Overview

- Last Lecture
  - Scheduled tasks and log management
- This Lecture
  - DNS and BIND
  - Reference: DNS and BIND, 4th Edition, O'Reilly
- Next Lecture
  - Address assignment (DHCP)

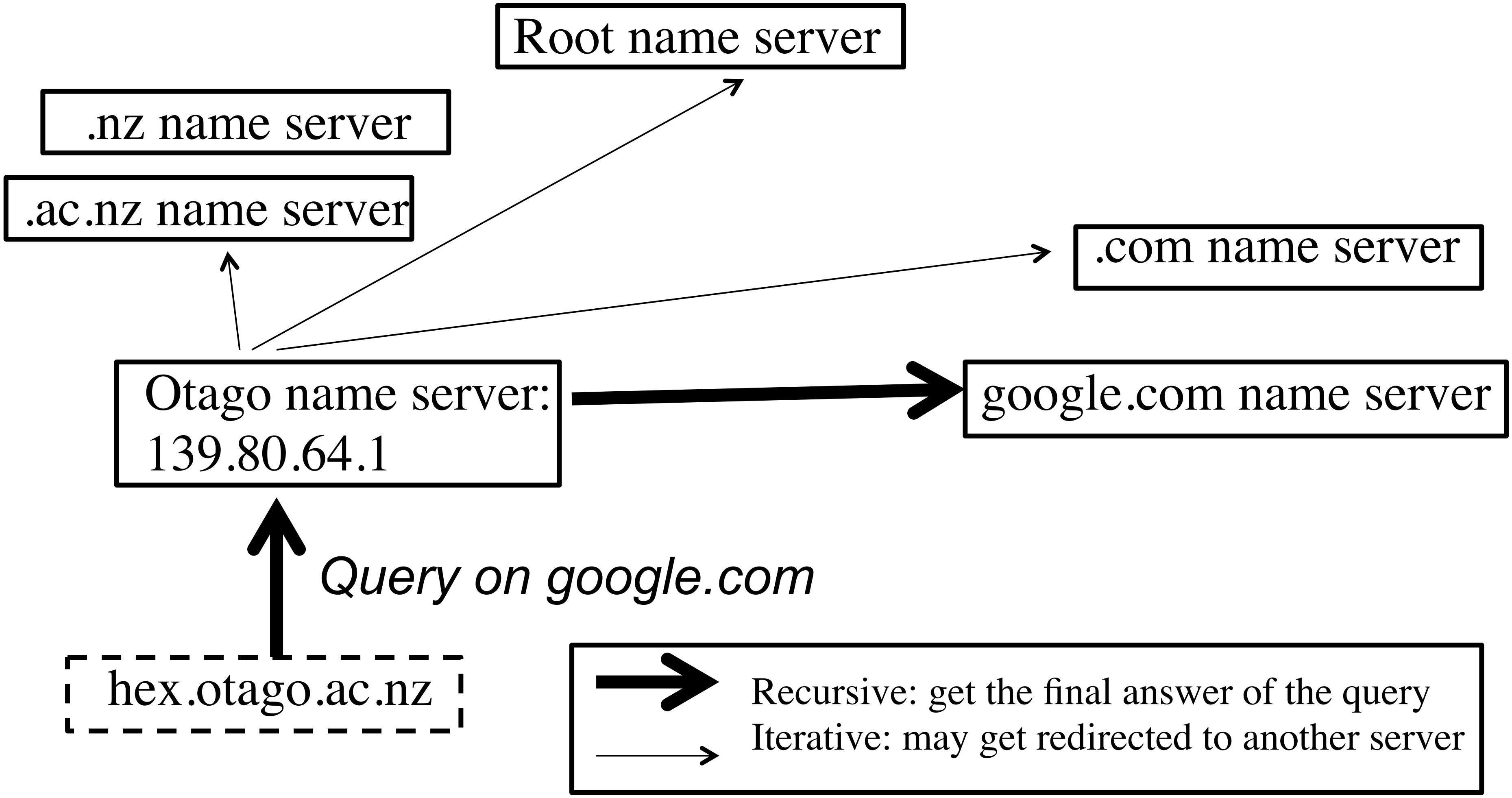
Problem

- How to get the IP address with an IP name?
  - Mapping between IP addresses and IP names
- Simple solution
  - Central database, like */etc/hosts* or Sun Microsystems' NIS (Network Information Service) or Windows' WINS for LAN.
- Domain Name Service (DNS)
  - A distributed solution
  - BIND is an implementation, an acronym for Berkeley Internet Name Domain

How DNS works?

- Terms: domain name, domain name space, domain, Fully Qualified Domain Name (FQDN)
  - Similar to file system hierarchy, root domain, subdomains, subsubdomains, …
  - otago.ac.nz., a fully qualified domain
- The name space is divided into zones
  - Zone is a domain including its child domains
  - At least one server in each zone
  - The servers in a zone are responsible for answering queries about the zone

How DNS works? (cont.)



Root name server

.nz name server

.ac.nz name server

.com name server

Otago name server: 139.80.64.1

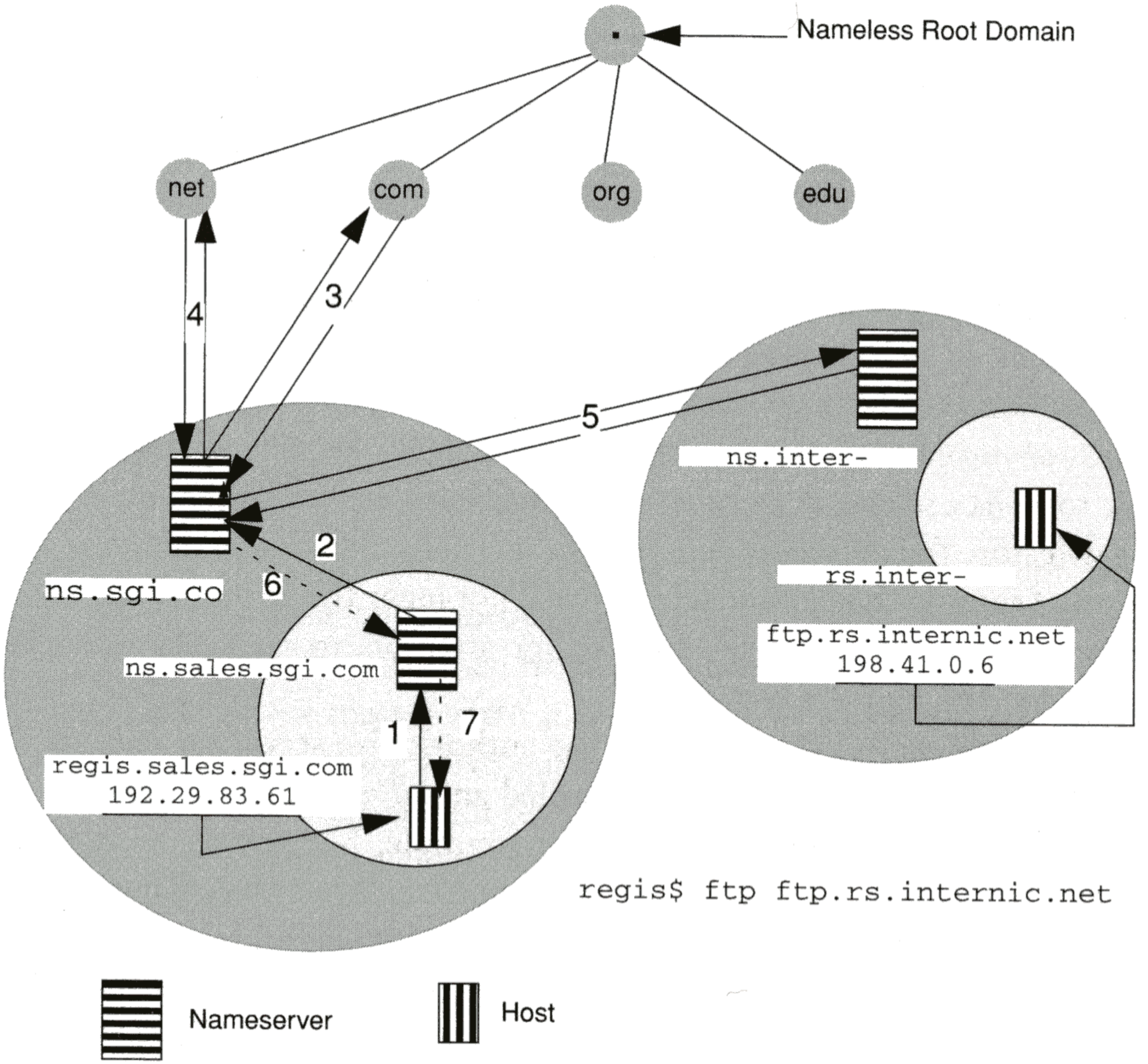
google.com name server

Query on google.com

hex.otago.ac.nz

Recursive: get the final answer of the query

Iterative: may get redirected to another server



```
regis$ ftp ftp.rs.internic.net
```

C

Figure 10–2 A Typical DNS Query

DNS client

- The DNS clients use the resolver library
  - Resolver is a collection of C functions. The central routines are **gethostbyname** and **gethostbyaddr**
- There are two important files used by those routines: **host.conf** and **nsswitch.conf**
- Note: DNS is not the only name service for a client!
  - It could be a file with the mappings, NIS, or DNS
  - You can decide in which order to use them

DNS client (cont.)

- **/etc/nsswitch.conf**: used by GNU standard library glibc
  - Set the order of **dns, nis**, **files,** etc
- **/etc/host.conf**: used by older Linux standard library libc
- If DNS is used, the name servers are specified in the file **resolv.conf** in a client machine

DNS server

- Types of servers
  - Primary (master) server: data is stored in authoritative source files and maintained by system administrators
  - Secondary (slave) server: mirrors the data in the primary server and downloads its data second-hand from a primary server at regular intervals

DNS server (cont.)

- To configure a name server, there are several important files
  - /etc/named.conf: specify where to find files for lookups and reverse lookups at the local domain, where to forward requests for outside of the domain
  - Database files for each local domain and local subnets
    - Often under **pz** or **sz**, but not required
    - Get a template to create them
  - **named.cache** or **named.root**
    - Contains the names of root servers
    - The name server needs them when a request is out of its knowledge
- To start up name server simply type
  - **/usr/sbin/named**

Resource record

- Resource record format for **named** server
  - *Domain [ttl] [class] type data*
  - Domain: to which domain this entry applies
  - ttl: force resolvers to discard info after a certain time
  - Class: IN for IP addresses (Internet), other classes like Hesiod is mostly confined to MIT)
  - Record type: A, AAAA, SOA, PTR, NS, and MX
  - Data: depends on type
  - E.g. hex 14400 IN A 139.80.137.37, hex is hex.otago.ac.nz
  - vex A 139.80.137.40, the missing fields inherit the values of the previous record.
- Best practices
  - Set proper ttl field for RR
  - Create PTR records (reverse mapping records) for verifying IP addresses with IP names

RR types

- Types of records in DNS database
  - **SOA**: indicates the start of authority for this domain
  - **NS**: lists a name server for this domain or sub-domain
  - **MX**: lists a mail exchanger for this domain with priority
  - **A**: defines the canonical name of a host with a given IP address; AAAA for IPv6
  - **CNAME**: associates an alias with a canonical name
  - **PTR**: for reverse lookup (IP address to name)
  - **HINFO**: advertises host info, CPU and OS
  - **TXT**: description

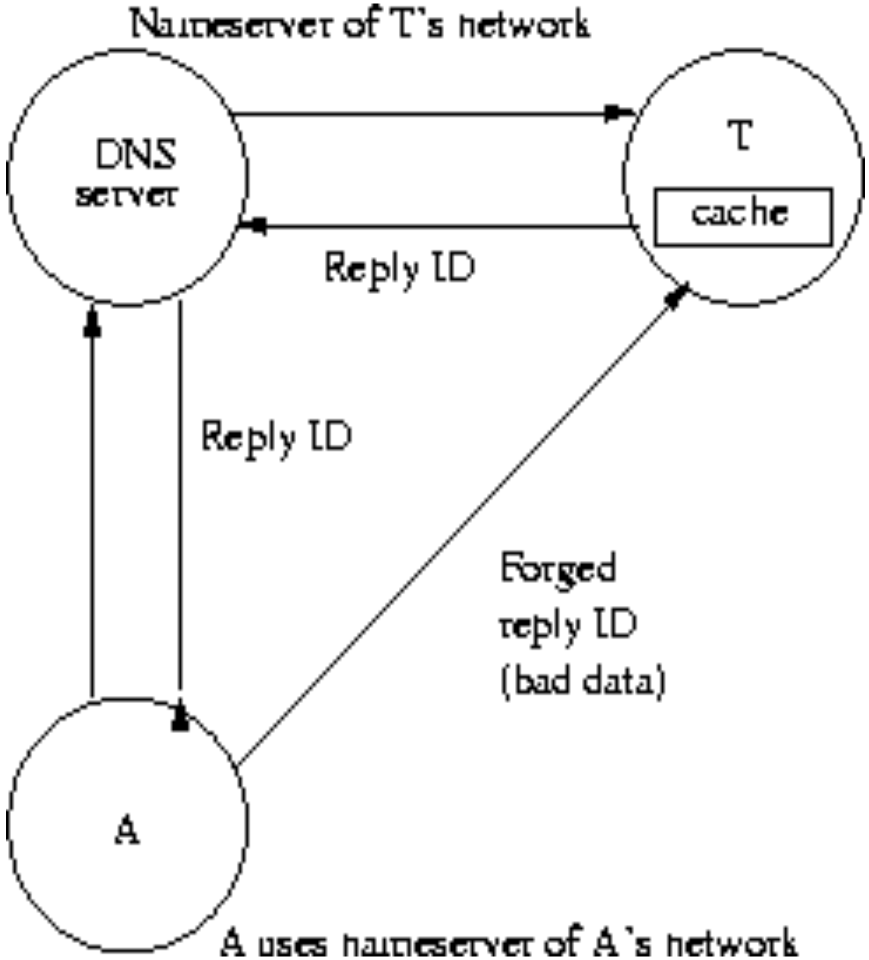
Advanced features

- Caching makes DNS more efficient
  - Store what the server has learnt
  - TTL is important parameter for caching
- Delegation of sub-domains
  - Forward requests to servers in sub-domains
  - Needs to specify a server for each sub-domain
  - The delegated sub-domain is called a zone in the domain. The domain itself is a zone as well.
- Forwarders
  - Reduce offsite DNS traffic
- Reverse mapping: address to name mapping
- Remote control of name server
  - rndc: talk with named through a TCP connection

Security issues

- DNS cache poisoning for Recursive DNS
  - The attacker sends or causes a query to be sent from a victim machine, say a query of kiwibank.co.nz
  - Then it quickly forges an answer and send it to the victim machine before the correct answer arrives.
  - The forged answer will be accepted into the cache of the victim machine for a few hours, depending on TTL of the forged answer.
  - The attacker can take advantage of the poisoned cache to impersonate a trusted host, i.e. kiwibank.co.nz.

DNS cache poisoning


Nameserver of T's network
DNS server
T
cache
Reply ID
Reply ID
Forged reply ID (bad data)
A
A uses nameserver of A's network

Other DNS cache poisoning

- Request: what are the address records for subdomain.attacker.example?
- Attacker's response:
  - Answer:
    - (no response)
    - Authority section: attacker.example. 3600 IN NS ns.kiwi_bank.co.nz.
    - Additional section: ns.kiwi_bank.co.nz. IN A w.x.y.z (an IP address controlled by the attacker)
- A vulnerable server would cache the additional A-record (IP address) for ns.kiwi_bank.co.nz, allowing the attacker to resolve queries to the entire kiwi_bank.co.nz domain.

Solutions for DNS cache poisoning

- Don’t allow clients to use recursive query outside your network
- Use TCP instead of UDP
- Use secure DNS
- Don’t use non-authoritative answers

Bit-squatting

At Black Hat 2011, Artem Dinaburg introduced a “bit-squatting” attack against DNS where an attacker registers new domain names that differ by one bit from popular ones, in order to collect traffic intended for those names when bit errors occur during communications or storage.

DNSSEC has begun to be established as a mechanism for securing information distributed in DNS records. The new DANE protocol being developed in the IETF leverages DNSSEC to enable a domain name owner to inform relying parties which certificates and certificate authorities it trusts.

Secure DNS protocols

- DNS-SEC: DNS Security Extensions
  - origin authentication of DNS data
  - data integrity
  - authenticated denial of existence
  - No confidentiality of data (no encryption)
  - No protection against DDoS (Distributed Denial of Service)
- DNS-TSIG
  - Secret Key Transaction Authentication for DNS

Summary

- The general understanding of how DNS works in a distributed system.
- Recursive and iterative queries
- The impact of the TTL field of a DNS record
- DNS cache poisoning attack