NASA SpaceWire Activities/Comments/Recommendations

SpaceWire Working Group Meeting

ESTEC, Noordwijk, NL
May 18-19, 2006

Presented by
Glenn Rakow - NASA/GSFC

Contents

- Protocol ID assignment
- Protocol development
- Plug & Play (PnP)
- Recommended additions to SpW protocol
- SpaceFibre trade

Protocol ID Assignment

- **Protocol ID assignment philosophy**
  - Large Protocol ID space (16 bits)
    - Advantages to having more assignments?
      - More choices
        - More manpower to solve common satellite applications problems and to improve on existing work
      - More confusing
        - Too many choices
  - Will SpW working group support multiple similar protocols?
    - Example - General Access Protocol (GAP) and RMAP
    - Perhaps all supported protocols not all part of ECSS-E-50-11 or standardized under ECSS
- **How will future protocols be documented?**
  - Web-site?
  - Standardized?
- **Differences at protocol level between devices should not necessarily present architectural problem**

Protocol Development

- **Most US satellite missions use protocols in experimental range**
- **Several protocols have been developed with working implementations from multiple institutions**
  - General Access Protocol (GAP)
    - Similar to RMAP
      - Can differences between RMAP and GAP be resolved?
  - Reliable Data Delivery Protocol (RDDP)
    - Acknowledgement & retry mechanism
      - For generic packet cargo identifiable via sub-protocol ID
- **GAP is base lined for multiple missions**
- **RDDP is base lined for GOES-R**
  - NOAA/NASA weather satellite
- **Developers of protocols would like permanent Protocol ID assignments**
  - Recommend formal presentation of GAP & RMAP at next working group meeting

Plug and Play (PnP)

- **What needs to be done to make SpW routers & nodes to PnP?**
- **US industry & government investigating these issues**
  - How can US & SpW working group collaborate
    - New working group with ECSS path?
- **Network Discovery**
  - Using RMAP and/or GAP
- **SpW standard needs clarification for**
  - Priority
  - Group Adaptive routing
  - Configuration 0 space
- **Device Enumeration**
  - Not necessary SpW specific
    - However some advantages to use RMAP and/or GAP

Recommended Additions to SpW protocol

- **Many satellite architectures require redundancy at Physical level**
  - Transparent to user is preferred
    - Autonomous switch-over
  - This is something that should be address by standard
  - NASA has a implementation for Physical level redundancy
- **Single Time-Code (TC) master is restrictive**
  - Many systems would like to have more than one TC master
  - Current standard may be easily extended to four

SpaceFibre Trade

SpaceFiber Goals

- **Use DC balanced encoding to obtain Gigabit rates**
  - 8b10b
  - Ability to use copper or Fiber depending upon requirements
  - To what extent is variable rate possible? How do you change rates? PLL? On fly?
- **Backward compatible to SpW**
  - Bridge between two link protocols via Switch
  - Maintain worm-hole routing capability
- **Ability to check for packet errors on fly but not have to wait until the end of the packet for faster recovery**
  - How do you place error detecting code on data
  - At what boundary - byte, field (size?), packet
- **Take advantage of K codes for logical characters to simplify implementation**
  - Is error coding required on K Codes
- **Minimize synchronization sequence**
  - Is it necessary?
  - If so how often?
  - And how long?
- **Maintain bandwidth efficiency as much as possible**
  - Should Flow Control Tokens (FCTs) represent more than 8 N characters
  - Should N-Characters be replaced with Data characters

SpaceFiber Trade Investigations

- **What is the optimal length for error detection coding for SpW to reduce overhead but yet react quickly to prevent network blockage?**
  - Error detection code at end of packet or per data length field?
    - How long a field?
  - What type of error detection code
    - CRC (8 bit?)
      - Length?
    - Checksum?
- **Can K codes errors be detected as something other than what is desired? Can they be interpreted as good data another K code, etc.**
- **Should a bad K code bring down the link?**
  - If so then a bad K code can not be ignored?
- **What is the longest run without a synchronization sequence?**
- **Does there have to be a synchronization sequence?**
  - If so, is it only at start-up or does it have to be periodic?
- **What size should the FCTs represent?**

SpaceFiber Trade Scenario

- Use 8b10b encoding
- Encode data every 32 bytes (what should value be?) with 8 bit CRC (something better?) to allow earlier detection of error
  - Truncated portion of packet may be less than 32
  - Packet may be less than 32
- Use K codes for Logical characters
- Use 8 bit CRC with K codes and Data values associated with K codes
- Flow control is only for Data characters and not N-characters
- Flow control represents 32 bytes of data
  - About 5% overhead (about same as current standard)

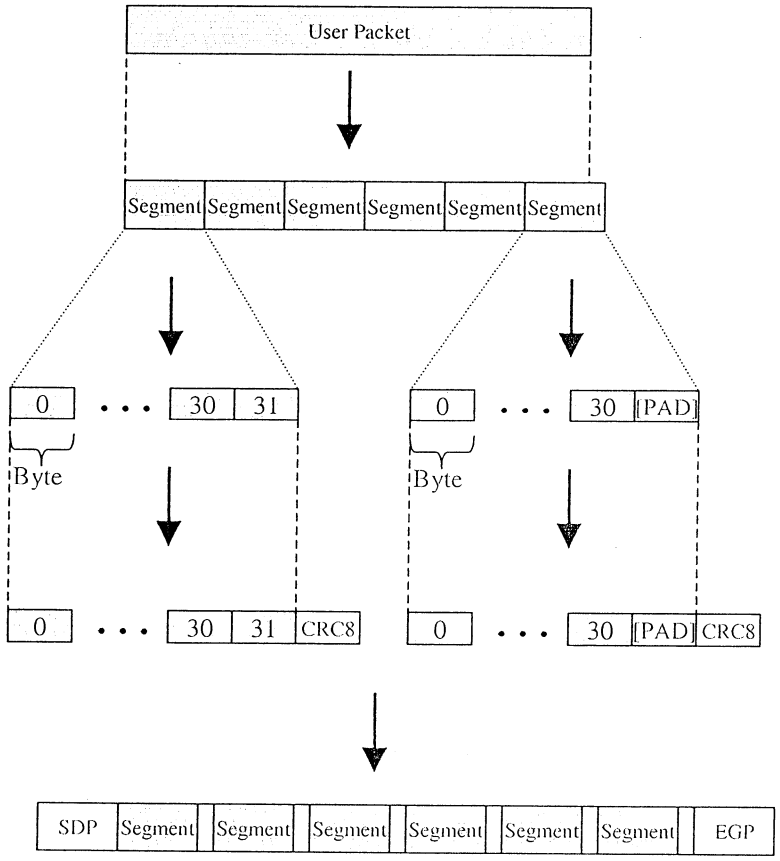
Proto-type

- **Proto-type SpaceFiber on *SerialLite* or *Aurora* protocols**
  - *SerialLite*
    - Altera
  - *Aurora*
    - Xilinx
  - Probably easier to do with *SerialLite*, but *Aurora* quicker path due to users and experience with Xilinx
- **Flight design should be based upon TLK2711 or other Rad-Hard Giga-Bit Per Second (GBPS) Transceiver**
  - Do not want to have IP licensing restrictions (SerialLite or Aurora) so proto-type solutions will have to be migrated over to final solution based upon unique designs

Assumptions

- Full Duplex operation
- Symmetric and asymmetric operation (allows different rates in each direction)
- In-band control signaling using K codes
- Packet protocol (SpW) – No streaming
- Use packet and priority packet types – Priority packets for Time-Code, (FCT/NULL?)
- Nesting (Priority packet within Data packet) for time critical control packet
- Use single Lane
  - Simplifies design by not having complexity of Striping (at Tx) and Bonding (at Rx)
    - See Figure 3 of "*SerialLite* Protocol Overview", Revision 1.0, November 2003
  - Multi-Lane Links may be something to consider for future
    - If bandwidth becomes a limitation
- Packet sizes (Data & Priority): minimum one byte ; no maximum
- 8b/10b physical encoding
- Asynchronous operation – no synchronous operation
  - Necessary for Box-to-Box operation where independent oscillators exist
  - See page 8 of "*SerialLite* Protocol Overview", Revision 1.0, November 2003
- No Lane polarity reversal – LSB transmitted first (less confusion)
- Data field integrity protection (not packet) using CRC8 – better for worm-hole routing
- Payload and Idle scrambling????????????
- No Channel Multiplexing
  - Not supported by SpW standard
  - Once packet starts on wire it must be completed before another packet may start
    - Does not preclude priority packets
      - Used for Time-Code (?)
- Serial*Lite* Flow Control not used
  - Pause commands (XON/XOFF)
- Flow control represents Rx Buffer space, except different value and meaning
  - Represents space for <u>only</u> Data Characters and not N-Char (Data and EOP/EEP Characters)
  - Value represents Rx Buffer space for more than 8 Characters (SpW standard)
    - Suggest 32 Data characters per FCT

SpaceFibre Packet Format



Packet length independent. Still aligned on byte boundaries (same as original SpW).

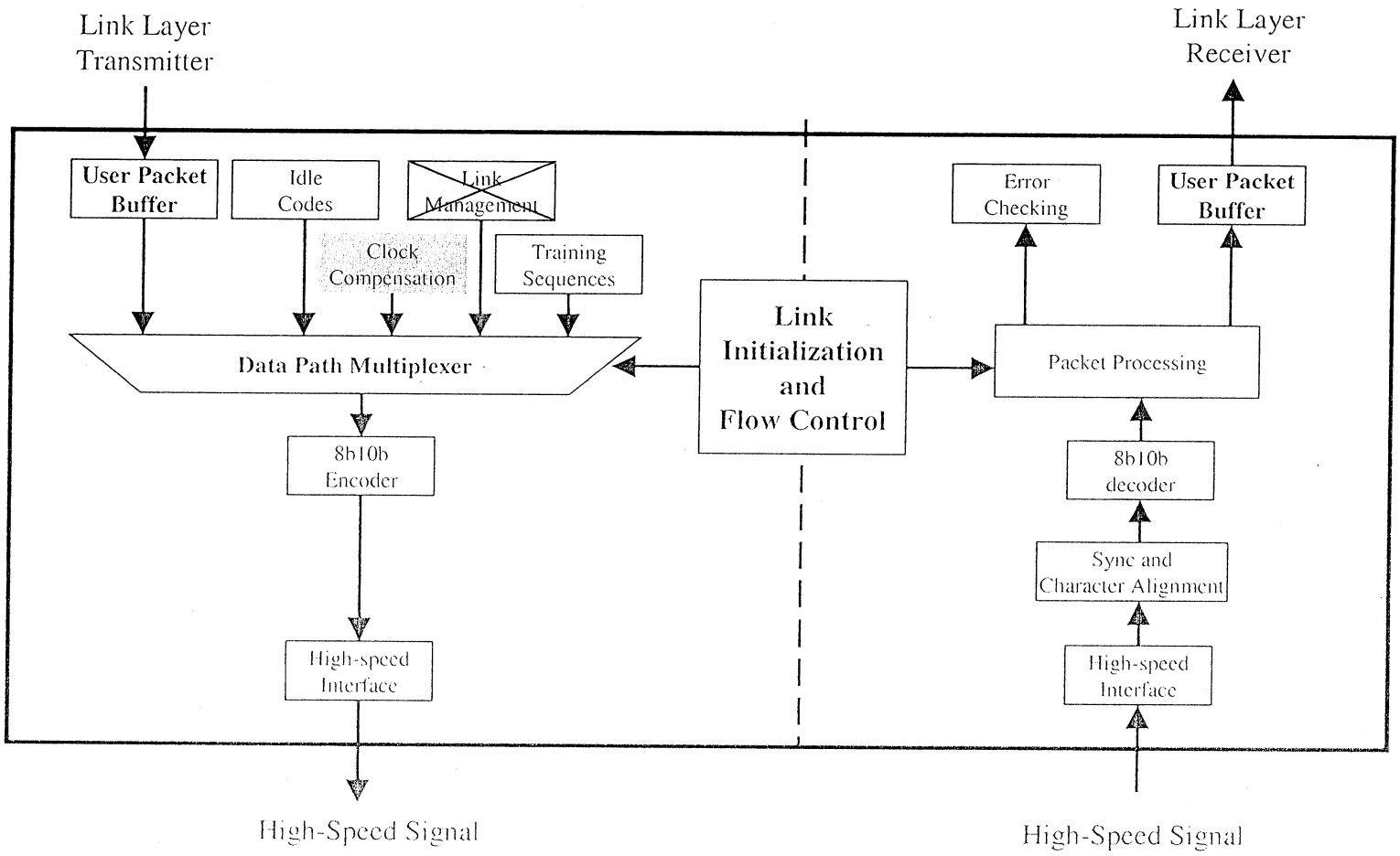
Each segment is 32 bytes (Better number?). What to do if last segment is less than 32?

PAD required if last segment has
an odd # of characters (should we keep data 16 bit aligned?)

CRC8 inserted after every 32 bytes so that
error detection is periodic and not just at end-of-packet.
This feature is useful for wormhole routing to quickly
detect error and prevent network blockage. Thanks Cliff! (should we use checksum instead?)

Comma characters (K characters) Start of Data Packet (SDP) and
End-of Good Packet (EGP) frame the packet
Note: End-of Bad Packet (EBP) may also replace EGP

High Level Data Path*



* Diagram modified from Figure 3 of *SerialLite* Protocol Overview, Revision 1.0, November 2003

Functions

- **Transmit Direction**
  - Serialization of Data
  - 8b/10b encoding (Does this keep track of running disparity in the TLK2711?)
  - Link Initialization
  - Insertion of clock compensation characters for asynchronous operation
  - Idle character conversion
  - Payload and Idle scrambling
- **Receive Direction**
  - Clock recovery
  - Deserialization of data
  - Character alignment using a comma control characters
  - 8b/10b decoding
  - Link Initialization
  - Check for running disparity error and invalid character error
  - Clock tolerance compensation for asynchronous operation
  - Payload and Idle descrambling

Clock Compensation

- **For +/- 100 ppm => Clock Offset Frequency Calculation = 5,000**
  - See "**SerialLite II Protocol Reference Manual**", pg 34 & 35 for definition and explanation
  - Clock Offset Frequency Calculation = 1,000,000/(2 * n)
  - Transmitter must insert one clock compensation sequence,{CC}, once every 5,000 characters (character is byte after conversion to it's 10 bit encoded value)
- **Elastic buffer must be designed after the Transceiver to compensate for the frequency difference between the reference clock and the recovered clock by deleting the {CC}**
  - Rules for {CC} described in "**SerialLite II Protocol Reference Manual**", pg 34 & 35