Prototyping IP over CCSDS for Manned Space Applications

Michael Cheng and Loren Clare

Jet Propulsion Laboratory
California Institute of Technology

Presented at
SpaceOps
Huntsville AL
28 April 2010
Overview

- Introduction
- IP over CCSDS Elements
- Prototypes Developed
- Initial Interoperability Testing
- Spoofing for IP Header Compression
- Real-time Interoperability and IP Header Compression Test Results
- Conclusions
Networking in space is advancing using both Internet Protocol (IP) and Delay/Disruption Tolerant Networking (DTN)

- Focus of this presentation: IP in space
- IP and DTN can be combined (but outside the scope of this briefing)

Advantages IP in space

- Functionalities of networking, addressing, differentiated services, multiplexing, security (IPSec), management
- Ubiquitous application of IP standard in terrestrial data networks promises availability of COTS technology in ground systems and lowered cost for space flight adaptation

CCSDS

- Establish the underlying space communications protocols
- IP over CCSDS Working Group determining required interfaces
  - IP over CCSDS Space Links Draft Red Book CCSDS 702.1-R-3
  - Updated affected Blue Books: Encapsulation Service and Space Link Identifiers

SCaN Network Integration and Engineering (NI&E) project conducted prototyping and testing to advance IP over CCSDS
IP over CCSDS Elements

Space Link Identifiers CCSDS 135.0-B-4
Annex A

Encapsulation Service CCSDS 133.1-B-2

Encapsulation Packet

<table>
<thead>
<tr>
<th>PACKET VERSION NUMBER</th>
<th>PROTOCOL ID</th>
<th>LENGTH OF LENGTH FIELD</th>
<th>USER DEFINED FIELD</th>
<th>PROTOCOL ID EXTENSION</th>
<th>CCSDS DEFINED FIELD</th>
<th>PACKET LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘110’</td>
<td>‘XXX’</td>
<td>‘00’</td>
<td>0 bits</td>
<td>0 octets</td>
<td>0 octets</td>
<td>0 octets</td>
</tr>
<tr>
<td>‘111’</td>
<td>‘XXX’</td>
<td>‘01’</td>
<td>0 bits</td>
<td>0 octets</td>
<td>0 octets</td>
<td>1 octet</td>
</tr>
<tr>
<td>‘111’</td>
<td>‘XXX’</td>
<td>‘10’</td>
<td>4 bits</td>
<td>4 bits</td>
<td>0 octets</td>
<td>2 octets</td>
</tr>
<tr>
<td>‘111’</td>
<td>‘XXX’</td>
<td>‘11’</td>
<td>4 bits</td>
<td>4 bits</td>
<td>2 octets</td>
<td>4 octets</td>
</tr>
</tbody>
</table>

010=IP, 000=idle

Used for prototype testing
Prototypes Developed

- **NI&E SCaN Emulation**
  - software: SCaN Layer 2 Emulator (SL2E)
  - hardware (Verilog): SCaN Emulator (SE)

- **Modified RT Logic T500HDR Telemetry Processor**
  - “RT Logic Layer 2 (L2) Processor”
  - software implementation of IPE/ENCAP/AOS
  - optionally insert IP data via Insert Zone service directly into AOS frames
  - SLE capability

- **NASA Software Defined Radio (SDR) developed by the Goddard Space Flight Center (GSFC) Communications Standards Technology Lab (CSTL)**
  - also fully functional implementation of the IP over CCSDS specifications
  - hybrid platform: IPE/ENCAP/AOS functions in software, physical layer attached synchronization marker and the frame synchronizer in hardware
Prototype Development – NI&E Emulator

<table>
<thead>
<tr>
<th>C3I Spec</th>
<th>High-rate point-to-point</th>
<th>Operational point-to-point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Packet</td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>Data Link Framing</td>
<td>CCSDS ENCAP</td>
<td></td>
</tr>
<tr>
<td>Space Link Framing</td>
<td>CCSDS AOS</td>
<td></td>
</tr>
<tr>
<td>Forward Error Correction</td>
<td>CCSDS 1024-bit rate 1/2 LDPC</td>
<td></td>
</tr>
<tr>
<td>Physical Layer Requirements</td>
<td>Bit Randomization and a 64-bit attached synchronization marker</td>
<td></td>
</tr>
</tbody>
</table>

- Software ENCAP/AOS implementation is called the SCaN Layer 2 Emulator or SL2E.
- Hardware ENCAP/AOS implementation is called the SCaN emulator or SE.
Prototypes Developed – RT Logic T500HDR

- Can be configured in both in IP or SLE mode
- For IP over CCSDS studies, we use the SLE bypass
Initial Interoperability Testing

- Before data is exchanged in real-time, we perform offline compatibility testing.
- An image is partitioned into IP packets and pass through the data path one implementation to create and IPE/ENCAP/AOS stream and saved into a file.
- Another independent implementation takes the saved file and attempts to reconstruct the original image.
- Any difference in the original and the recovered image indicates incompatibility.
- COTS routers and frame relay adapters are used to translate IP header compression information into the correct IP extension byte value for Encapsulation.
Initial Findings

- The three implementations did not agree.
- We documented some of the differences.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Implementations 1, 2, and 3</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCAP header size</td>
<td>1. Used 3-Byte ENCAP header</td>
<td>The allowed ENCAP header sizes are 1, 2, 4, and 8. Can transmit header with fixed sizes but must handle dynamic sizes on receive.</td>
</tr>
<tr>
<td></td>
<td>2. Used variable header size</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Used 4-Byte ENCAP header</td>
<td></td>
</tr>
<tr>
<td>AOS Fill</td>
<td>1. Set M_PDU pointer to 0x7FF and stuff fill with arbitrary pattern</td>
<td>Approaches 2 and 3 are acceptable for transmission. And an implementation should handle both transmit options on the receive side.</td>
</tr>
<tr>
<td></td>
<td>2. Used a four byte ENCAP header and set packet length to fill size</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Used one byte ENCAP header</td>
<td></td>
</tr>
<tr>
<td>AOS Idle</td>
<td>1. VCID in AOS header set to “all ones”</td>
<td>All three are acceptable on transmission and should be all be handled on the receive side.</td>
</tr>
<tr>
<td></td>
<td>2. M_PDU data pointer set to “all ones” – 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. AOS frames filled with 1-byte ENCAP headers</td>
<td></td>
</tr>
<tr>
<td>Contiguous Transmissions</td>
<td>One implementation always starts off a new IP frame with a new ENCAP frame. This approach consumes bandwidth.</td>
<td>If the IP queue is nonempty, the new IP packet should continue to fill the current partially filled ENCAP frame without starting a new frame.</td>
</tr>
<tr>
<td>CRC</td>
<td>CRC polynomials did not match between two implementations.</td>
<td>The CCSDS Green Book has a reference implementation.</td>
</tr>
</tbody>
</table>
# Summary of Independent Implementations

<table>
<thead>
<tr>
<th></th>
<th>Transmit Fixed or Dynamic ENCAP Header size</th>
<th>Idle Fill: SO, MO</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCaN Emulator (hardware)</td>
<td>Fixed (4 octet)</td>
<td>SO</td>
</tr>
<tr>
<td>SCaN Layer 2 Emulator (software)</td>
<td>Fixed (4 octet)</td>
<td>SO</td>
</tr>
<tr>
<td>RT Logic L2 Processor</td>
<td>Dynamic</td>
<td>MO (4 octet), MO</td>
</tr>
<tr>
<td>CSTL SDR</td>
<td>Dynamic</td>
<td>SO, MO</td>
</tr>
</tbody>
</table>

SO = Single Octet idle fill (PID = 000, LoL = 00)
MO = Multiple Octet, Zero or more idle fill bytes (no encapsulated data)
IP Header Compression (IPHC)

- The redefined space link protocol identifier and the IP extension (IPE) byte were established to accommodate IPHC on a simplex link.

- Commercial Off The Shelf (COTS) routers are expected to perform IPHC according to RFC 2507 and 2508. However, COTS implementations assume a bi-directional link to exchange IPHC compression state between the compressor and decompressor.

- To get around the handshaking requirement, we use an adapter connected to the local COTS router to spoof negotiation messages with the remote router.

- For each router generated request, the local adapter generates a reply.

- We demonstrated IPHC on a simplex link using our approach and showed that the link with IPHC is robust under dynamic link conditions with packet drops.
COTS routers support IPHC only on a low data rate serial link such as Frame Relay (FR).

This state diagram lists the exchange of FRF.20 negotiation messages that are required to put both the compressor and decompressor into IPHC mode.

Without a duplex link to exchange the states, the compression configuration parameters can be pre-negotiated and the COTS routers be put into IPHC mode by spoofing the replies for each request using an FRF.20 adapter connected to the local router.

Tables of compression options and descriptions of the header flags are found in the FRF.20 agreement.
Two spoofing options are available: no negotiation and one-way negotiation.

1. In the no negotiation startup sequence, none of the FRF.20 control messages are passed between the routers. In ground processing, the FRF.20 adapter puts the ground router into IPHC mode using the sequence described in the previous slide. The same is repeated for space processing.

2. In one-way negotiation, forward negotiation is allowed on the simplex link to synchronize the decompressor IPHC configuration to that of the compressors. All negotiation messages from the compressor to the decompressor are sent on the simplex link and mapped to the IPE value 0x23h. To protect against packet drops, forward negotiation messages are sent periodically on the link.
Real-time Interoperability and IPHC Testing

- Have successfully demonstrated IPHC on a simplex link using this test configuration.
- Prototype A processes the incoming MPoFR frames using the new IPE recommendation, extracting information from the data link and generating the “shim byte” as described in the updated Space Link Identifiers.
- Encapsulated packets are then passed to for AOS space data link processing, forming a conventional continuous stream of AOS frames.
- This stream is passed through a channel to the other Prototype, where each of these functions is processed in reverse order.
- The intervening channel may be error-free, or intentionally dropped AOS frames may be induced in order to test the robustness of the prototypes.
- Prototypes were placed in remote locations therefore testing “internet” connectivity.
- Observed that on average 40-byte IP headers are compressed down to 5 bytes.
### Real-time Interoperability and IPHC Test Results

<table>
<thead>
<tr>
<th>Prototype pairing (2-way in every case)</th>
<th>SL2E-SDR</th>
<th>SL2E-RT Logic</th>
<th>SL2E-SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable IP sizes</td>
<td>100B, 1000B, 1700B, 8000B</td>
<td>Random over 50B to 1500B</td>
<td>Random over 50B to 1500B</td>
</tr>
<tr>
<td>Link Rate</td>
<td>128 kbps</td>
<td>192 kbps and 24 kbps</td>
<td>192 kbps</td>
</tr>
<tr>
<td>Offered Loads</td>
<td>65 kbps and 120 kbps</td>
<td>~180 kbps and ~20 kbps respectively</td>
<td>~180 kbps</td>
</tr>
</tbody>
</table>

1. Variable IP Sizes: tests were run with the offered IP traffic having different packet sizes first with uniform 100B, 1000B, 1700B, 8000B IP packet sizes, then with varying sizes uniformly distributed between 15B and 1500B.

2. Link rates tested were 24 kbps, 128 kbps, and 192 kbps.

3. Traffic offered during the tests was less than the link rates, thereby necessarily creating numerous conditions where idle fill was required to be generated by the Encapsulation Service. The different idle fill methods that the different prototypes implement were tested.
4. Perfect channel and AOS frame drops
   - Tests were conducted under conditions where no channel errors occurred
   - And conducted in which AOS frame drops were induced. One, two, or ten contiguous AOS frames were dropped and each drop occurs every 10, 100, and 1000 good AOS frames. Test is successful only if the error condition persisted through the minimum number of affected received IP packets.

5. Variable length Encap packet header on transmit and receive
   Interoperability between dynamic header size and fixed 4-byte ENCAP header was verified to be of no problem.

<table>
<thead>
<tr>
<th>Prototype pairing (2-way in every case)</th>
<th>SL2E-SDR</th>
<th>SL2E-RT Logic</th>
<th>SL2E-SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect Channel</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AOS Frame Drops</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Transmit variable length Encap Header</td>
<td>SDR</td>
<td>RT Logic</td>
<td>none</td>
</tr>
<tr>
<td>Receive variable sized Encap Headers</td>
<td>SL2E</td>
<td>SL2E</td>
<td>none</td>
</tr>
</tbody>
</table>
6. **IPE values covered:** IPE involves the processing that includes a simple fixed one-to-one mapping of information derived from the interfaced data link layer to the shim byte, according [Table A-1 of Annex 1 of Space Link Identifiers](#). Tests were conducted on a number of these different IPHC settings, and confirmed to interoperate successfully.

7. **Mixture of compression types:** In order to further test the robustness of IPE, SL2E – SE tests were conducted in which multiple simultaneous IP source-destination flows were offered, such that this flows were configured to have different IPHC instantiations.

8. **Support of ENCAP header fill:** All prototypes accept Encap Header PID=000 (fill) in addition to PID=010 (IPE), as per Space Link Identifiers. All prototypes interoperate.
Conclusions

- IP, the ubiquitous networking protocol, can operate over internationally recognized standard CCSDS space data links
  - Both IPv4 and IPv6
- General subprotocol demultiplexing framework enables IP header compression information to be passed between adjacent IP routers through the space link
- Several IP over CCSDS-capable prototypes were implemented for AOS data link
  - Hardware and software implementations
  - In-house and vendor developments, multiple NASA centers
- Interoperability tests among prototypes
  - Revealed inconsistencies in interpretations of original specifications
  - Corrections resulted in accurate final standards documents
- Performed validation testing of IP header compression for simplex space link
  - Developed spoofing mechanism to allow use of COTS routers
  - Proved robustness in degradation from duplex to simplex operation
- These results will enable the advancement of a common, internationally agreed-upon means for IP-based networking in space