



Simple Protocol for Heterogeneous Embedded Communication Networks

Martin Zenzes

martin.zenzes@dfki.de

February 25, 2016

Contents

Background

Protocol

Measurements

Contents



Background

Protocol

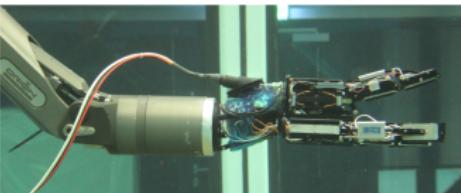
Measurements

Building robotic systems

DFKI RIC: *Robotics Innovation Center*

- ▶ Education:  Universität Bremen
- ▶ Heterogeneous, versatile specimens
- ▶ Off-the-shelf & Custom designs
- ▶ Components reused between generations

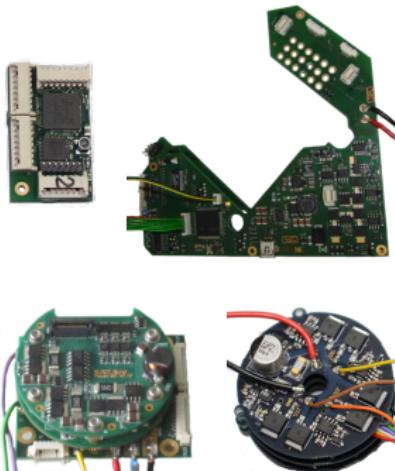
Me: Embedded Engineer since 2010
...lots of Linux and low level



Heterogeneous devices

Electronics distributed, in confined volumes:

- ▶ Actuators: BLDC, Relay, LED
- ▶ Sensors: ForceTorque, IMU, Position
- ▶ Silicon: FPGA, Microcontroller, CPU



Communication:

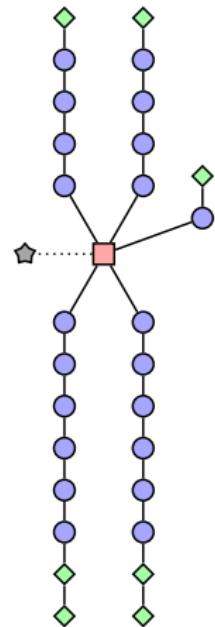
- ▶ *Unreliable* UART / byte stream
- ▶ Tree created by point-to-point connections
- ▶ Different communication speeds



Network Topology



Charlie [1]



■ Embedded PC ★ Control PC

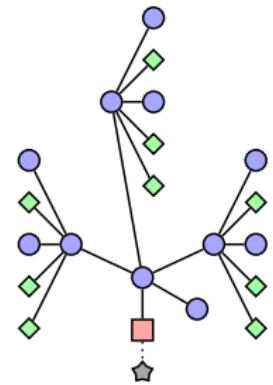
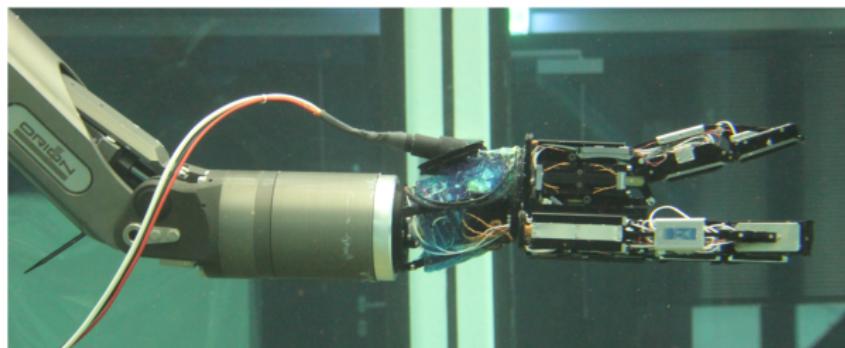
◆ Microcontroller ● FPGA

..... WiFi

——— UART

Network Topology

Gripper [2]



■ Embedded PC ★ Control PC

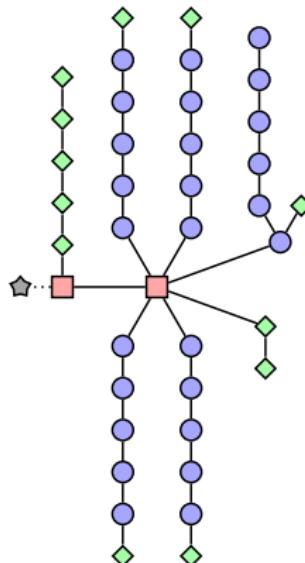
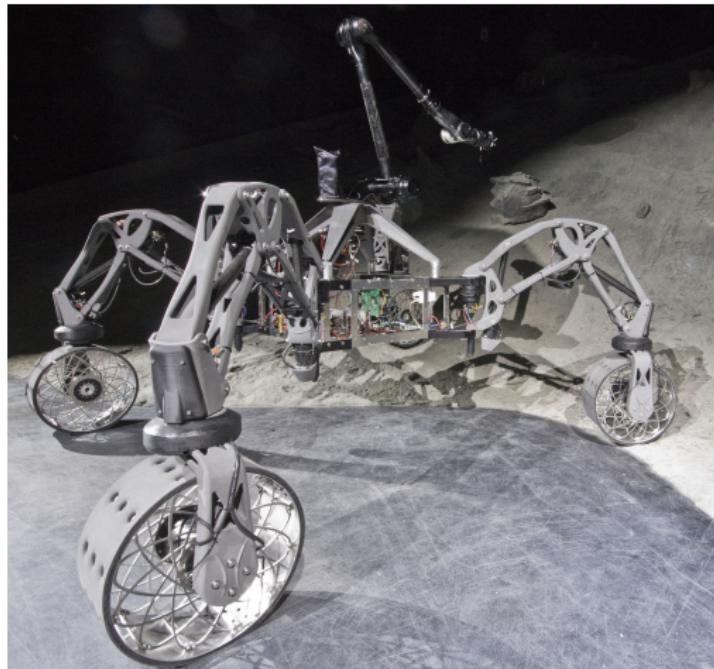
◆ Microcontroller ● FPGA

..... WiFi

——— UART

Network Topology

SherpaTT [3]



■ Embedded PC ★ Control PC

◆ Microcontroller ● FPGA

..... WiFi

——— UART

To be applicable from ATmega88 to Zynq-7000:

- ▶ Reuse old firmware in new systems (Stable interface)
- ▶ Distributed control loops (Decentral communication)
- ▶ System wide emergency messages (Broadcasts)
- ▶ Keep bad transmission in mind (Checksum, Counter)
- ▶ Flexible implementation (VHDL and C)



Contents

Background

Protocol

Measurements

HDLC [4]: Encapsulate a chunk of *data* as *datagram*

- Escaping: Replace all occurrences of reserved bytes (0x7e and 0x7d)
- Framing: Surround datagram with reserved byte (0x7e)

Unencoded data:

0xa0 0x13 0x7e 0x3f 0x7d

Encoded data:

0x7e 0xa0 0x13 0x7d 0x5e 0x3f 0x7d 0x5d 0x7e

HDLC [4]: Encapsulate a chunk of *data* as *datagram*

- Escaping: Replace all occurrences of reserved bytes (0x7e and 0x7d)
- Framing: Surround datagram with reserved byte (0x7e)

Unencoded data:

0xa0 0x13 0x7e 0x3f 0x7d

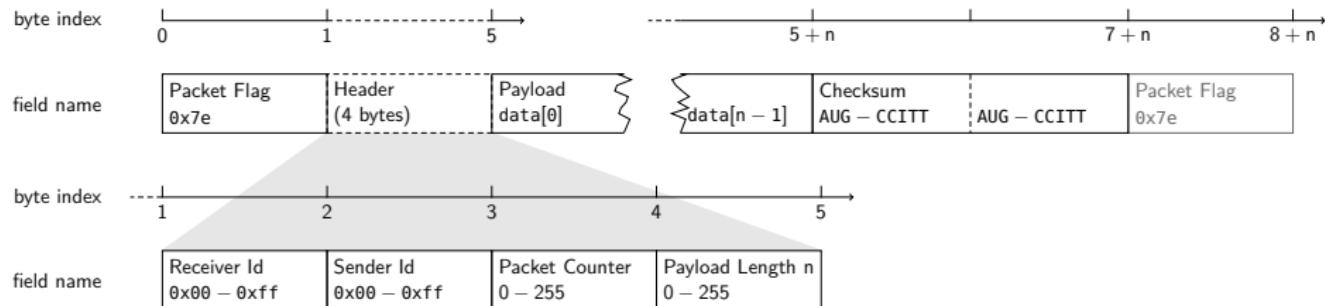
Encoded data:

0x7e 0xa0 0x13 0x7d 0x5e 0x3f 0x7d 0x5d 0x7e

to be reversed by receiver

Packet Format

- ▶ 8 bit SenderId, ReceiverId
- ▶ Packet Counter (detect lost packages)
- ▶ Small maximum payload (reduced buffering)
- ▶ AUG – CCITT to guard transmission [4]



- ▶ Do one thing and do it well
- ▶ Reusing implementations via separation of concerns
- ▶ Be able to exchange layers

Layer	Data Unit	Function	Examples
4. Transport	Segments	Transmission of data segments between points on a network, including segmentation and acknowledgement	TCP, UDP
3. Network	Packet, Datagram	Structuring and managing multi node network, including addressing and routing	AppleTalk, IPv4, IPv6 NDLCom
2. Data Link	Frame	Reliable transmission of data frames between two nodes connected by physical layer	PPP, IEEE 802.2, L2TP HDLC [4]
1. Physical	Bit	Transmission and reception of raw bit stream over physical medium	DSL, USB

Forwarding strategies

Incoming message to outgoing interface

Cut Through: Retransmit packet as soon as ReceiverId known

- + Reduced delay in long chains
- Broken packages not filtered
- Timeout management needed
- No crossing of bandwidth domains

Store & Forward: Decode whole packet before retransmission

- + Simpler implementation
- Increased usage of resources

Forwarding strategies

Incoming message to outgoing interface

Cut Through: Retransmit packet as soon as ReceiverId known

- + Reduced delay in long chains
- Broken packages not filtered
- Timeout management needed
- No crossing of bandwidth domains

Store & Forward: Decode whole packet before retransmission

- + Simpler implementation
- Increased usage of resources

implemented both

Contents



Background

Protocol

Measurements

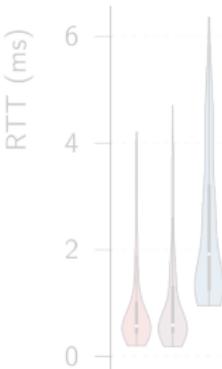
Round Trip Time (RTT)

Timespan from request to response (many many times):

- ▶ Observed online, during full operation!
- ▶ How much Jitter and Delay is created?
- ▶ How do devices and algorithms differentiate?
- ▶ How usable is the transmission?

Violin plot:

- ▶ Similar to box plot, also shows probability density
- ▶ One plot per probed device, grouped per chain



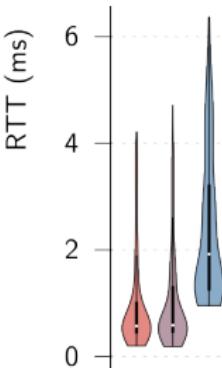
Round Trip Time (RTT)

Timespan from request to response (many many times):

- ▶ Observed online, during full operation!
- ▶ How much Jitter and Delay is created?
- ▶ How do devices and algorithms differentiate?
- ▶ How usable is the transmission?

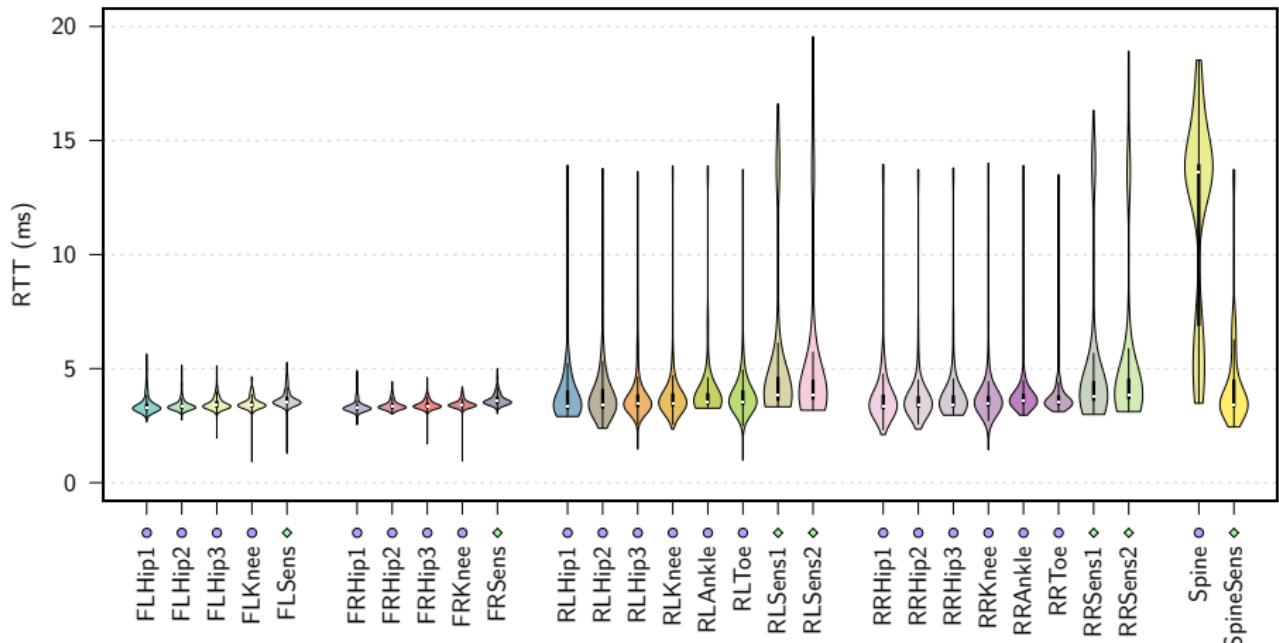
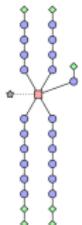
Violin plot:

- ▶ Similar to box plot, also shows probability density
- ▶ One plot per probed device, grouped per chain



RTT in real life

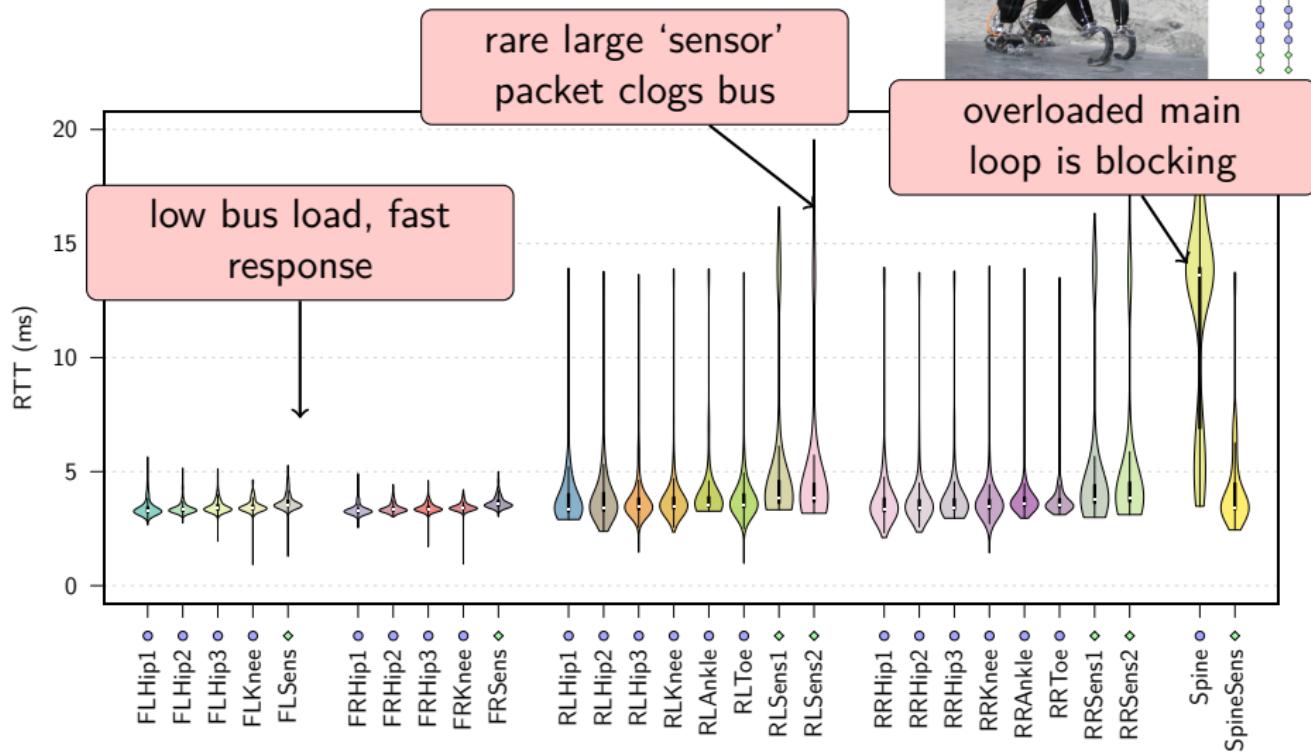
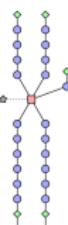
Charlie [1]



◆ Microcontroller ● FPGA

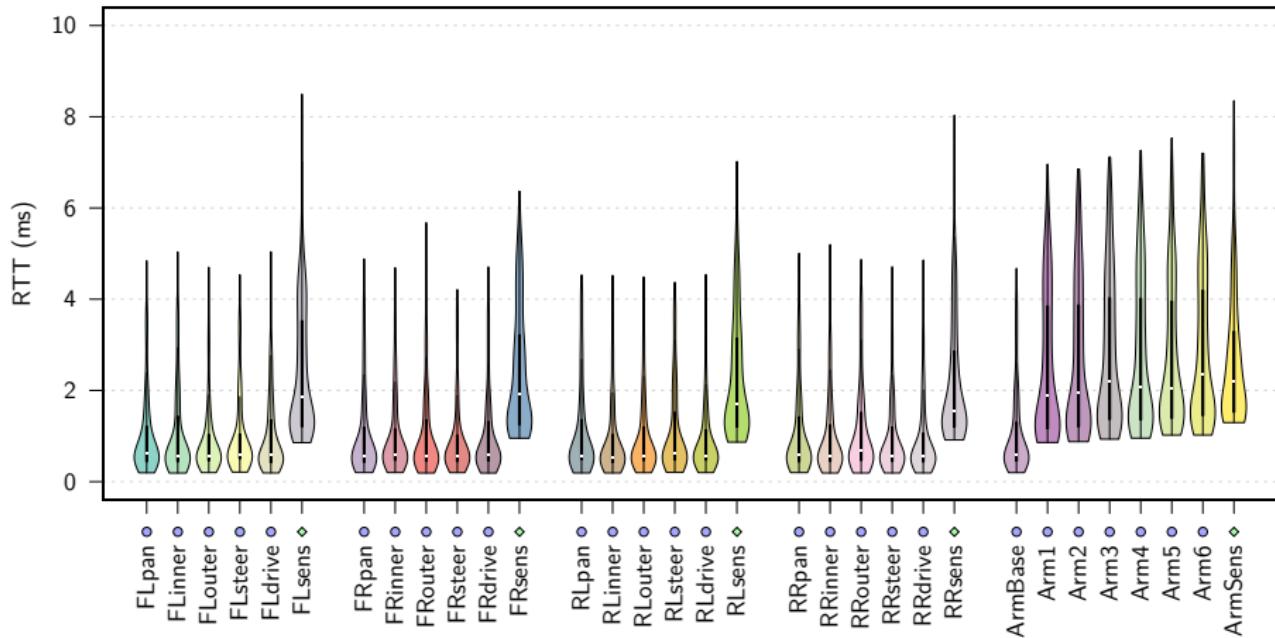
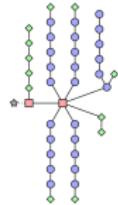
RTT in real life

Charlie [1]



RTT in real life

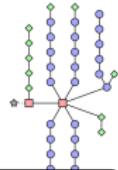
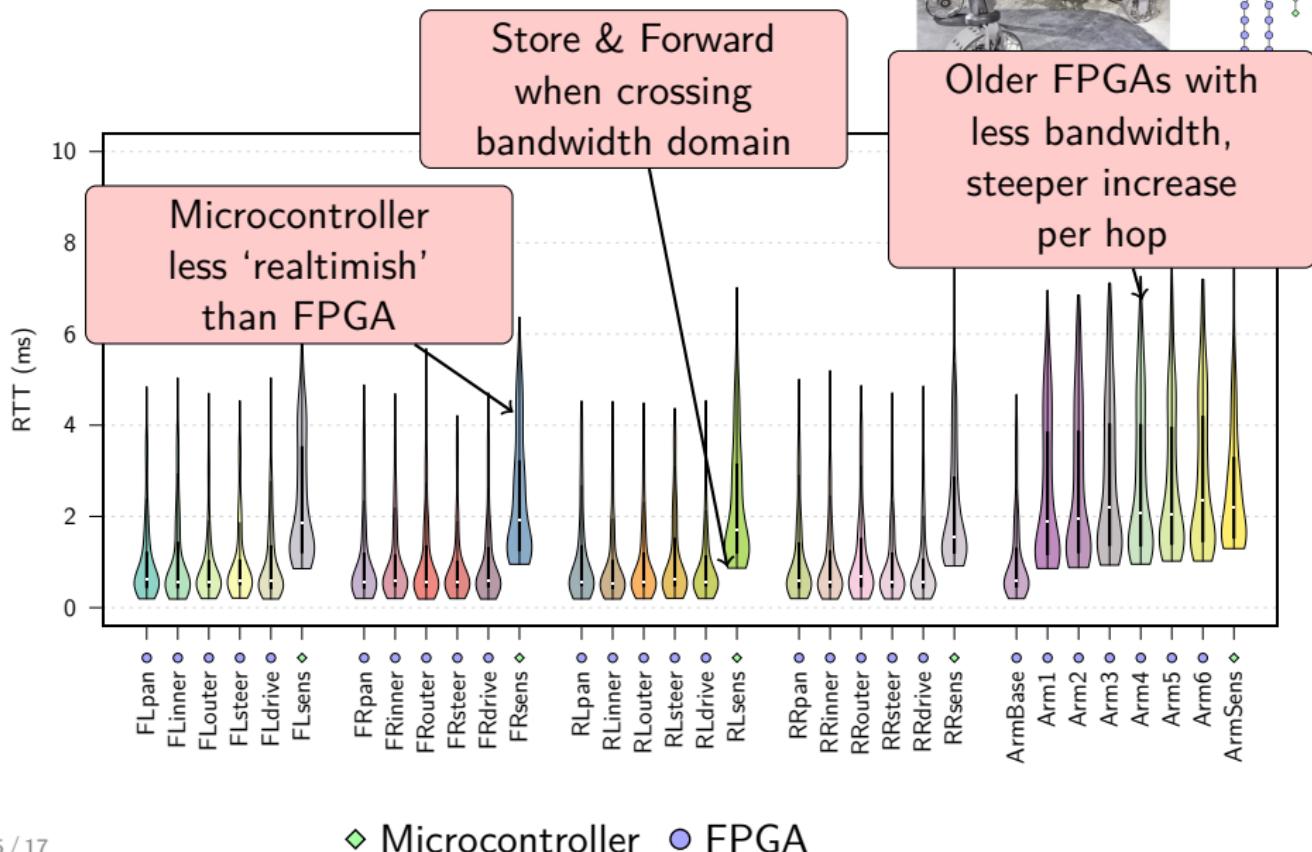
SherpaTT [3]



◆ Microcontroller ● FPGA

RTT in real life

SherpaTT [3]



...The Next Steps:

- ▶ Exploitation: Keep-alive, Clock Sync, Tree Detection
- ▶ Tooling: Online visualization of monitored RTT
- ▶ Licensing: Release as Open Source

References



-  D. Kuehn, M. Schilling, T. Stark, M. Zenzes, and F. Kirchner.
System Design and Field Testing of the Hominid Robot Charlie.
submitted to Journal of Field Robotics, July 2015.
-  Peter Kampmann and Frank Kirchner.
Towards a fine-manipulation system with tactile feedback for deep-sea environments.
Robotics and Autonomous Systems, 67:115–121, 2015.
-  Florian Cordes, Christian Oekermann, Ajish Babu, Daniel Kuehn, Tobias Stark, and Frank Kirchner.
An Active Suspension System for a Planetary Rover.
In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2014)*, June 17-19, Montreal, Canada. o.A., 6 2014.
-  ISO 13239:2002.
Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures.
[Online; accessed 12-January-2016].