

Lab 9 Custom Network

Group: Route-On

Group Member:

Hao Zhang, Vidhi Goel, Venkata Srinivas Chowdary Reddy

Oct 10th, 2015

1. Backward Engineer to find the static bits

An Ethernet frame is preceded by a preamble and start frame delimiter (SFD), which are both part of the Ethernet packet at the physical layer. The static bits present in the packet are the Preamble (7 Octets) and SFD (Start Frame Delimiter/1 Octet). These fields are present before the Ethernet Header. The Preamble allows devices on the network to easily synchronize their receiver clocks. The SFD marks the end of the preamble, which is the first field of an Ethernet packet, and indicating the beginning of the Ethernet frame.

We performed testing using different packet sizes to send data from sender to receiver using OSI model. We used libpcap to see the raw packets. Preamble and start frame delimiter are not displayed by packet sniffing software because these bits are stripped away at OSI layer 1 by the network interface controller (NIC) before being passed on to the OSI layer 2, which is where packet sniffers collect their data. Below is the result of our experiment:

Sender

```
generating packets of 512 bytes...
DONE
generating packets of 1499 bytes...
DONE
generating packets of 1500 bytes...
DONE
generating packets of 1501 bytes...
DONE
generating packets of 1513 bytes...
DONE
generating packets of 1514 bytes...
DONE
```

Receiver

```
===== here is a packet of size 512 =====
68 65 72 65 20 69 73 20 61 20 72 61 6E 64 6F 6D   here is a random
20 70 61 63 6B 65 74 20 77 69 74 68 20 73 69 7A   packet with siz
65 20 35 31 32 2A 63 6D 10 AE 8A 31 F6 6D FD 06   e 512*cm...1.m..
38 FD E0 0E 43 E0 57 C3 01 30 80 76 AE 08 A5 8D   8...C.W..0v....
C5 A8 12 5D FF 75 CA 0F 23 54 41 1A C2 3E 20 FA   ...].u...#TA..> .
3C 01 09 7F E1 60 42 E3 91 C3 59 3F CB FF CD 90   <..`B...Y?....
A7 DF EE A6 55 B8 B5 78 0D F6 92 CF 35 B3 C9 71   ....U..x....5..q
B4 D2 F0 95 33 32 78 C4 F5 D2 03 C0 D1 D0 51 78   ....32x.....Qx
B0 3F 1E 05 F7 D3 7D 04 CA 10 D3 FF C3 9D 70 77   .?....}.....pw
6F 60 0C A2 92 85 66 88 57 6A 48 28 3A 99 A0 EA   o`....f.WjH(:...
D8 BE EF D0 91 6D D4 5B 7D A8 5A 40 45 CA B7 B4   .....m.[}.Z@E...
```

The sender packet begins with line “here is a random packet with size 512” and then followed by random data. From the above tests, we can see that whatever data we are sending to the ethernet device, the same data is being received at the receiver right from the beginning. (check the size and first line). This proves that when the data is sniffed at kernel, the Preamble and SFD are already stripped off by the NIC.

2. Custom Packet Format

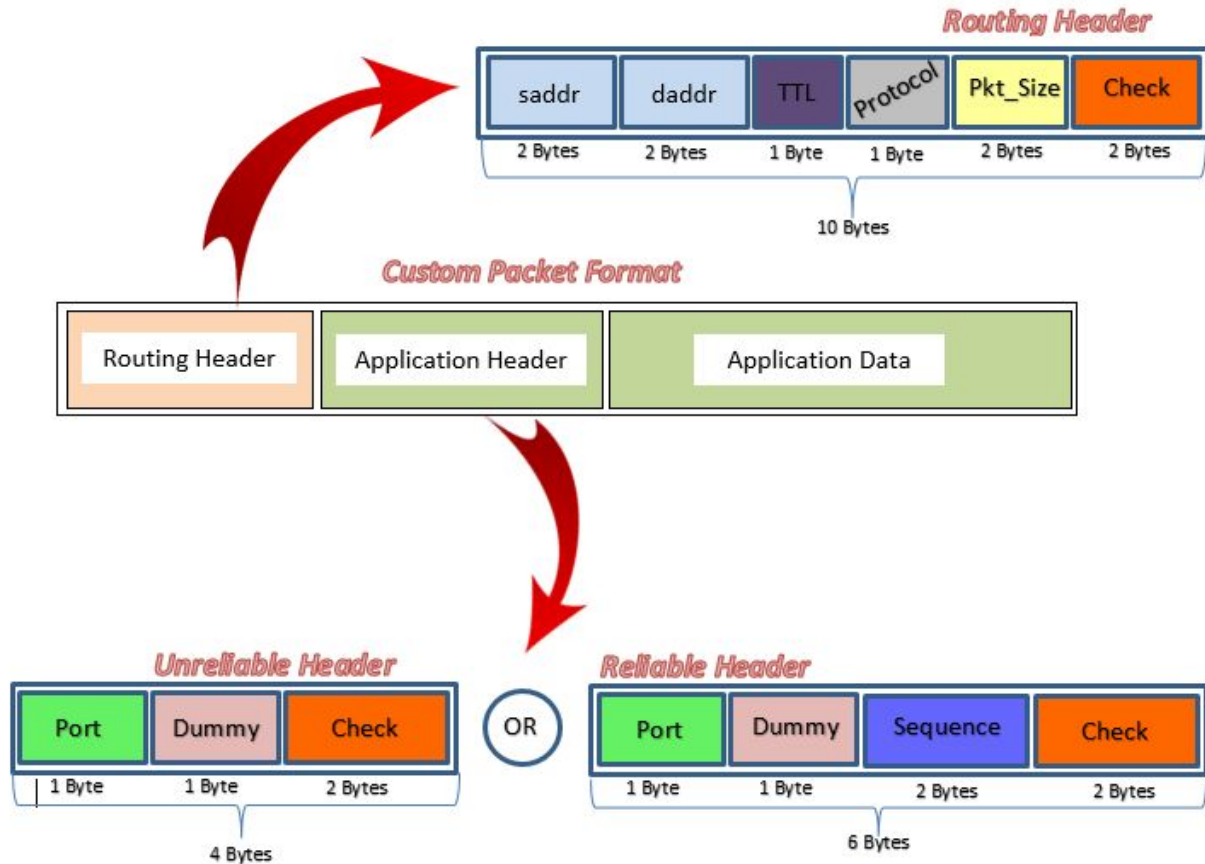
The traditional OSI model is designed to be universal. A data-link frame consists of the following parts: data link header (e.g. Ethernet header), network header (e.g. IP header), transport header (e.g. TCP header, UDP header, ICMP header) and actual data sent by the application. When the OSI model is applied to small-mid scaled data center / cloud, in which only specific functionalities are performed, there are a lot of redundancy in these headers. We try to simplify the OSI model and create our own packet format and routing mechanisms.

To make things simple, we try to extract the most essential information that make it possible to send packets among different applications. These essential information are:

- a. An identifier of host
- b. An identifier of application
- c. An identifier of application instance

Identifier of host makes it clear for the application to know which host the target application sits on. Identifier of application prevents application from receiving unnecessary information that is not target to itself. As a single application can have multiple instances, each instance should not interfere with each other in sense of data transmission, and therefore we need an identifier of application instance.

Here is our design based on the information above:



The maximum size of our Custom Packet is 1514 Bytes, and the theoretical minimum size of application data is 1 Byte. However, since the smallest packet the NIC card can send is 60 Bytes, we are going to add 0x00 for each byte until the packet size reaches 60.

3. Protocol Functionality

As described in the problem statement there are two most popular protocols UDP/IP & TCP/IP. Our Custom protocol as shown above also supports the UDP & TCP cases. In the Routing Header of Custom Protocol we have a field called "Protocol". It is customizable as per the requirements of the Node/Router. Following are the 3 kinds of Protocols, the Custom Packet Format supports.

Protocol = 0 ---- Control Protocol;

Protocol = 1 ---- Unreliable Protocol;

Protocol = 2 ---- Reliable Protocol;

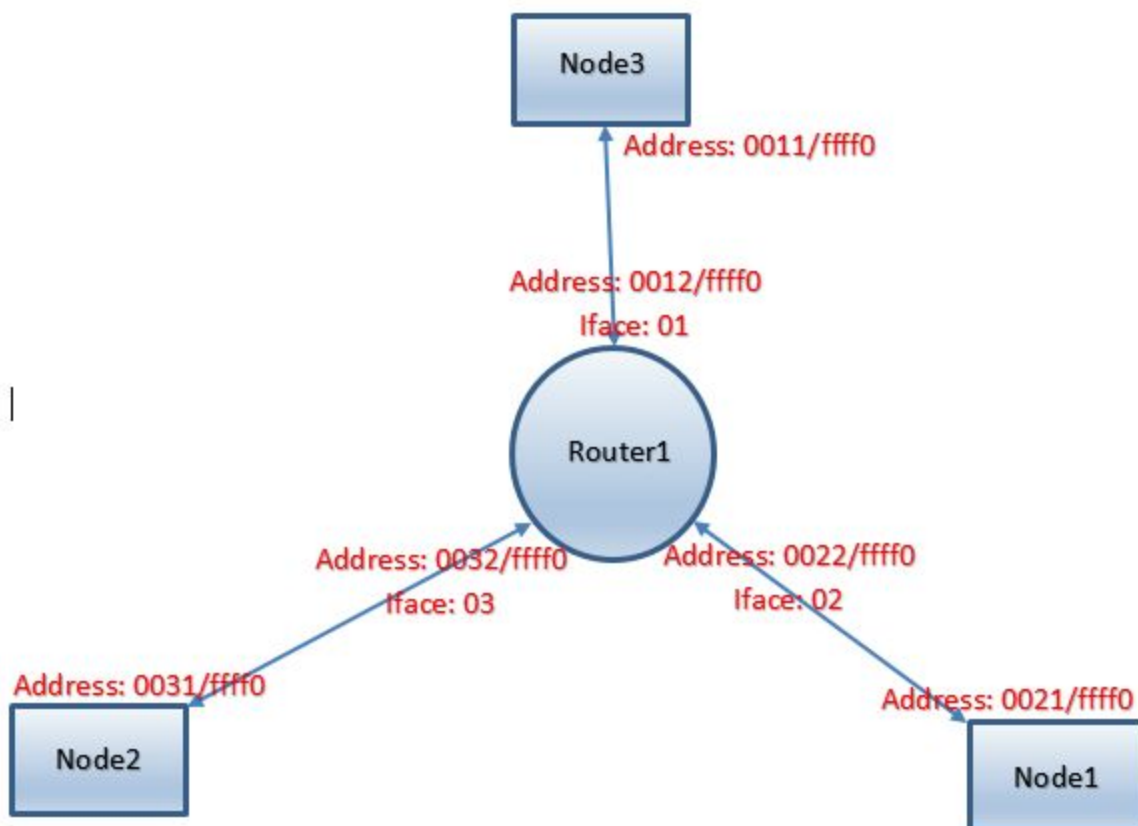
Unreliable Protocol has an Unreliable Protocol Header comprising of Port & Check fields. The Port field holds the Port number which is randomly regenerated or manually taken. It allows multiple instances of Unreliable File Transfers at a time. Check field is to compute the

Checksum of the Application Header (Unreliable Protocol) and Application Data. The total length of the Packet is not fixed.

Reliable Protocol has a Reliable Protocol Header comprising of Port, Sequence & Check fields. The purpose of the Port and Check fields remains same as of Unreliable Protocol Header. The Sequence fields holds the sequence number of the Packet that is being transferred. This field is important for reliability factor as we are track the sequence numbers of the dropped/corrupted packets so as to send the NACK.

4. DESIGN IMPLEMENTATION

We designed the topology based on the 3 Node, 1 Router model as described in 9(b). The connectivity of Nodes/Router is as shown. An example of Addresses and Interfaces are also mentioned.



The Routing table for the sample topology mentioned above is

56~63	48~55	40~47	32~39	24~31	16~23	8~15	0~7
iface	Metric	Gateway		Mask		Dest	
01	00	00	00	ff	f0	00	10
02	00	00	00	ff	f0	00	20
03	00	00	00	ff	f0	00	30

5. CODE REVIEW

Router.c

Following are the functions implemented in this module

- Sniffs at all ethernet devices infinitely
- Processing the incoming packets
- Based on the routing option, does the following:
 - Forward it to appropriate destination
 - Send a timeout response
 - Handle the error scenario

Packet.h

- Define the structure for routing protocol header
- Defines the structure for application header - Control protocol header, unreliable transfer protocol and reliable transfer protocol

Routing.c

Following are the functions implemented in this module

- Creates and Prints the Routing Table
- Checks for Routing Options
- Routing Table lookup for next hop address
- Taken the input packet, set the routing option based on protocol and destination address

Packet_Util.c

Following are the functions implemented in this module

- It generated router header checksum
- It also generated total packet checksum
- Verifies the router header checksum (Router functionality)
- Verifies the total packet checksum (Node functionality)
- Modifies the forwarding packet

Printp.c

Following are the functions implemented in this module

- It prints packet header information for reliable/unreliable/control protocols

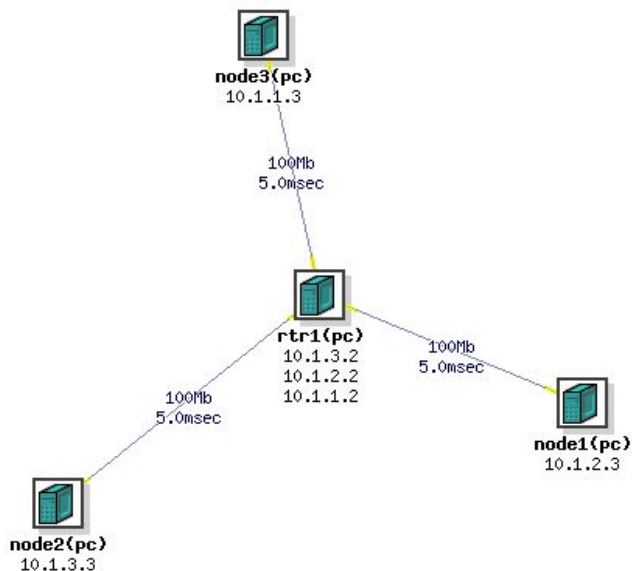
- It prints packet raw data.

6. DETER IMPLEMENTATION

NS File:

```
set ns [new Simulator]
source tb_compat.tcl
# Nodes
set rtr1 [$ns node]
tb-set-node-os $rtr1 Ubuntu1004-STD
set node1 [$ns node]
tb-set-node-os $node1 Ubuntu1004-STD
set node2 [$ns node]
tb-set-node-os $node2 Ubuntu1004-STD
set node3 [$ns node]
tb-set-node-os $node3 Ubuntu1004-STD
set link0 [$ns duplex-link $rtr1 $node1 100000.0kb 5.0ms DropTail]
set link1 [$ns duplex-link $rtr1 $node2 100000.0kb 5.0ms DropTail]
set link2 [$ns duplex-link $rtr1 $node3 100000.0kb 5.0ms DropTail]
$ns rtproto Manual
$ns run
```

Visualization



7. TEST RESULTS

1. Simple Test

In this test we generate one simple packet and send it from receiver to sender in the above mentioned topology. The simple packet contains our own header, and a sentence "This is a dummy test packet, with a size of xxx. If this packet is successfully received, this sentence should be displayed exactly the same. Here starts the random data:", and finally followed by random data. Test is SUCCESSFUL as sender side printout of packets and receiver side print out of packets are identical.

Sender Side Output (One short packet and one long packet):

```
sc558bq@node2:/tmp/lab9$ sudo ./sender
Scanning available devices ... DONE
Here are the available devices:
0. eth0      -      (null)
1. eth1      -      (null)
2. usbmon1   -      USB bus number 1
3. usbmon2   -      USB bus number 2
4. usbmon3   -      USB bus number 3
5. usbmon4   -      USB bus number 4
6. any -     Pseudo-device that captures on all interfaces
7. lo        -      (null)
Which device do you want to sniff? Enter the number:
0
Trying to open device eth0 to send ... DONE
generating packets of 256 bytes...
===== packet received, size = 256
=====
Routing Header:
|-source:      0031
|-destination: 0011
|-ttl:         214
|-protocol:    2
|-size:        256
|-checksum:    29ba
checksum test: 29ba

Reliable Protocol Header:
|-port:        86
|-sequence number: 26104
|-checksum:    e4b5
checksum test: e4b5

Data:
31 00 11 00 D6 02 01 00 29 BA 56 00 E4 B5 65 F8
54 68 69 73 20 69 73 20 61 20 64 75 6D 6D 79 20
74 65 73 74 20 70 61 63 6B 65 74 2C 20 77 69 74
68 20 61 20 73 69 7A 65 20 6F 66 20 32 35 36 2E
20 49 66 20 74 68 69 73 20 70 61 63 6B 65 74 20
69 73 20 73 75 63 63 65 73 73 66 75 6C 6C 79 20
72 65 63 65 69 76 65 64 2C 20 74 68 69 73 20 73
65 6E 74 65 6E 63 65 20 73 68 6F 75 6C 64 20 62
65 20 64 69 73 70 6C 61 79 65 64 20 65 78 61 63

1.....).V...e.
This is a dummy
test packet, wit
h a size of 256.
If this packet
is successfully
received, this s
entence should b
e displayed exac
```

74 6C 79 20 74 68 65 20 73 61 6D 65 2E 20 48 65	tly the same. He
72 65 20 73 74 61 72 74 73 20 74 68 65 20 72 61	re starts the ra
6E 64 6F 6D 20 64 61 74 61 3A 00 B9 75 E5 4A 76	ndom data:..u.Jv
9C 09 8F A4 17 D6 BC E0 4E C9 4B 4E 5E C6 EE 9AN.KN^...
56 72 58 1F 6E 2A C1 F0 F4 00 A9 69 E6 F4 E0 82	VrX.n*.....i....
FD 6F 26 14 45 E2 F5 93 AC 40 E1 0A 06 D0 A4 5C	.o&.E....@.....\
42 FC 7B B0 26 3D A1 1B 3D 4A 84 23 3E 64 A5 3C	B.{.&=..=J.#>d.<
===== end of packet =====	

DONE

generating packets of 256 bytes...

===== packet received, size = 1514

=====

Routing Header:

-source:	0031
-destination:	0011
-ttl:	80
-protocol:	2
-size:	1514
-checksum:	c5b5

checksum test: c5b5

Reliable Protocol Header:

-port:	24
-sequence number:	44693
-checksum:	4661

checksum test: 4661

Data:

31 00 11 00 50 02 05 EA C5 B5 18 00 46 61 AE 95	1...P.....Fa..
54 68 69 73 20 69 73 20 61 20 64 75 6D 6D 79 20	This is a dummy
74 65 73 74 20 70 61 63 6B 65 74 2C 20 77 69 74	test packet, wit
68 20 61 20 73 69 7A 65 20 6F 66 20 31 35 31 34	h a size of 1514
2E 20 49 66 20 74 68 69 73 20 70 61 63 6B 65 74	. If this packet
20 69 73 20 73 75 63 63 65 73 73 66 75 6C 6C 79	is successfully
20 72 65 63 65 69 76 65 64 2C 20 74 68 69 73 20	received, this
73 65 6E 74 65 6E 63 65 20 73 68 6F 75 6C 64 20	sentence should
62 65 20 64 69 73 70 6C 61 79 65 64 20 65 78 61	be displayed exa
63 74 6C 79 20 74 68 65 20 73 61 6D 65 2E 20 48	ctly the same. H
65 72 65 20 73 74 61 72 74 73 20 74 68 65 20 72	ere starts the r
61 6E 64 6F 6D 20 64 61 74 61 3A 00 D6 65 AC 50	andom data:..e.P
14 B3 EE 7B C5 94 D8 B3 9D D3 9E 9D FF C9 B5 B2	...{.....
F5 46 F4 AB 3F 66 E5 BD 59 61 9E 30 C6 4B 80 DB	.F...?f..Ya.0.K?.
FE 6E 56 C3 03 2E 76 A0 02 15 3D 01 DE F2 B3 D4	.nV...v...=.....
38 A8 7F 77 0E 64 35 67 C5 D3 97 8C 1E 18 67 1D	8.w.d5g.....g.
86 BD E0 89 EC 57 2A EE 6C 67 EF 4A 5A A2 1E 92W*.lg.JZ...

4A 9D 0A 58 02 3F C0 C7 12 57 53 31 6F BA 4E F6
78 2E 7F 64 85 A9 52 F1 11 41 3C 6B E3 5A FD 2E
F8 07 86 FA 46 46 C1 59 9E 15 8A 0D CF D8 03 47
06 83 AB 8C 2C FD 7D 3D 3E B9 A8 22 14 A6 50 0C
AD D6 06 F4 1D C7 4D BB DC D7 C8 AC AF CC F3 B5
4F 9F 41 7B 9C BF B9 DB 78 61 FD 8C 07 4D 98 B5
23 9E A9 40 66 F6 FB 42 CD C4 EE 7C 90 E2 31 DF
81 73 5A 1D 32 13 F8 AA 75 F5 37 7C 42 CF 31 66
6E DA A6 D4 D0 A2 16 9D 66 05 19 F6 E7 4B D5 68
BE 2F 85 F0 43 7E 9A B8 73 D1 34 B6 A1 66 1C 0F
40 C2 E3 11 64 F9 AE CA FE C8 C0 E5 13 95 4D D1
C5 D3 C1 08 51 5B C0 C4 2D F4 7A CE 5A 96 DD 9B
59 C0 AC BD B9 5A 88 B8 22 48 9D 35 DE EB 06 A3
BE C7 AB 0F 23 6B D3 50 5F 4E 1E BA E4 FB 55 3D
BB 01 FB 74 5B 83 2C 7E CB CA B3 A9 B5 BA 4C 73
81 F7 82 A4 62 55 F4 C2 A3 12 7C 88 0D D1 C5 C8
D2 C0 3D 2D 43 69 AB 0F 33 5F B8 E8 19 05 5B 9A
FC DD 3F 5F 33 33 21 D6 46 9D 5E 53 6E 24 1C 40
E4 59 6D 28 C2 19 37 F6 78 EF DE 91 F4 3A 2B F1
17 6A 50 4A 9E 71 21 E4 0E 7F 37 7C A3 53 BC 88
AC 29 B0 6F 42 E7 65 BA D6 43 4B CB 7D 77 BC 95
E1 0C DF 7F 7D 00 63 8B 80 9B 07 23 EE C3 AB 9B
EC 5B 0A 2F 42 6F E9 19 B2 35 E4 30 AC A0 C5 8D
AC A4 0D 29 A5 70 B4 25 0B BB 48 FA 7E F4 95 6A
4F 9F 99 92 0E 83 AB C0 B8 8F F0 64 2F B5 F1 DB
5A FE 04 FF 6F B8 24 7A 73 6C 74 F1 60 09 5B B0
A8 F5 42 B6 78 ED 77 30 7C 67 94 AB 1D 85 86 77
84 8A 76 F3 42 9A 6D B5 06 E2 A6 67 EB 01 17 94
F6 59 4A 6E 46 C1 9E C2 29 32 6D 46 B8 F3 BD 3C
7D 33 2F BF CD 9C 74 D3 7E 1A 3A 6A 1B 51 FE 12
AA 48 80 F0 0A 1F B2 33 51 1F 79 09 12 36 45 8F
69 74 4E 36 11 C2 09 8F DC 44 F9 F8 95 F7 0A 40
40 8A 30 4A A9 E3 7D FB 02 F6 04 15 2C 4A A4 95
BE F3 CB CF B5 D4 5F 92 18 58 8A AE 50 94 EE 90
1E 1E DA C8 01 57 C3 04 4D C7 19 79 11 BD 0E D0
B0 D9 9F 66 AD FE F8 C6 57 82 74 A7 16 62 37 34
80 11 FC 82 68 BF 86 B5 87 9F 2E 98 5C 3C 68 0D
15 08 73 C2 06 6B 88 5D ED FC 04 03 5E 3B 37 DF
4C 34 61 B4 F3 E7 69 7A 86 97 13 E2 D3 7B EF E8
83 62 AB 8A CD 33 E7 BA 30 EC BD 8E 27 F5 6D 74
29 CE 28 1C B5 92 97 3B 29 AA 1E FD 25 0D E5 A9
70 90 33 3D C4 1A F8 F4 06 B5 82 2E AA F0 A2 D3
BE CA F0 74 5C 87 AF 86 31 CD 83 56 DB 68 FF 4B
F9 32 88 BD 4D 80 B1 53 36 33 81 E0 23 23 B4 E2
EE A4 56 4A 2B 05 D0 5C D3 53 B2 AE BC B2 F9 B5
E4 81 72 31 02 23 85 38 56 06 18 7A 2A CC 5C 18
70 B2 62 9B B7 33 F7 8A 86 AA 38 42 5C 31 F7 40
B3 69 72 B5 8C F7 ED E3 FD 05 5D 27 D2 B9 3F 42
6B A2 DE 22 D5 D5 AD 5B 7F E5 9E DB 17 95 1C CA
FF 8E 7F 8B 85 6C 6E 82 71 CB AA 43 84 E9 86 EF

J..X.?...WS1o.N.
x.d..R..A<k.Z..
....FF.Y.....G
....,.)=>..."P.
.....M.....
O.A{...xa...M..
#...@f..B...|.1.
.sZ.2...u.7|B.1f
n.....f....K.h
./..C~..s.4..f..
@...d.....M.
....Q[...-z.Z...
Y....Z..."H.5....
....#k.P_N....U=
...t[,~.....Ls
...bU....|.....
..=-Ci...3_....[.
..?_33!.F.^Sn\$.@
.Ym(..7.x.....+.
.jPJ.q!..7|.S..
.).oB.e..CK.}w..
...}.c.?..#....
.[./Bo...5.0....
...).p.%..H.~..j
O.....d/...
Z...o.\$zslt.`.[.
..B.x.w0|g.....w
..v.B.m....g....
.YJnF...)2mF...<
}3/...t.~.:j.Q..
.H?...3Q.y..6E.
itN6.....D.....@
@.0J..}.....,J..
....._..X..P...
.....W..M..y....
...f....W.t..b74
?...h.....\<h.
..s..k.]....^;/7.
L4a...iz.....{..
.b...3..0...'..mt
p.3=.....
...t\...1..V.h.K
.2..M?.S63..##..
..VJ+..\S.....
..r1.#.8V..z*.\.
p.b..3....8B\1.@
.ir.....]'...?B
k..."...[.....
....ln.q..C....

8B 64 12 60 39 BF BC B9 A4 5A 94 BB EF B0 85 EE	.d.`9....Z.....
3E 04 7A C3 70 E8 46 E2 B4 F0 25 38 D9 AB 28 65	>.z.p.F...%8..(e
0F 3A C5 49 F9 81 02 9D DB 96 59 CB 47 DE B9 85	...I.....Y.G...
E3 33 49 53 1C 8F 35 D0 7F 5B 08 58 06 30 BD 16	.3IS..5.[.X.0..
6A 83 5F 63 04 61 01 E0 F7 5A AB 3E 38 64 C4 1B	j._c.a...Z.>8d..
98 0D 6F B4 9C A4 84 1B FF 8C 73 06 BD 31 1C 27	..o.....s..1.'
B4 7B 8B B8 DC 8C 98 D3 E6 43 12 1E A8 D6 3A 40	.{.....C.....:@
E3 A9 F4 7F 4D 78 9A 4D 04 0D 53 C1 3E 6F E9 F2	...Mx.M..S.>o..
EA 74 AB C6 00 43 99 E6 87 AB 04 2F 81 3E 6F 64	.t...C...../.>od
E7 63 E3 35 DB 7D 82 DF 8B D5 A1 C9 44 8A BC 2E	.c.5.}.....D...
FE 67 F4 FE AA 8D E4 31 39 E8 60 BA 27 CF 1F 0E	.g.....19.`.'...
32 02 43 0D 80 C5 ED 0B 9A 8E D4 DE 18 90 0C 16	2.C.?.....
F7 00 14 A2 8E F8 D3 C7 E0 34 81 07 03 A0 16 364.....6
A3 59 43 23 1F 30 2E B9 BE 02 98 D6 93 A4 EC 8A	.YC#.0.....
A5 00 2C 33 F8 00 FA D9 34 7B E0 37 1C F6 6D BF	...3....4{.7..m.
50 B1 E2 6F E1 10 28 A0 12 C0 76 A5 65 63 30 0A	P..o..(...v.ec0.
63 5C 3D 5C 5C 37 35 90 B2 15 C8 CE 0C 35 8D 5C	c\=\75.....5.\
E6 6F CB C8 7F F3 68 92 B4 DE 37 19 41 67 23 A5	.o...h...7.Ag#.
C4 60 01 20 97 36 B1 49 4B 79 18 57 AE A5 B3 95	.`. .6.IKy.W....
15 7E 5D 94 72 C5 26 26 A3 5E 3F E5 C5 62 8A 89	.~].r.&&.^?...b..
C2 8B AA 59 C1 5B A2 0C D4 BA 64 82 60 17 17 75	...Y.[....d.`..u
96 74 09 08 39 30 2E DD 8E 6D C2 53 CF 4C DD 91	.t..90...m.S.L..
D7 87 EA 98 E2 8C A4 B6 47 08 38 A7 20 50 1C B6G.8. P..
C4 25 BE FE 55 EC DB E3 59 9D 37 28 E9 14 B9 C0	.%..U...Y.7(....
9B A3 58 7D 2F FC 33 76 05 6B 1D 25 BB 39 DB 80	..X}/.3v.k.%9.?
5F 99 7E B4 85 59 98 DE F6 CF 06 DF E3 BF 9F 7E	_~...Y.....~
62 F7 FB 91 F3 2E 08 F8 99 25 1D 55 5F F8 D5 BE	b.....%U_...
91 53 72 16 AC 0A F4 A2 D9 FA	.Sr.....

===== end of packet =====

DONE
END OF TEST

Receiver Side Output (One short packet and one long packet):

```
sc558bq@node3:/tmp/lab9$ sudo ./receiver
Scanning available devices ... DONE
Here are the available devices:
0. eth0      -      (null)
1. eth1      -      (null)
2. usbmon1   -      USB bus number 1
3. usbmon2   -      USB bus number 2
4. usbmon3   -      USB bus number 3
5. usbmon4   -      USB bus number 4
6. any -     Pseudo-device that captures on all interfaces
```

7. lo - (null)

Which device do you want to sniff? Enter the number:

0

Trying to open device eth0 to send ... DONE

===== packet received, size = 256

=====

Routing Header:

-source:	0031
-destination:	0011
-ttl:	213
-protocol:	2
-size:	256
-checksum:	2aba

checksum test: 2aba

Reliable Protocol Header:

-port:	86
-sequence number:	26104
-checksum:	e4b5

checksum test: e4b5

Data:

31 00 11 00 D5 02 01 00 2A BA 56 00 E4 B5 65 F8
54 68 69 73 20 69 73 20 61 20 64 75 6D 6D 79 20
74 65 73 74 20 70 61 63 6B 65 74 2C 20 77 69 74
68 20 61 20 73 69 7A 65 20 6F 66 20 32 35 36 2E
20 49 66 20 74 68 69 73 20 70 61 63 6B 65 74 20
69 73 20 73 75 63 63 65 73 73 66 75 6C 6C 79 20
72 65 63 65 69 76 65 64 2C 20 74 68 69 73 20 73
65 6E 74 65 6E 63 65 20 73 68 6F 75 6C 64 20 62
65 20 64 69 73 70 6C 61 79 65 64 20 65 78 61 63
74 6C 79 20 74 68 65 20 73 61 6D 65 2E 20 48 65
72 65 20 73 74 61 72 74 73 20 74 68 65 20 72 61
6E 64 6F 6D 20 64 61 74 61 3A 00 B9 75 E5 4A 76
9C 09 8F A4 17 D6 BC E0 4E C9 4B 4E 5E C6 EE 9A
56 72 58 1F 6E 2A C1 F0 F4 00 A9 69 E6 F4 E0 82
FD 6F 26 14 45 E2 F5 93 AC 40 E1 0A 06 D0 A4 5C
42 FC 7B B0 26 3D A1 1B 3D 4A 84 23 3E 64 A5 3C

1.....*.V...e.
This is a dummy
test packet, wit
h a size of 256.
If this packet
is successfully
received, this s
entence should b
e displayed exac
tly the same. He
re starts the ra
ndom data:..u.Jv
.....N.KN^...
VrX.n*.....i....
.o&.E....@.....\
B.{.&=..=J.#>d.<

===== end of packet =====

===== packet received, size = 1514

=====

Routing Header:

-source:	0031
-destination:	0011
-ttl:	79

```
| -protocol:      2
| -size:          1514
| -checksum:      c6b5
checksum test: c6b5
```

Reliable Protocol Header:

```
| -port:          24
| -sequence number: 44693
| -checksum:      4661
checksum test: 4661
```

Data:

```
31 00 11 00 4F 02 05 EA C6 B5 18 00 46 61 AE 95
54 68 69 73 20 69 73 20 61 20 64 75 6D 6D 79 20
74 65 73 74 20 70 61 63 6B 65 74 2C 20 77 69 74
68 20 61 20 73 69 7A 65 20 6F 66 20 31 35 31 34
2E 20 49 66 20 74 68 69 73 20 70 61 63 6B 65 74
20 69 73 20 73 75 63 63 65 73 73 66 75 6C 6C 79
20 72 65 63 65 69 76 65 64 2C 20 74 68 69 73 20
73 65 6E 74 65 6E 63 65 20 73 68 6F 75 6C 64 20
62 65 20 64 69 73 70 6C 61 79 65 64 20 65 78 61
63 74 6C 79 20 74 68 65 20 73 61 6D 65 2E 20 48
65 72 65 20 73 74 61 72 74 73 20 74 68 65 20 72
61 6E 64 6F 6D 20 64 61 74 61 3A 00 D6 65 AC 50
14 B3 EE 7B C5 94 D8 B3 9D D3 9E 9D FF C9 B5 B2
F5 46 F4 AB 3F 66 E5 BD 59 61 9E 30 C6 4B 80 DB
FE 6E 56 C3 03 2E 76 A0 02 15 3D 01 DE F2 B3 D4
38 A8 7F 77 0E 64 35 67 C5 D3 97 8C 1E 18 67 1D
86 BD E0 89 EC 57 2A EE 6C 67 EF 4A 5A A2 1E 92
4A 9D 0A 58 02 3F C0 C7 12 57 53 31 6F BA 4E F6
78 2E 7F 64 85 A9 52 F1 11 41 3C 6B E3 5A FD 2E
F8 07 86 FA 46 46 C1 59 9E 15 8A 0D CF D8 03 47
06 83 AB 8C 2C FD 7D 3D 3E B9 A8 22 14 A6 50 0C
AD D6 06 F4 1D C7 4D BB DC D7 C8 AC AF CC F3 B5
4F 9F 41 7B 9C BF B9 DB 78 61 FD 8C 07 4D 98 B5
23 9E A9 40 66 F6 FB 42 CD C4 EE 7C 90 E2 31 DF
81 73 5A 1D 32 13 F8 AA 75 F5 37 7C 42 CF 31 66
6E DA A6 D4 D0 A2 16 9D 66 05 19 F6 E7 4B D5 68
BE 2F 85 F0 43 7E 9A B8 73 D1 34 B6 A1 66 1C 0F
40 C2 E3 11 64 F9 AE CA FE C8 C0 E5 13 95 4D D1
C5 D3 C1 08 51 5B C0 C4 2D F4 7A CE 5A 96 DD 9B
59 C0 AC BD B9 5A 88 B8 22 48 9D 35 DE EB 06 A3
BE C7 AB 0F 23 6B D3 50 5F 4E 1E BA E4 FB 55 3D
BB 01 FB 74 5B 83 2C 7E CB CA B3 A9 B5 BA 4C 73
81 F7 82 A4 62 55 F4 C2 A3 12 7C 88 0D D1 C5 C8
D2 C0 3D 2D 43 69 AB 0F 33 5F B8 E8 19 05 5B 9A
FC DD 3F 5F 33 33 21 D6 46 9D 5E 53 6E 24 1C 40
E4 59 6D 28 C2 19 37 F6 78 EF DE 91 F4 3A 2B F1
17 6A 50 4A 9E 71 21 E4 0E 7F 37 7C A3 53 BC 88
AC 29 B0 6F 42 E7 65 BA D6 43 4B CB 7D 77 BC 95
```

```
1...O.....Fa..
This is a dummy
test packet, wit
h a size of 1514
. If this packet
is successfully
received, this
sentence should
be displayed exa
ctly the same. H
ere starts the r
andom data:..e.P
...{.....
.F..?f..Ya.0.K?.
.nV...v...=.....
8.w.d5g.....g.
.....W*.lg.JZ...
J..X.?...WSlo.N.
x.d..R..A<k.Z..
....FF.Y.....G
....,.,}=>..."..P.
.....M.....
O.A{...xa...M..
#..@f..B...|...1.
.sZ.2...u.7|B.1f
n.....f....K.h
./..C~...s.4...f..
@...d.....M.
....Q[...-z.Z...
Y....Z..."H.5....
....#k.P_N....U=
...t[.,~.....Ls
....bU....|.....
..=-Ci..3_....[.
..?_33!.F.^Sn$.@
.Ym(..7.x.....:+.
.jPJ.q!..7|.S..
.) .oB.e..CK.}w..
```

E1 0C DF 7F 7D 00 63 8B 80 9B 07 23 EE C3 AB 9B
EC 5B 0A 2F 42 6F E9 19 B2 35 E4 30 AC A0 C5 8D
AC A4 0D 29 A5 70 B4 25 0B BB 48 FA 7E F4 95 6A
4F 9F 99 92 0E 83 AB C0 B8 8F F0 64 2F B5 F1 DB
5A FE 04 FF 6F B8 24 7A 73 6C 74 F1 60 09 5B B0
A8 F5 42 B6 78 ED 77 30 7C 67 94 AB 1D 85 86 77
84 8A 76 F3 42 9A 6D B5 06 E2 A6 67 EB 01 17 94
F6 59 4A 6E 46 C1 9E C2 29 32 6D 46 B8 F3 BD 3C
7D 33 2F BF CD 9C 74 D3 7E 1A 3A 6A 1B 51 FE 12
AA 48 80 F0 0A 1F B2 33 51 1F 79 09 12 36 45 8F
69 74 4E 36 11 C2 09 8F DC 44 F9 F8 95 F7 0A 40
40 8A 30 4A A9 E3 7D FB 02 F6 04 15 2C 4A A4 95
BE F3 CB CF B5 D4 5F 92 18 58 8A AE 50 94 EE 90
1E 1E DA C8 01 57 C3 04 4D C7 19 79 11 BD 0E D0
B0 D9 9F 66 AD FE F8 C6 57 82 74 A7 16 62 37 34
80 11 FC 82 68 BF 86 B5 87 9F 2E 98 5C 3C 68 0D
15 08 73 C2 06 6B 88 5D ED FC 04 03 5E 3B 37 DF
4C 34 61 B4 F3 E7 69 7A 86 97 13 E2 D3 7B EF E8
83 62 AB 8A CD 33 E7 BA 30 EC BD 8E 27 F5 6D 74
29 CE 28 1C B5 92 97 3B 29 AA 1E FD 25 0D E5 A9
70 90 33 3D C4 1A F8 F4 06 B5 82 2E AA F0 A2 D3
BE CA F0 74 5C 87 AF 86 31 CD 83 56 DB 68 FF 4B
F9 32 88 BD 4D 80 B1 53 36 33 81 E0 23 23 B4 E2
EE A4 56 4A 2B 05 D0 5C D3 53 B2 AE BC B2 F9 B5
E4 81 72 31 02 23 85 38 56 06 18 7A 2A CC 5C 18
70 B2 62 9B B7 33 F7 8A 86 AA 38 42 5C 31 F7 40
B3 69 72 B5 8C F7 ED E3 FD 05 5D 27 D2 B9 3F 42
6B A2 DE 22 D5 D5 AD 5B 7F E5 9E DB 17 95 1C CA
FF 8E 7F 8B 85 6C 6E 82 71 CB AA 43 84 E9 86 EF
8B 64 12 60 39 BF BC B9 A4 5A 94 BB EF B0 85 EE
3E 04 7A C3 70 E8 46 E2 B4 F0 25 38 D9 AB 28 65
0F 3A C5 49 F9 81 02 9D DB 96 59 CB 47 DE B9 85
E3 33 49 53 1C 8F 35 D0 7F 5B 08 58 06 30 BD 16
6A 83 5F 63 04 61 01 E0 F7 5A AB 3E 38 64 C4 1B
98 0D 6F B4 9C A4 84 1B FF 8C 73 06 BD 31 1C 27
B4 7B 8B B8 DC 8C 98 D3 E6 43 12 1E A8 D6 3A 40
E3 A9 F4 7F 4D 78 9A 4D 04 0D 53 C1 3E 6F E9 F2
EA 74 AB C6 00 43 99 E6 87 AB 04 2F 81 3E 6F 64
E7 63 E3 35 DB 7D 82 DF 8B D5 A1 C9 44 8A BC 2E
FE 67 F4 FE AA 8D E4 31 39 E8 60 BA 27 CF 1F 0E
32 02 43 0D 80 C5 ED 0B 9A 8E D4 DE 18 90 0C 16
F7 00 14 A2 8E F8 D3 C7 E0 34 81 07 03 A0 16 36
A3 59 43 23 1F 30 2E B9 BE 02 98 D6 93 A4 EC 8A
A5 00 2C 33 F8 00 FA D9 34 7B E0 37 1C F6 6D BF
50 B1 E2 6F E1 10 28 A0 12 C0 76 A5 65 63 30 0A
63 5C 3D 5C 5C 37 35 90 B2 15 C8 CE 0C 35 8D 5C
E6 6F CB C8 7F F3 68 92 B4 DE 37 19 41 67 23 A5
C4 60 01 20 97 36 B1 49 4B 79 18 57 AE A5 B3 95
15 7E 5D 94 72 C5 26 26 A3 5E 3F E5 C5 62 8A 89
C2 8B AA 59 C1 5B A2 0C D4 BA 64 82 60 17 17 75

...}.c.?...#....
.[./Bo...5.0....
...)..p.%..H.~..j
O.....d/...
Z...o.\$zslt.`.[.
..B.x.w0|g.....w
..v.B.m....g....
.YJnF...)2mF...<
}3/...t.~.:j.Q..
.H?....3Q.y..6E.
itN6.....D.....@
@.0J..}.....,J..
....._..X..P...
.....W..M..y....
...f....W.t..b74
?...h.....\<h.
..s..k.]....^;/7.
L4a...iz.....{..
.b...3..0...'.mt
)..(....;)...%...
p.3=.....
...t\...1..V.h.K
.2..M?.S63...##..
..VJ+..\S.....
..r1.#.8V..z*.\.
p.b..3....8B\1.@
.ir.....]'..?B
k.."...[.....
....ln.q..C....
.d.`9....Z.....
>.z.p.F...%8..(e
...I.....Y.G...
.3IS..5.[.X.0..
j._c.a...Z.>8d..
..o.....s..1.'
. {.....C.....:@
...Mx.M..S.>o..
.t...C...../.>od
.c.5.}.....D...
.g.....19.`.'...
2.C.?.....
.....4.....6
.YC#.0.....
..,3....4{.7..m.
P..o..(...v.ec0.
c\=\\75.....5.\
.o...h...7.Ag#.
.`. .6.IKy.W....
..~].r.&&.^?..b..
...Y.[....d.`..u

```

    96 74 09 08 39 30 2E DD 8E 6D C2 53 CF 4C DD 91      .t..90...m.S.L..
    D7 87 EA 98 E2 8C A4 B6 47 08 38 A7 20 50 1C B6      .....G.8. P..
    C4 25 BE FE 55 EC DB E3 59 9D 37 28 E9 14 B9 C0      .%..U...Y.7(...
    9B A3 58 7D 2F FC 33 76 05 6B 1D 25 BB 39 DB 80      ..X)/.3v.k.%9.?
    5F 99 7E B4 85 59 98 DE F6 CF 06 DF E3 BF 9F 7E      _..~...Y.....~
    62 F7 FB 91 F3 2E 08 F8 99 25 1D 55 5F F8 D5 BE      b.....%.U....
    91 53 72 16 AC 0A F4 A2 D9 FA                        .Sr.....
===== end of packet =====

sc558bq@node3:/tmp/lab9$

```

2. File Transfer Test

We tried to send a file from node2 to node3, using our protocol and it is successful:

Sender Side Output:

```

sc558bq@node2:/tmp/lab9/FT$ sudo ./sender testfile.txt node3 12345
Scanning available devices ... DONE
Here are the available devices:
0. eth0      -      (null)
1. eth1      -      (null)
2. usbmon1   -      USB bus number 1
3. usbmon2   -      USB bus number 2
4. usbmon3   -      USB bus number 3
5. usbmon4   -      USB bus number 4
6. any -     Pseudo-device that captures on all interfaces
7. lo -      (null)
Which device do you want to sniff? Enter the number:
0
Trying to open device eth0 to send ... DONE
Current time: Sat Oct 10 16:19:11 2015
generating packets...
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 1414
Sending a UNRELIABLE packet of size: 470
DONE

```

Receiver Side Output:

```

sc558bq@node3:/tmp/lab9/FT$ sudo ./receiver 12345 testfile_2.txt 17256

```

```
Scanning available devices ... DONE
Here are the available devices:
0. eth0      -      (null)
1. eth1      -      (null)
2. usbmon1   -      USB bus number 1
3. usbmon2   -      USB bus number 2
4. usbmon3   -      USB bus number 3
5. usbmon4   -      USB bus number 4
6. any -     Pseudo-device that captures on all interfaces
7. lo -      (null)
Which device do you want to sniff? Enter the number:
0
Trying to open device eth0 to receive ... DONE
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 1414
Received an UNRELIABLE packet of size: 470
File successfully written
Current time: Sat Oct 10 16:19:11 2015
DONE
sc558bq@node3:/tmp/lab9/FT$ diff testfile.txt testfile_2.txt
sc558bq@node3:/tmp/lab9/FT$
```

The diff test passed.

