Design for a Satellite Communication Link in a Space Based Internet Emulation System

Pooja Wagh
(Masters Thesis Defense)
July 10, 2001

Defense Committee:
Dr. Gary Minden (Chair)
Dr. Joseph Evans
Dr. Victor Frost
Organization

- Introduction
- Problem Definition
- Proposed Solution - SBI Approach
- SBI Emulation System - An Overview
- Emulating the Satellite Communication Link
  - Virtual Ethernet (VETH)
  - Constant Bit Rate Control
  - Link Propagation Delays
- Summary
- Future Work
Introduction

• Satellite communication is based on *Line of Sight* principle

• The Earth Observation Satellites (EOS) communicate with GEO relay satellites such as TDRSS to transmit their information to the Earth stations

• Each EOS satellite has a fixed time slot to transmit its data to TDRSS
Problem Definition

- EOS satellites need to have high data rate and high capacity recorders to store data onboard

- Communications systems on satellite is satellite-specific

- Each satellite needs to have its communication frequency, protocol and command structure

- This approach leads to incompatible and non-reusable communication components
Proposed Solution: SBI Approach

• Create a *Space Based Internet (SBI)* between the satellites and ground stations

• Each satellite would be capable of switching traffic between other satellites and ground stations

• Communication protocol would be Internet Protocol (IP)

• IP eliminates the need for any satellite-specific communications systems or any specialized ground station equipment
SBI Approach

• Apply wireless Ethernet network technology to near-Earth satellite systems

• Develop innovative topology and routing algorithms suitable for satellite systems

• SBI network software contains standard modules that can be deployed to minimize the individual satellite cost

• Evaluate and test the SBI software on an emulation system that will model a satellite system
SBI Emulation Approach

- Execution of actual scenarios on the SBI system hardware with the nodes executing SBI network software

- Emulate communication channels on the satellites and ground stations

- Emulate the features of satellite transmission link on the Ethernet connections between the SBI nodes

- Model the data traffic and measure the network performance using KU NetSpec system
SBI Emulation System

• SBI Nodes
  • Emulation Nodes representing satellites and ground stations

• Emulation Manager
  • Controls and monitors the entire emulation scenario

• Emulation Network
  • Data Network - for data transfer between the nodes
  • Management Network - Control and status communication between the Emulation Manager and the nodes
SBI Node Types

• Standard capacity Data Observation and Routing satellites

• High Capacity Routing Satellites

• Ground Stations for collecting and routing data

• Operations Facility
  • Acts as a centralized ground station
  • Responsible for Data Instrument Scheduling
  • Establishes network topology and routing protocols
Emulation Manager

- Reads the configuration files and maintains logs of the network
- Creates and deletes connections between the nodes
- Responsible for interactive control of the scenario
- The Emulation Manager software on the nodes emulates the communication links between the nodes
Emulation Networks

- **Data Network**
  - Emulates the communication channels for data transfer between the SBI nodes
  - Utilizes managed Ethernet switches for switching data traffic

- **Management Network**
  - Control commands from the Emulation Manager to the nodes
  - Status monitoring information from the nodes to the Emulation Manager
  - Scheduling commands and routing information from the Operations Node to the respective nodes
  - Utilizes unmanaged Ethernet switches
Node Emulation Software

- **Node Controller**
  - receives Control commands
  - transmits Node Status information
  - *Comm Control* controls communication emulation

- **Operations Interface**
  - information from Operations Node

- **Comm Emulation**
  - models space communication on the connections between the nodes
Communication Emulation

- Emulate multiple Virtual Ethernet (VETH) interfaces to facilitate multiple point-point connections on a single node

- A point-point connection represents a communication link

- Following features of the communication link are emulated:
  - CBR traffic by performing bandwidth limitation on the connection
  - Simulate link propagation delays occurring on actual satellite links
Virtual Ethernet (VETH)
Virtual Ethernet (VETH)

- Models multiple communication channels over a single physical Ethernet interface

- Each VETH device represents a communication channel on the SBI nodes

- The VETH is a layer between the IP layer and the physical Ethernet layer in the Linux Kernel

- VETH devices are controlled and configured by the *veth controller* module, a part of the emulation software
VETH : Network Layers and Controls

• **Veth Control** operations:
  • Create a VETH device
  • Destroy a VETH device
  • List all the devices created

• **ifconfig** command operations:
  • Setting the device IP address
  • Netmask
  • MTU (Maximum Transfer Unit)
  • Hardware Address
VETH : Specifications

- Each VETH device has:
  - IP address
  - 6-byte unique MAC address

- The first 3 bytes of VETH MAC address is the ITTC KU vendor Identifier:
  - useful for de-multiplexing packets
  - ITTC Vendor Id: 00:04:86
VETH : Create

- A VETH device can be created by an `ioctl()` system call from the `veth controller` program.

- Every VETH device has an instance of `struct device`, which is a “C” structure to represent a device in Linux Kernel.

- The function pointers of `struct device` such as `open`, `init`, `transmit`, etc point to appropriate functions in the VETH layer.
Sending Data on VETH device

- Data Traffic from the IP layer to the VETH device
- The VETH device provides CBR control
- Delays the packets to simulate the link propagation delay
- Transmits the packets to the physical Ethernet device

Packet Flow through the VETH device

```plaintext
vethDev->enqueue(skb, veth_dev-queue)

Dev_queue_xmit(skb)

vethDev->dequeue(skb, veth_dev->queue)

vethDev->hard_start_xmit(skb, veth_dev)

Physical_dev->hard_start_xmit(skb, physical_dev)
```
Receiving Data on VETH device

- Packets are checked for the destination MAC address in the Ethernet header

- If the MAC address corresponds to a VETH device, then the packets are routed to the receive function of the VETH device
Destroying the VETH device

- A VETH device can be destroyed by an `ioctl()` system call from the `veth controller` program
- The kernel memory occupied by the device structures has to be freed
Constant Bit Rate (CBR) control
Requirement for CBR control

• The VETH devices are capable of handling different types of satellite traffic
  • continuous - Telemetry Data
  • periodic - Observational Data such as Land-based measurements.

• Nodes representing Router satellites need to have high-capacity dedicated links

• Nodes emulating Observational satellites need to have some links dedicated to data collection and some links for routing data
Requirements

• Each communication link should have a continuous dedicated bandwidth

• The total capacity of the SBI node should be efficiently utilized to reserve bandwidth for each link

• Utilizing Token Bucket Filter (TBF) queuing disciplines to limit the rate on the link

• TBF queuing forms a part of Traffic Control mechanisms in Linux
TBF Queuing

- TBF generates tokens at a specific rate

- The tokens are filled into a bucket (buffer). The buffer limit specifies the maximum number of tokens it can store

- The number of bytes that can be de-queued from the device queue is equivalent to the number of tokens in the bucket

- The data rate on the link is limited to the rate at which the tokens are generated
Traffic Controller “tc”

- User level program to create and associate queues with the output devices
- Interacts with the kernel through *netlink sockets*

“tc” usage for setting TBF queue:

```
tc qdisc add dev <device name> handle <handle #> root tbf rate <link rate>
burst <bytes> limit <bytes>
```

- *rate*: value at which the link rate should be limited
- *burst*: maximum number of tokens the buffer can hold at one time
- *limit*: burst size + queue size which places the packets in case the tokens are exhausted
CBR control for SBI

• The TBF queue has to be created on the VETH device

• Parameters passed by the Emulation Manager to the nodes:
  • Virtual device on which the TBF queue has to be created
  • data rate for that virtual device

• The *Comm Control* module has the “tc” utility, sets up the TBF queue on the specified device

• The Central Operations Node monitors the bandwidth utilization for each SBI node
Link Propagation Delays
Link Propagation Delay

- Large distances between the satellites and ground stations
- Link propagation delays can be vary from 10 - 250 ms (one way) between satellites
- Large amount of data is *in-flight* on the communication link
- Propagation delay on a link can vary due to satellites being in motion
Delay Analysis

- Find out the delay variations with respect to time for some of the actual satellite links

- Using Satellite Tool Kit (STK) to obtain a report of the delay variations

- The following types of links were examined:
  - LEO-MEO, LEO-GEO, LEO-Ground Station
  - MEO-GEO, MEO-Ground Station, GEO-Ground Station
STK Report

- STK gives the following reports:
  - Duration Time (minutes) for each access for a link
  - Delay Variations with respect to time for each access duration

- Example: Access Report for a MEO-Ground Station link

<table>
<thead>
<tr>
<th>Access</th>
<th>Start Time (Epoch Minutes)</th>
<th>Stop Time (Epoch Minutes)</th>
<th>Duration (Minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>121.344</td>
<td>159.444</td>
<td>38.1</td>
</tr>
<tr>
<td>2</td>
<td>276.7916</td>
<td>314.3661</td>
<td>37.574</td>
</tr>
<tr>
<td>3</td>
<td>626.3372</td>
<td>646.9397</td>
<td>20.602</td>
</tr>
<tr>
<td>4</td>
<td>776.8671</td>
<td>817.618</td>
<td>40.751</td>
</tr>
<tr>
<td>5</td>
<td>932.7947</td>
<td>971.2767</td>
<td>38.482</td>
</tr>
</tbody>
</table>
STK Report

- A graph showing the delay variations for Access 1, 4 and 5

- Delay varies from 24ms to 12ms

- Delay variations per minute: 1ms
Delay Algorithm

- Packets are queued on the VETH device before transmission on the physical layer

- Queue size should be equal to number of bytes that can be in-flight at one time on the link
  - Number of packets in-flight = Propagation delay / Transmission delay
  - Number of bytes in-flight = packets * (average packet size in bytes)

- The packets queued are delayed for the required amount of time and then de-queued for transmission
Delay Algorithm

• The queue size has to change with the delay variations

• Delay variations are 1-3 ms per minute

• The queue size should change by +/- 25 packets for an average packet size of 1500 bytes and a 100 Mbps link
Simulating the delay

- Packets coming on the VETH device are time stamped and placed in the *veth queue*

- While de-queuing, the time stamp on the packet is compared with the current time

- If time difference equals the delay value, then the packet is de-queued and sent on the physical device

- Otherwise, the packet is queued at the head of the queue
Simulating the delay

• The following parameters have to be given to the SBI Nodes:
  • The name of virtual device
  • link rate and average packet size for calculating the transmission delay
  • STK report giving the delay values with respect to time

• The delay value along with the queue size is passed to the VETH device through an ioctl () system call
Summary

- This work presents a design for emulating a Satellite Communication Link

- Multiple communication channels on the satellites can be emulated by creating multiple VETH devices on a single node

- Traffic control mechanisms such as TBF queuing can be utilized to model CBR traffic on the communication links

- Propagation delays can be simulated on the emulation link to model actual space communication
Future Work

- Modeling Bit Error Rate on the link
  - introducing bit errors in packets to achieve the specified bit error rate

- Implementation of the proposed design
Questions?