2.10 Example of routing in real life

Example of routing in real life

Have you ever thought about how information travels inside the network? Have you ever noticed that, when you browse the Internet, some pages are faster than others to open? And no, it is not all about how powerful your computer is or how fast your connection is. Millions of packets travel the Internet, trying to reach different points. We can think of the Internet like a long highway. Along this highway there are signs for the important destination, however, not for all the possible destinations. So, how would a packet be able to reach you home?

The key idea behind the routing technique on the Internet is similar to the routing technique applied to telephone numbers and postal codes. As telephone numbers start with a country code, continue with an area code and finally a local exchange, IP Addresses have a similar structure, identifying the network with the first sequence of bits and identifying the individual node in the network with the final bits. In order to manage the routing of the packets and provide services that improve the routing itself (eg rerouting packets in case of congestion, etc) a protocol is needed. This protocol is called BGP.

You will be asked to find information about the protocol in the next activity.

2.13 DNS and DNS cache poisoning

DNS and DNS cache poisoning

Humans access information online through domain names, like nytimes.com or espn.com because we find it easier to remember these names rather than plain IP addresses. However, web browsers interact through Internet Protocol (IP) addresses. The **Domain Name System** translates domain names to IP addresses so browsers can load Internet resources.

The Domain Name System (DNS) is the phonebook of the Internet.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorise IP addresses such as 192.168.1.1 (in IPv4), or more complex, newer, alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6). Historically, this mapping was kept in a local file. However, as the Internet keeps evolving it was necessary to create a system that was scalable. This is the DNS.

DNS has a hierarchical structure composed of three servers.

Root nameserver

The root server is the first step in translating (resolving) human readable host names into IP addresses. It can be thought of like an index in a library that points to different racks of books. Typically it serves as a reference to other more specific locations.

TLD nameserver

The top level domain server (TLD) can be thought of as a specific rack of books in a library. This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (in google.com, the TLD server is ‘com’).

Authoritative nameserver

This final nameserver can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative nameserver is the last stop in the nameserver query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Recursor (the librarian) that made the initial request.

In addition, there is what is defined as a DNS recursor server. The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers. Typically, the recursor is then responsible for making additional requests in order to satisfy the client's DNS query.

There are two ways to interrogate a DNS server: iteratively or recursively.

Iterative approach

Let's have a look at how an iterative approach works. We assume that a hostname se.sjsu.edu is requesting the IP address of mail.yahoo.com. We also assume that the authoritative DNS server for mail.yahoo.com is dns.yahoo.com. Here's how the DNS resolves a request.

The host se.sjsu.edu sends a DNS query to the local DNS server to translate the hostname 'mail.yahoo.com', provided in the query, to the IP address. In response, the local DNS server ie dns.sjsu.edu forwards the query to the root DNS server. The root DNS server finds the suffix as 'com' and returns a list of IP addresses of the top-level DNS server responsible for 'com'. The local DNS server then sends the same query to one of the top level DNS servers which were provided by the root DNS server. The top-level DNS server finds a suffix yahoo.com and returns the local DNS server with an IP address of the authoritative DNS server for Yahoo, ie yahoo.com. Finally, the local DNS server sends the same query again to the authoritative DNS server dns.yahoo.com, which in turn responds with the IP address of mail.yahoo.com.

The above process can be summarised as:

User's computer: 'What is the IP address of mail.yahoo.com?'
Local name server: 'I don't know that but I'll check with a name server that does.'

Local name server: 'Root name server, what is the IP address of mail.yahoo.com?'
Root name server: 'Here are the addresses for the TLD name servers for .com.'

Local name server: 'TLD server, what is the IP address of mail.yahoo.com?'
TLD .com name server: 'Here are the addresses of the authoritative name servers for yahoo.com.'

Local name server: 'Authoritative name server, what is the IP address of mail.yahoo.com?'
Authoritative mail.yahoo.com name server: 'Here is the IP address for mail.yahoo.com, it's 205.139.94.60.'

Recursive approach

Let's now have a look on how recursive approach. The complete flow of the recursive query is as follows.

1. Requesting host se.sjsu.edu asks its local DNS server dns.sjsu.edu to solve a DNS query mail.yahoo.com and to give its IP address.
2. The local DNS query asks the root DNS server for the IP address of mail.yahoo.com.
3. The root DNS server finds the 'com' suffix in the query and requests one of the top level DNS servers responsible for 'com'.
4. The top level DNS server keeps track of the entire authoritative DNS server and it asks the authoritative DNS server of Yahoo (dns.yahoo.com) for the IP address of mail.yahoo.com.
5. The authoritative DNS server of Yahoo returns the IP address to the top level DNS server who queries the authoritative DNS.
6. The top level DNS server returns this IP address to the root DNS server.
7. The root DNS server, in turn, returns the IP address to the local DNS query.
8. The host receives the IP address of its desired query.

In theory, a recursive query is resolved in the manner explained above. But in practice, recursive query is not used as it is not very efficient.

DNS caching

In the above two approaches of solving the query, if you count the number of requests sent you will find a total of 10 messages. This makes the DNS not very efficient; therefore a caching mechanism is designed so as to reduce the flooding of DNS packets in cyberspace. The DNS makes extensive use of a cache to improve the performance which is otherwise decreased by going through the root DNS server, TLD server and the authoritative DNS server. Caching reduces traffic of DNS packets over the internet.

When a client creates a request for a specific website, if the specific website is not in the cache of the local DNS server, a DNS query process is started as explained before. However, once the local DNS server gets the response, it records it in the cache and every time a client requests the IP address of that result, the local DNS server won't run the whole DNS query process but rather it will return the cached result to the client.

DNS attack: poisoning

In previous sections, we saw that approaches that use a cache (eg ARP) can be subject to cache **poisoning**. The DNS can also be subjected to that type of attack. Let's see briefly how it works.

As before, a client creates a request for a specific website. The root DNS server finds the suffix as 'com' and returns a list of IP addresses of the top-level DNS server responsible for 'com'. The local DNS server then sends the same query to one of the top level DNS servers responsible for 'com' which were provided by the root DNS server. Then the top-level DNS server finds a suffix google.com and returns the local DNS server with an IP address of the authoritative DNS server for google, ie google.com. Finally, the local DNS server sends the same query again to the authoritative DNS server dns.google.com.

This time, however, before the authoritative DNS server replies with the IP address of www.google.com, an attacker replies to the request indicating a specific IP address that belongs to the attacker. So, the local DNS server will record in its cache that www.google.com is the IP of the attacker and every time a client requests the IP address of www.google.com, the DNS server will return the poisoned entry.

Using DNS poisoning, an attacker can point users to wherever they want, however, most of the time, this technique is combined with **website spoofing**. For example, a client is trying to get access to his bank. A wireless router with a DNS poisoned server redirects the request to the attacker who is pretending to be the client and it takes the request, modifies it, and sends it to the legit bank while still pretending to be the client. The bank will respond with the client data to the attacker and the client's bank account is compromised.