Sprint 3 Deliverable Questions

Jacob Buckelew

1. What assumptions, if any, does IP make about local networks and lower level links used to transmit datagrams? How are these consistent with design goals IP?

IP makes very minimal assumptions about local networks since it was designed to run over all types of networks in an internetwork. IP is very undemanding of local networks as it is very effective to have a best-effort service over a reliable network than a reliable service model over an unreliable network. Thus, IP will want to provide a best-effort service that runs over all networks no matter the performance barriers that may exist on certain networks to increase heterogeneity of internetworking. This means more networks can remain connected with another and be able to send data between routers regardless of how reliable the particular networks are in their own performances.

2. Describe IP's best-effort service model

IP makes no guruntees when datagrams are sent across networks and gives its best effort when sending datagrams since it doesn't assume the performance of local networks. IP is very undemanding and allows just about any network to be part of an internetwork. Furthermore, this model can be split into two key ideas: an addressing scheme that identifies networks and the hosts on those networks, and a connectionless model for datagram delivery. There's no set of rules for a network that tells it what to do when datagrams arrive, only the fundamental concept that every datagram must be sent with the network making its best effort to pass it along to its destination. The "best-effort" model also means that if packets get lost, corrupted, misdelivered, or fail to reach their destination, nothing is done to recover from the failure. Thus, the best-effort service is very unreliable.

3. Why does IP need to use IP addresses to identify source and destination of IP datagrams?

Ethernet addresses may be globally unique, but because of the issues that routing raises on a large internetwork, Ethernet is not enough since it doesn't provide much structure or clues to routing protocols.
In other words, Ethernet is very flat as opposed to IP addresses which are very hierarchical since they consist of 2 components: the network part and host part.
The hierarchical aggregation that results from differentiating network from host allows routers to only have to deal with reaching the right network, and once data has been sent to the right network the local network can then deal with the host part.

4. IP address with classless network prefix 128.96.16/20. What is max number of hosts that can be attached to the network, assuming 32 bit addresses?

128.96.16/20

Host

$\underbrace{10000000.01100000.0001}_{\text{same for all} \quad 20 \text{ prefix}}\overbrace{0000.00000000}^{\text{Host}}$

20 bits	12 bits
Network part	Host part

2^{12} unique hosts on network

$4096 - 2 = 4094$ hosts

0 not valid and also the broadcast address isn't available

5. Dijkstra

Node A

Node	Best Distance	Path
B	3 ✓	A → B
D	5 ✓	A → D
C	9 ✓	A → B → C
E	11 ✓	A → B → E
F	13 ✓	A → B → E → F

6. RIP Table 1

	A	B	C	D	E	F
A	0	3	∞	5	∞	∞
B	3	0	6	∞	8	∞
C	∞	6	0	∞	∞	∞
D	5	∞	∞	0	7	∞
E	∞	8	∞	7	0	2
F	∞	∞	∞	∞	2	0

Table 2 = After 1 Exchange

	A	B	C	D	E	F
A	0	3	9	5	11	∞
B	3	0	6	8	8	10
C	9	6	0	∞	14	∞
D	5	8	∞	0	7	9
E	11	8	14	7	0	2
F	∞	10	∞	9	2	0

Table 3 = After 2 Exchanges

	A	B	C	D	E	F
A	0	3	9	5	11	13
B	3	0	6	8	8	10
C	9	6	0	14	14	16
D	5	8	14	0	7	9
E	11	8	14	7	0	2
F	13	10	16	9	2	0

SDN

What is the control plane? What is the data plane? Why might we want to seprate them? What advantages does SDN offer over traditional network architectures?

The control plane refers to the background processing and software that are required to control the network using routing algorithms like RIP. The data plane refers to the actual processing of packets and the forwarding decisions that get made in the white-box switches. We might want to seprate the two to maximize the ability of the data plane to execute packet processing and increase efficiency. For example, by giving the data plane its own Network Processing Unit (NPU), a multi-stage pipeline can be implemented that allows for the NPU to process multiple packets at any given moment. This is much more efficient than having one single CPU that runs both the data plane and the control plane. Now, just the CPU can focus on routing algorithms and be able to sent "data programs" to the NPU if forwarding decisions need to be updated.