Lecture 14: AS Hierarchy, BGP, and Subnetting

1 Announcements

1. Complaints on midterm grading must be handled via written petition – please read bboard – by Friday 5 p.m..
2. These network level lectures are going to be getting more complicated. Please make use of office hours to ask questions about the lectures.

2 AS Hierarchy

So far we have seen two routing algorithms. Both however are only practical in the case of networks with fewer than 100 routers.

But the Internet has **many more routers** and **over 50,000 networks**!

The size of the Internet creates several big headaches:

1. Any routing algorithm like Distance-Vector or Link-State will never complete when applied to the large number of routers in the Internet, since the time for those algorithms to converge is proportional to the number of routers.
2. Any routing algorithm will have to pass around too much data, flooding all the links in the Internet.
3. The forwarding tables, assuming that they ever get created, will be huge! Remember we need one entry for each network. How will we store them?

The solution: We subdivide the Internet into it's naturally separate administrative domains, also called **Autonomous Systems (AS)**. Each Autonomous System might correspond to a different ISP (remember the hierarchy of ISPs that we talked about?).

The idea is that within each AS we will run either RIP or OSPF to find shortest-paths within that AS. (An AS is not too big – maybe 100 routers, so it works to use RIP or OSPF). We then however need a way to communicate between AS's and exchange information.

The exchange of information between AS's is done by just a few **gateway routers** which sit at the border of the AS. The protocol that these gateway routers use for exchanging routing information is called the **Border Gateway Protocol** – **BGP**. The **BGP** protocol is different from both RIP and OSPF and is quite complicated! We will be discussing **BGP4** (version 4). Cisco offers a 2 week course to explain BGP4. I will try to cover the main points in half a lecture. In preparing the material below, I have used 4 textbooks, plus several Web resources. If you would like more information than what I cover here, please send me email.

To help us organize our discussion of routing in the big bad Internet, we will repeatedly return to 2 questions:

1. What do the **forwarding tables** look like for each of the routers. That is, which networks does each router know about (know the next hop to)?
2. What is the **routing information** that is exchanged between the routers? What do the routing tables look like? How do routers choose the "best path?"

Rather than dumping the whole algorithm on you, let's try to *discover* it ourselves a little at a time. We'll start with an **unrealistic** view of the Internet as a tree, and discuss what we would want BGP to do in that circumstance. Then we will add in the real-world complexities one at a time.

3 If the world were a tree ...

Consider Figure 1. This is a much simplified abstraction of the Internet as a tree.

Observe that there are three types of AS's: **Local**, **Regional** , and **Backbone (National)**. The local AS's charge customers like you and me for the service of connecting us to the Internet, so we can send and receive messages. Each local AS in turn pays a fee to its parent in the tree, the regional AS, which in turn pays a fee to the backbone AS.

The relationship between AS's shown in Figure 1 is a **transit** relationship. Local AS's are called stubs. Regional AS's provide *transit*, (a.k.a. access) to local AS's. Backbone AS provides *transit* to regional. This is in exchange for payment. Payment is typically based on total *bandwidth used* per month.

Now let's discuss what we would want the forwarding tables to look like in this scenario, and that will help us to figure out the appropriate routing algorithm.

Question: The forwarding table of each router in a local AS should have how many entries?

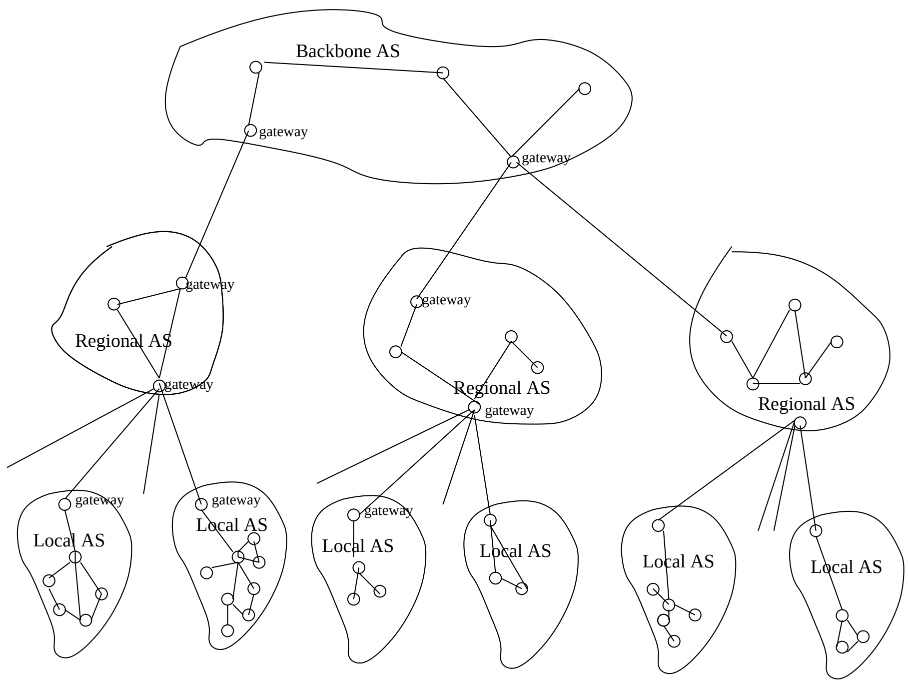

Figure 1: *If the Internet were a tree*

Answer: One entry for each network in the AS plus one entry which is a default entry. That is, each router in the AS should know the next hop router to every network in the AS. However, if it sees a network prefix which is not a network in the AS, it should send this packet to its default router, which should be the **gateway** router for the AS. The gateway router will then have the very simple job of just forwarding the packet up one hop to the transit regional AS.

Question: The forwarding table of each router in the regional AS should have how many entries?

Answer: Each router in the regional AS should be able to reach any network in the subtree below it. Thus each router in the regional AS should have an entry in its forwarding table for all the networks in all the AS's for which it is a transit provider. This may sound like a lot of entries, but it's not as bad as it seems. First of all, we'll talk later about how network prefixes can be aggregated so that we don't need to have so many of them. Secondly, the routers in the regional AS are very powerful and can handle long forwarding tables. The forwarding tables in these routers also need one extra entry which is the default entry pointing to the gateway router which connects to the backbone AS. If a router doesn't recognize a network prefix, it sends the packet up to the gateway router connected to the backbone AS. This gateway router has the simple job of just forwarding the packet up one hop to the backbone.

Question: The forwarding table of each router in the backbone AS should have how many entries?

Answer: Every router in the backbone must know about every network in the Internet. In practice this works out to about 50,000 entries in the forwarding table of such routers. Luckily, they're very powerful routers who can handle this.

Question: So how would these forwarding tables get created? What might be the routing algorithm for getting this information to each of the routers?

Answer: Routing could work like this: Within each AS, we would use RIP or OSPF to determine the shortest paths to all networks in the AS. We'd quickly get to the point where every router in the local AS has a forwarding table entry for every network within that local AS. But now we need something more. We need to broadcast the set of reachable networks up a level to our transit provider, the regional AS. This would be done by a protocol, BGP, that is run only on the gateway routers (no point in getting too many routers involved). The gateway routers advertise all the networks they can reach to their transit (provider) AS and put themselves down as the *next hop*. The gateway routers in the Regional AS would now use their usual OSPF or RIP to get this information distributed within the whole Regional AS.

It is important to observe that in this artificial tree-like world, the BGP algorithm is quite simple. It's simply a way for gateway routers to broadcast their reachable networks up to their provider AS.

4 The world is not a tree: cooperation vs. competition

Unfortunately the world is not a tree, and therefore our BGP algorithm is more complicated than what's described above. The "real world" is shown in Figure 2 below.

1. In the tree world, there was only one path of AS's connecting any two networks. In the real world there may be many possible paths of AS's connecting two networks. A local AS may be **multi-homed**, meaning that it receives service from more than one regional AS (Figure 2 shows two multi-homed AS's). Also, regional AS's may connect to each other (as shown in Figure 2).

2. In the tree world, all inter-AS relationships were **transit** relationships, where an AS pays its transit (provider) AS to forward its packets (the payment is proportional to bandwidth used). In the real world, there are also **peering** relationships between AS's, where two regional AS's might agree to use each other to some limited degree.

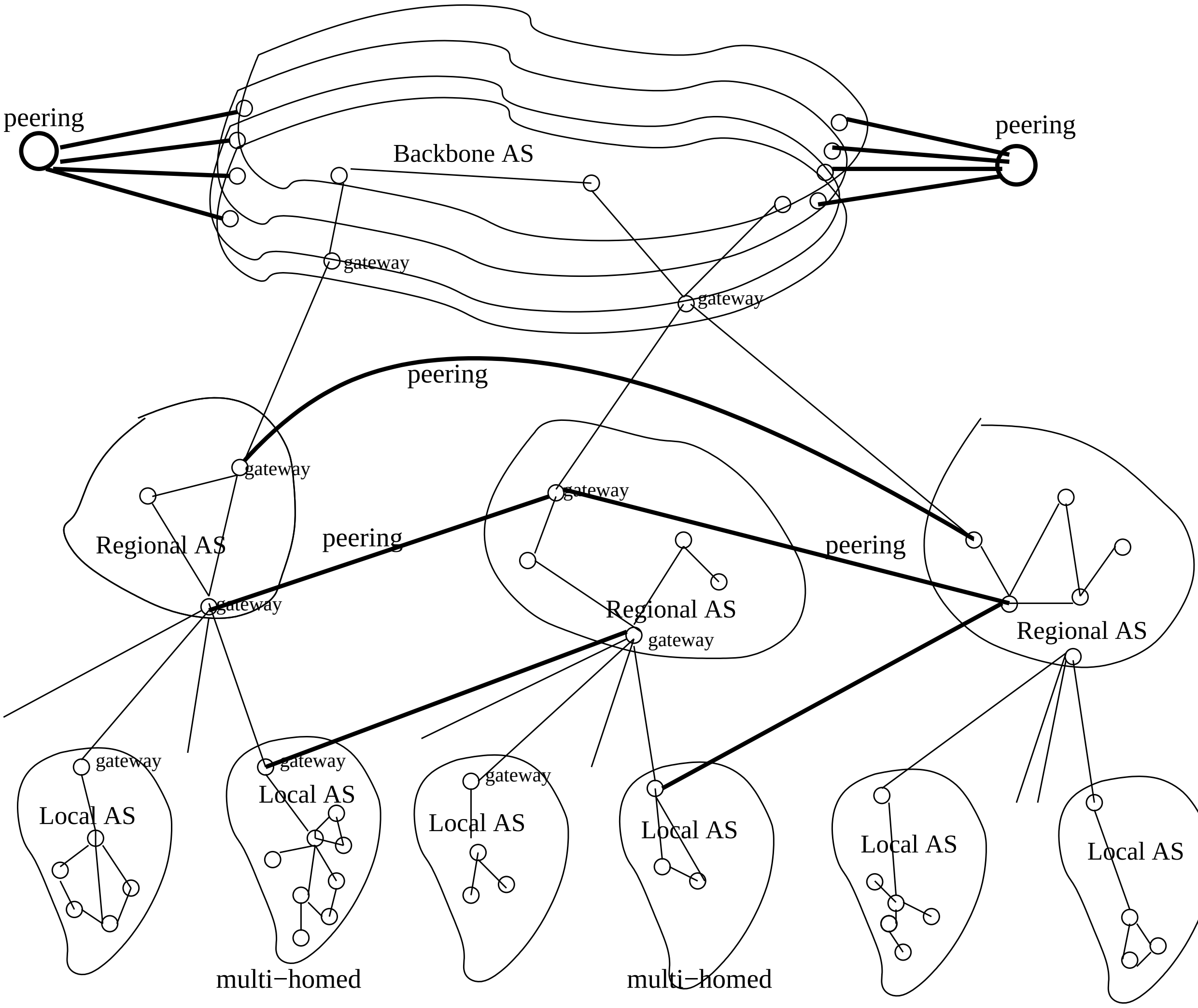

Figure 2: *The real Internet. I've started with the tree and show additional connections in bold. There are a lot more additional connections than what is shown here.*

Typically there is no payment in peering. Peering is typically done to save money. Rather than pay your transit (provider) AS to forward your packet, you might be able to set up a relationship with your peer AS who can forward your packet instead, in exchange for your forwarding their packet.

Question: More specifically, if regional AS1 wants to set up a peering relationship with regional AS2, how is this done in terms of exchanging routes?

Answer: Regional AS1 will give all its network entries to regional AS2. Thus regional AS2's routers know know how to forward through regional AS1 to get to those networks. At the same time, regional AS2 will give its network entries to regional AS1.

Observe that peering relationships might mean that a packet originating in a Local AS might travel through several regional AS's before it ends up at the local AS of its destination.

3. In the tree world, there was only one backbone AS. In the real world there are **20 backbone AS's**. An individual backbone AS may not be able to connect to every local AS, so there are **peering points** between different backbone AS's, set up out of necessity, so that the Internet stays connected. See Figure 2.

We now discuss some *consequences* of the above differences.

1. In the artificial world, the costs used in evaluating routes were still the usual cost metrics (propagation delay, or transmission time, or queueing delay), however costs mean less now because different AS's may use different cost metrics.

2. Since there are now many ways to get from one Local AS to another Local AS, a regional router now has a **choice** of whether to go up to the backbone or through another regional router. The cost metric in regional ASs is therefore typically *not* path length, but rather financial cost.

3. Also, an AS needs to now be careful what it advertises. A local AS still needs to advertise all of its networks to its transit regional AS (because it's customers are paying for that service), however regional AS1 may not want to advertise the fact that it can get to the local AS's of its peer, regional AS2.

   Question: Why not?

   Answer: If regional AS1 tells its backbone that it has a route to the local AS's of regional AS2, then the backbone will start sending traffic through regional AS1 that is directed for those local AS's or regional AS2. Regional AS1 doesn't receive payment for providing this bandwidth, so this is not in its interest.

Likewise, if regional AS1 can get to the local AS's of regional AS2, and it lets regional AS3 know about that, then regional AS3 may decide to go through regional AS1 to get to regional AS2.

4. It's clear that the billing structure is very complicated now. To make things worse, so AS's may for **security reasons** not want traffic from certain other AS's going through them, even if a route exists.

Again we ask the same two questions:

1. What's in the forwarding table of each router? That is, which networks does each router know about?
2. What's the routing algorithm which determines the entries in the forwarding table – i.e. the chosen next hop router?

With respect to question 1, we restrict our attention to non-gateway routers at first: The forwarding tables of routers within a local AS will contain entries for each network within the local AS, plus a default entry which points to the gateway router for the local AS. The forwarding table of a router within a regional AS will have entries for every network in every local AS below it (because it provides transit service to these local AS's, so they want to advertise all their networks upward). However, a router within a regional AS may also have entries for some networks which are local to its peer regional AS's. Regional routers will also likely have a default entry pointing to the gateway router for the regional AS. The forwarding table of a router within a national backbone AS will have entries for all the networks spanned by the regional AS's in the tree below it, but may also have entries for networks covered by its peer backbone AS's. In fact, it's likely, for performance reasons, that backbone routers have entries for all networks in the Internet (modulo aggregation due to subnetting and supernetting – coming up soon).

With respect to question 2, let's try to figure out what must happen:

Again let's start by assuming that within each AS we're doing OSPF to exchange information so all routers in the AS know the same set of networks.

Again, we imagine a second routing algorithm, BGP, which is only spoken by the gateway routers of each of the AS's. This goal of the BGP algorithm is of course to let each router know the best next hop to each network. Observe that there are now many possible paths to each network. Some paths to network 128.2 go through the backbone, some don't, etc.. We want each router to deduce it's "best" path to network 128.2 (actually just the next IP hop on its best path to 128.2). To achieve this, one could imagine the gateway routers simply running RIP amongst each other, but rather than exchanging distances to *all* the networks they know, they've been programmed by their sys admins to be *selective* about which routing tables they exchange and with whom (as agreed by contracts).

Question: The above idea sounds like it would let each router deduce the cheapest way to get to network 128.2 and the next IP hop on that cheapest route. What's wrong with the above idea?

Answer: There are two things wrong:

1. As RIP is run among the gateway routers, a gateway router in AS0 might find itself hearing an advertisement from AS1 that it can get to 128.2 in cost 8 (8 hops), and an advertisement from AS2 that it can get to 128.2 in cost 5 (5 hops).

   Question: Should the router therefore pick to go through AS2, and assign its routing table entry for 128.2 a cost of 6 and the next hop to be the AS2 gateway router?

   Answer: Not necessarily. There may be financial reasons for why going through AS2 will be more expensive from the perspective of AS0.

2. So suppose that AS2 is in fact desirable for AS0.

   Question: Should the router therefore pick to go through AS2, and assign its routing table entry for 128.2 a cost of 6 and the next hop to be the AS2 gateway router?

   Answer: Not necessarily. Just because AS2 is a desirable next hop, doesn't mean that the rest of the route after AS2 is also desirable. In fact, maybe the rest of the route after AS2 takes us through AS3 which is really expensive from AS0's perspective. The RIP messages don't let us see this *hidden fact*.

So it seems that doing RIP among the gateway routers is a good idea except that "cost" should be in terms of expense, rather than number of hops. The problem is that expense is relative to who you are. One router might view a path via AS1 as expensive and undesirable, while another router might view a path via AS1 as cheap. Remember that different AS's might have **different payment contracts** set up.

Question: So what's the solution?

Answer: Do RIP among the gateway routers (of course exchanging only limited subsets of your knowledge as specified by your sys admin). However, rather than passing just the next hop router to each network and the hop count, instead pass the **entire path** to each destination network. This entire path could contain all the IP hops, however, it suffices to only pass around a list of the AS's that are traversed on the path (each AS has an identification number). Thus a routing table entry that is passed around will contain an attribute called **ASPATH** which is a list of all the AS's on the path to that network. When a gateway router is receiving information it may receive info about 2 possible ASPATHs through which it can get to 128.2. For example:

$$AS1 \rightarrow AS3 \rightarrow AS16 \rightarrow 128.2$$

and

$$AS2 \rightarrow AS15 \rightarrow AS12 \rightarrow 128.2$$

The gateway router should choose the path which is cheapest for it according to a set of rules (preferences) set for it by its sys admin. These rules may be simple, like "always choose to go through a peer AS as opposed to a backbone AS," or complex, like "avoid going through AS17 at all cost." Often these rules are proprietary. You will only get a real feel for what happens if you end up working for one of these ISPs.

The above algorithm (RIP with paths) is called a **path-vector algorithm**, since it is the same as the **distance-vector** algorithm, except that rather than passing around distances it passes around whole paths. BGP is a path-vector algorithm.

Note that passing around the full path also has other added benefits: it prevents **routing loops** and **count-to-infinity problems** associated with RIP.

Question: Suppose, in our above example, our gateway router hears about path:

$$AS1 \rightarrow AS3 \rightarrow AS16 \rightarrow 128.2$$

from AS1's gateway, and hears about path:

$$AS2 \rightarrow AS15 \rightarrow AS12 \rightarrow 128.2$$

from AS2's gateway. Suppose for financial reasons, our gatway router chooses to go with the first path above, because going through AS12 is more expensive. Now at some later time, our gateway router hears from AS1 that AS1 is advertising a different path to 128.2. This third path looks like:

$$AS1 \rightarrow AS13 \rightarrow AS16 \rightarrow 128.2$$

Suppose that AS13 is even more expensive for our gateway router than AS12. How should our gateway router interpret this new message and how should it respond?

Answer: Our gateway router should realize that it may continue sending packets destined to 128.2 through AS1, however those packets will now be directed to AS13, rather than AS3, because that's the best choice for AS1. Consequently, our gateway would now like to change it's path to going through path 2:

$$AS2 \rightarrow AS15 \rightarrow AS12 \rightarrow 128.2$$

For this reason, it's important that our gateway router has stored all possible paths that it hears of to 128.2, in case it has to revert to an earlier (but still existing) choice.

A few more details:

The BGP routing algorithm runs over TCP – on port 179, and involves only the gateway routers. The BGP algorithm takes a while to run (on the order of minutes, because so much data is being passed around), so it's really important that none of the routing data is lost. This is why TCP is used.

When there's a breakdown of a link, the gateway routers have to start the BGP algorithm again, but they only send around **changes** from last time. A lot of work can also be avoided by storing **backup** routes.

It is important to realize that although the routing tables of the gateway routers are complex, the **final forwarding tables** for the gateway routers look like regular forwarding tables with entries of the form **(network part, outgoing interface, MAC addr of next IP hop)**. These routers just have more entries.

There's a recent Sigcomm 2000 paper by Labovitz et. al. which discusses some open research problems involving the BGP routing algorithm. In particular, the paper laments the slow convergence properties of BGP. Networking research in the area of routing protocols today is primarily focused on improving BGP.

In summary:

Routing on the Internet may be quite complex, involving traversal through many different ASs, due to the (often proprietary) financial agreements between AS's. Routing algorithms used to construct the forwarding tables take place at 2 levels: exchange of info within at AS (often using RIP or OSPF) and exchange between the gateways of the AS's (using BGP).

5 How to shrink the size of the forwarding tables

In the first half of this lecture we saw that routers within a local AS only have entries in their forwarding tables for those networks within the AS. In contrast, however, some routers (for example routers in the backbone) may have an entry for **every single network** in the Internet! The size of these forwarding tables can be a performance bottleneck. The purpose of the rest of this lecture and part of next lecture is to discuss techniques which are used for keeping the forwarding tables from being as huge as we might think.

Within our discussion we will once again be discussing the format of a forwarding table. So far we've assumed that the forwarding table in every router in the Internet has entries of the form:

(Network Part of destination addr, Outgoing Interface, MAC addr of next hop)

We'll now see some circumstances where the entries in the forwarding table are a little more complicated than what's shown above.

Goal 1: It would be really nice if we could reduce the number of entries in the forwarding tables of regional and backbone routers. That is, we'd like to reduce the number of network addresses that these routers need to know of.

A seemingly separate problem has to do with IP addressing. It turns out that we are quickly **running out of IP addresses**. This may sound surprising, given that there are 4 billion possible IP addresses.

Question: Explain the problem of **inefficient** address utilization:

Answer:

Suppose you are a network with 2 nodes. This means that you need a class C network address. But now you've wasted 253 out of the 255 possible addresses associated with that network address!

Here's an even more serious example. Suppose you are a network with 256 nodes. You no longer fit into a class C network address, which means you need to ask for a class B network address. But you only use 256 out of the $2^{16} = 65,536$ IP addresses possible for that network address, a utilization of only 0.3%.

Question: How many class B network prefixes are there?

Answer: Only $2^{14} \approx 16000$ class B network prefixes. Thus it is easy to see how the class B addresses could get used up.

In fact it is exactly the class B addresses which are running out. (There are a lot more class C network addresses). This brings us to a second goal:

Goal 2: We'd like to use the address space more efficiently, so it doesn't run out.

We will discuss dual techniques: **subnetting** and **supernetting (a.k.a., CIDR)**, both of which aim to satisfy these two goals.

6 Subnetting

Subnetting aims to both reduce the total number of network numbers that are assigned, while also creating better utilization of the address space.

The idea is simple:

Under subnetting, a **single IP network number** (e.g., a class B network number) is allocated to several different networks. Each of these networks is now referred to as a **subnet**, and functions as a separate network (i.e., it is separated by routers from other networks).

For example, say have we have a class B network number: 128.96. This network number could correspond to a network with up to $2^{16}-1$ nodes. Suppose that we don't have such a huge network. However we do have 3 smaller networks: Network X has $2^{15}-1$ nodes. Network Y has $2^{14}-1$ nodes. Network Z has $2^{14}-1$ nodes. We would like to share the single class B network number among these 3 networks.

Here's how we might do this:

```
X =  1 0 0 0 0 0 0 0 . 0 1 1 0 0 0 0 0 . 1 x x x x x x x . x x x x x x x x

Y =  1 0 0 0 0 0 0 0 . 0 1 1 0 0 0 0 0 . 0 1 x x x x x x . x x x x x x x x

Z =  1 0 0 0 0 0 0 0 . 0 1 1 0 0 0 0 0 . 0 0 x x x x x x . x x x x x x x x
```

The 3 subnets are shown in Figure 3.

Question: But how will the routers know how to interpret IP addresses? Remember that the network number is still 128.96.

Question: For routers outside of network 128.96, what should the forwarding table entry look like?

Answer: These routers only need a single entry in their forwarding table:

$$(128.96,\ \text{outgoing interface}\ ,\ \text{MAC addr of next IP hop}\ )$$

These networks see the collection of subnets as just the **single network**: 128.96.

Once the packet reaches a router with network number 128.96, that router needs to have a somewhat different forwarding table, which allows it to detect the appropriate subnetwork.

Question: How do you think the appropriate subnetwork is detected?

Answer: With a **mask**, or actually, with **multiple masks**.

Here's how it works. The router has a table which looks like this:

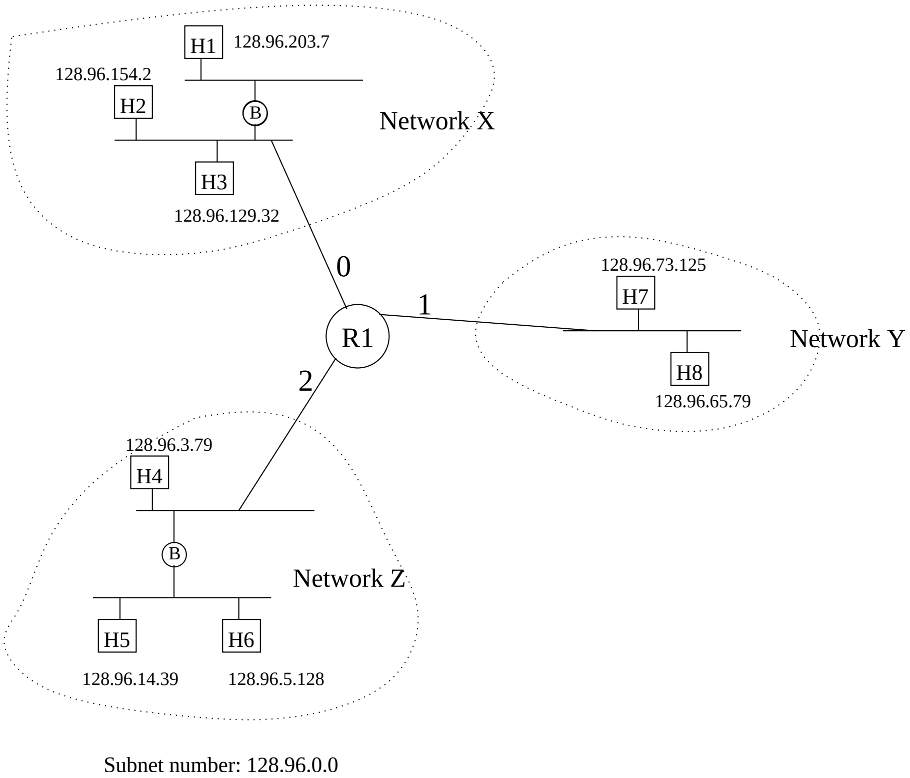

Figure 3: *3 subnets of network number 128.96.*

Mask	Subnet Number	Outgoing Interface
255.255.128.0	128.96.128.0	(Subnet X) Interface 0
255.255.192.0	128.96.64.0	(Subnet Y) Interface 1
255.255.192.0	128.96.0.0	(Subnet Z) Interface 2

The destination IP addr is ANDed with the first Mask entry. If the result matches the first Subnet Number, then the router knows that the destination is subnet X, and sends the packet there. If there is no match, then the destination IP addr is ANDed with the second Mask entry. If the result matches the second Subnet Number, then the router knows that the destination is subnet Y, and sends the packet there. If there is no match, then the destination IP addr is ANDed with the third Mask, and so on and so forth.

Observe that all the hosts on a given subnet are configured with the same mask. The bitwise AND of the host number and the mask defines the subnet number.

To be continued next time ....

Thank You!