throughout the entire network
  - Nodes A – D transmit Interrupt Codes simultaneously
  - Adjacent routers propagate interrupt of the nearest host first
  - The exact sequence of events cannot be predicted accurately

⇨ **Timer FIFOs are cannot be consistent throughout the network**

⇨ **FIFO output may not correspond to the received acknowledge**

⇨ **Similar case if more than one interrupts are pending and the Handlers have different Interrupt Handling times**

- Possible to receive simultaneous Interrupt Codes/Acknowledgements:
  - ⇨ one Buffer for the Interrupt Codes and one for the acknowledgement shall be implemented
  - ⇨ Interrupt Acknowledge handling by routers cannot be implemented in FIFO mode



Multicast

- For each Interrupt Code/Acknowledgement the router stores the allowed outgoing ports for each Interrupt Identifier
  - Avoids the problem of repetitive transmission of the same interrupt in redundant/circular networks
  - Requires 2xNxM bits, where N is the total number of interrupts and M are the router ports
  - For 256 Interrupts and 8 ports the total number is 4096 Bits
  - The ISR and Timeouts are not required at the routers and therefore N bits are saved for the ISR and NxP bits for the timeouts, where P is the timeout counters length
  - For 256 Interrupts this saves 256 + 256xP. For P = 8, 256 + 2048 bits are saved
  - For the SUAI proposal 512 Bits of memory are added and 32+256 bits are saved
- Failure mode:
  - A malfunctioning router may send the Interrupt/Acknowledge to a non-allowed port
  - The router which receives it can reject it if incoming protection is employed at the router (see slide 13) – no single point of failure

☑ Avoids the problem of network-wide timeout synchronization

☑ Allows redundant paths to the destination and at the same time solves the propagation problem in circular networks

☑ Simpler Logic implemented at the routers

☒ Violates the SpW 1.0 standard since the Signaling Codes are not broadcasted

☒ Additional memory resources are required per router

☒ More difficult Router configuration/reconfiguration