CSIS0234B Computer and Communication Networks (Class B)
Reading for Tutorial 7
Basic routing in the Internet

In the lectures we talk about routing tables, which allow hosts in a network to communicate, without regard to the underlying network architecture. In another word, the underlying network will do it for them. We will see how this is achieved in networks running the Internet Protocol.

1. Addressing

As is described in the lecture, most of the time the network address is part of the host address. This allows the routing software to find, for each packet, where should it be sent to. For the Internet protocol, the host address is called an "IP address", and is a 32-bit integer. The network address is some of the bits of the IP address. It is vitally important that you understand how one can extract the network address from the IP address. That information can be found in the lecture notes 7.13–7.17, so it is not repeated here. Please familiarize yourselves with the addressing scheme by reading those few pages.

2. Routing in IP networks

The Internet Protocol delivers the datagram by checking the destination IP address in the IP datagram header. If the it is an address of a local network, the packet is delivered directly to the destination, using the mechanisms like ARP and direct frame sending. Otherwise, the packet may be sent to another computer which knows how to send it to the destination, or it may simply be discarded. The decision about what to do for each packet (arriving either from the upper transport layer or from another host) is called *routing.*

Some computers, called *gateways* or *IP routers,* are responsible for relaying, or *forwarding,* packets from one computer to another until it reaches the intended destination. Obviously, an IP router must have multiple network interfaces so that it can receive packets from one and forward the packet to another. The reverse is not true: some computers (known as *multi-homed hosts*) may have more than one interface, but refuse to forward packets. Indeed, the default configuration in most OS is for computers having multiple network interfaces to be multi-homed rather than routing. This allows mis-configuration of routing tables to be discovered more easily and before major problem occurs. For setting up a router, one has to do an extra step to enable IP forwarding. E.g., in Linux, this is done by writing the string "1" to the file `/proc/sys/net/ipv4/ip_forward`:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Writing "0" instead of "1" will disable forwarding again.

3. Routing tables

For most hosts, the routing decisions are simple: if the destination host is on the local network, the data is delivered to the destination host directly using the network interface; if the destination host is on the remote network, the data is delivered to a "default" gateway. It is more complicated at gateways: depending on the *network address* of the packet, the packet should be forwarded to a different interface.

Such decisions are configured by a *routing table.* A routing table maps network addresses to either the router (when forwarding) or an interface (when using local network addresses) that must be used to reach the destination. In Linux, we use the `route` command (possibly with the `-n` option if you don't want DNS lookup) to display the routing table[1]. Here is a typical routing table:

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
```

Each entry of the routing table has a `destination` and a `genmask`. Together they give the network address for which routing is specified. For example, the `192.168.0.0` and `255.255.255.0` above specifies that the table entry is for the routing of destination addresses from `192.168.0.0` to `192.168.0.255`. Conceptually[2], each incoming packet will be matched with all these networks, in order to find the entries with a matching network address. There may be multiple entries that matches a packet, and in such cases the entry with the most specific (i.e., largest) `genmask` prevails. For example, if a packet arrives from `192.168.0.1`, both the first and the third entry are matched, but the first is chosen since it has a larger `genmask`. Note that it is also possible to have entries for a single host, by having `genmask` being `255.255.255.255`.

When an address matches an entry in the table, the `Gateway` field tells how to reach the specified destination. If the `Gateway` field contains the IP address of a router, the router is used. If the `Gateway` field contains `0.0.0.0`, the destination network is directly connected. The `Iface` field shows the network interface used for the route. In the example, it is either the first Ethernet interface (`eth0`) or the loopback interface (`lo`).

The remaining four fields (`Ref`, `Use`, `Flags` and `Metric`) display some extra information about the route, which are only of marginal values (e.g. a `U` in the `Flag` indicates that the route is up, and a `G` flag indicates whether or not an external gateway is used, etc.). For more details on these fields, read `route(8)`.

4. Manually building a routing table

The `route` command is also used to modify the routing table. It is invoked like this:

```
route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.0.1
```

The first argument `add` is a keyword (which should be either `add` or `del`), telling `route` either to add a new route or delete an existing one. The next argument `-net` specify that routing entry for a network, rather than a host, is installed. (For host routes it is not possible to use a netmask.) Then the destination, the genmask and the gateway are specified ("genmask" and "gateway" are specified by the keywords "netmask" and "gw" respectively). Alternatively, one can specify the keyword `default`, for an entry with both destination and genmask being `0.0.0.0` (i.e., the most general route). For other configurable options, you can also read `route(8)` for details.

---

[1] The `netstat -r` command is used to examine the routing table on Solaris 8 systems.

[2] In practice, the routing table rarely change, so the same address results in the same route. So a cache is kept to avoid having to scan the whole routing table to make the same decision everytime a packet arrives. You can examine the cache by using `route -C`.