



Basics of DNS

Pensri Arunwatanamongkol

intERLab/AIT and THNIC

Purpose of naming

- Addresses are used to locate objects
- Names are easier to remember than numbers
- You would like to get to the address or other objects using a name
- **DNS provides a mapping from names to resources of several types**

Names and addresses in general

- An address is how you get to an endpoint
 - Typically, hierarchical (for scaling):
 - 950 Main Street, Phayathai, Bangkok, 10120, Thailand
 - 204.152.187.11, +662-381-6003
- A “name” is how an endpoint is referenced
 - Typically, no structurally significant hierarchy
 - “Steve”, “Kyoto”, “nic.or.th”

Naming History

- 1970's ARPANET
 - host.txt maintained by the SRI-NIC
 - pulled from a single machine
 - Problems
 - traffic and load
 - Name collisions
 - Consistency
- DNS created in 1983 by Paul Mockapetris (RFCs 1034 and 1035), modified, updated, and enhanced by a myriad of subsequent RFCs

DNS

- A lookup mechanism for translating objects into other objects
- A globally distributed, loosely coherent, scalable, reliable, dynamic database
- Comprised of three components
 - A “name space”
 - Servers making that name space available
 - Resolvers (clients) which query the servers about the name space

DNS Features: Global Distribution

- Data is maintained locally, but retrievable globally
 - No single computer has all DNS data
- DNS lookups can be performed by any device
- Remote DNS data is locally cacheable to improve performance

DNS Features: Loose Coherency

- The database is always internally consistent
 - Each version of a subset of the database (a zone) has a serial number
 - The serial number is incremented on each database change
- Changes to the master copy of the database are replicated according to timing set by the zone administrator
- Cached data expires according to timeout set by zone administrator

DNS Features: Scalability

- No limit to the size of the database
 - One server has over 20,000,000 names
 - Not a particularly good idea
- No limit to the number of queries
 - 24,000 queries per second handled easily
- Queries distributed among masters, slaves, and caches

DNS Features: Reliability

- Data is replicated
 - Data from master is copied to multiple slaves
- Clients can query
 - Master server
 - Any of the copies at slave servers
- Clients will typically query local caches
- DNS protocols can use either UDP or TCP
 - If UDP, DNS protocol handles retransmission, sequencing, etc.

DNS Features: Dynamicity

- Database can be updated dynamically
 - Add/delete/modify of any record
- Modification of the master database triggers replication
 - Only master can be dynamically updated
 - Creates a single point of failure

DNS Concept: DNS Names

- The namespace needs to be made hierarchical to be able to scale.
- The idea is to name objects based on
 - location (within country, set of organizations, set of companies, etc)
 - unit within that location (company within set of company, etc)
 - object within unit (name of person in company)

Concept: DNS Names contd.

- How names appear in the DNS
 - Fully Qualified Domain Name (FQDN)
 - **WWW.NIC.OR.TH.**
labels separated by dots
- DNS provides a mapping from FQDNs to resources of several types
- Names are used as a key when fetching data in the DNS

Concept: DNS Names contd.

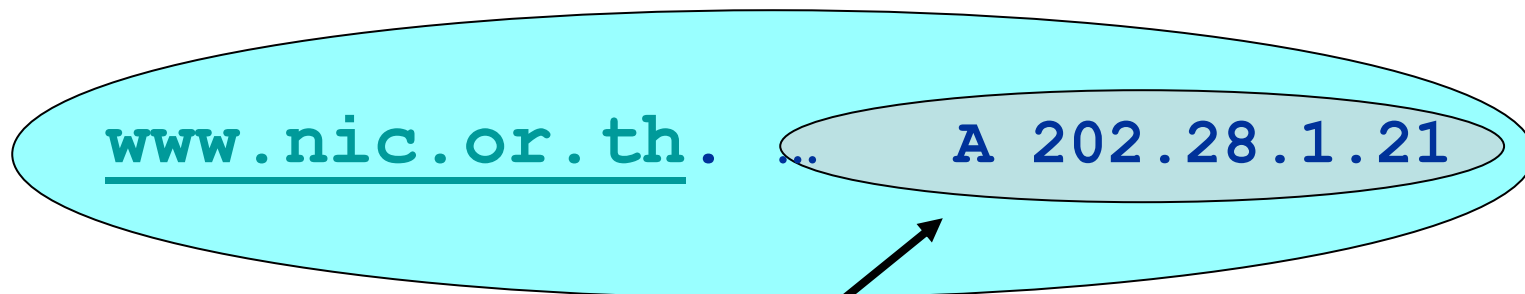


- Domain names can be mapped to a tree.
- New branches at the 'dots'

Concept: Resource Records

- The DNS maps names into data using Resource Records.

Resource Record

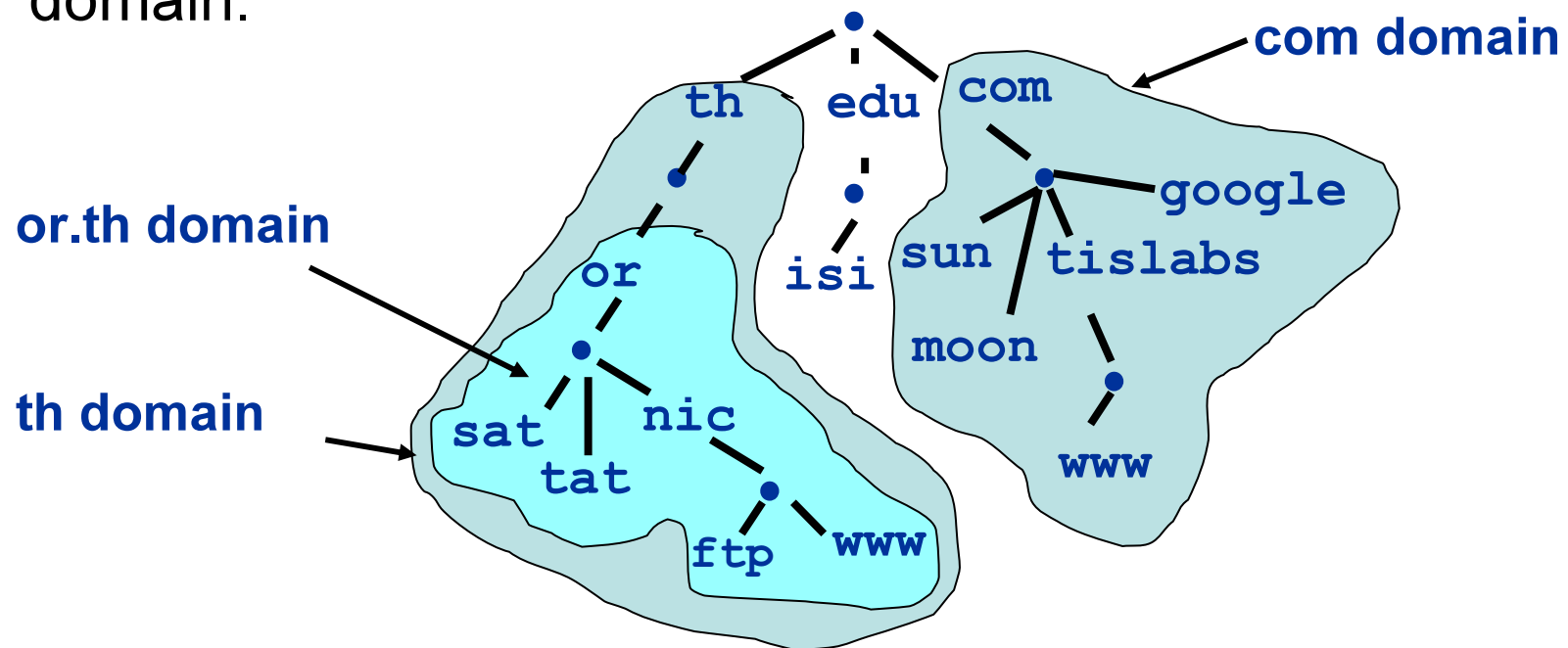


Address Resource

- More detail later

Concept: Domains

- Domains are “namespaces”
- Everything below .com is in the com domain.
- Everything below or.th is in the or.th domain and in the th domain.

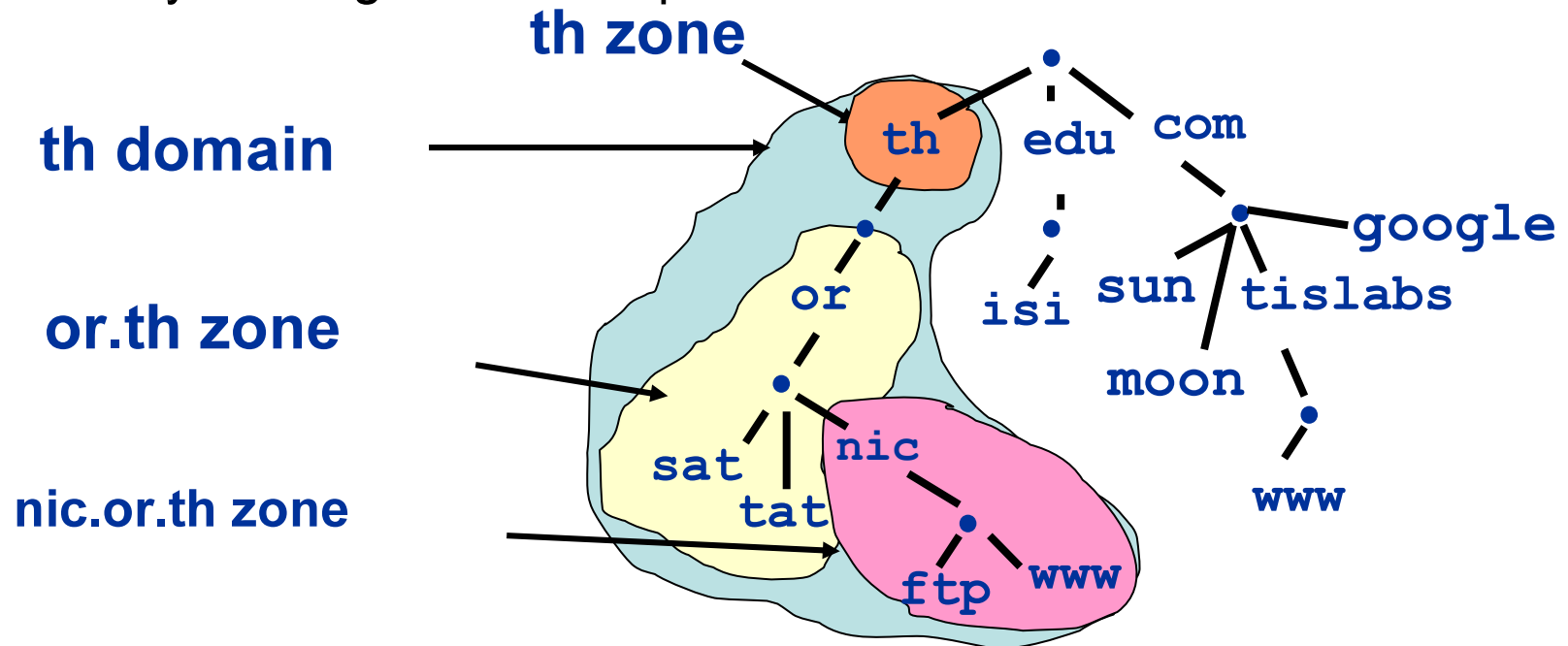


Delegation

- Administrators can create subdomains to group hosts
 - According to geography, organizational affiliation or any other criterion
- An administrator of a domain can delegate responsibility for managing a subdomain to someone else
 - But this isn't required
- The parent domain retains links to the delegated subdomain
 - The parent domain “remembers” who it delegated the subdomain to

Concept: Zones and Delegations

- Zones are “administrative spaces”
- Zone administrators are responsible for portion of a domain’s name space
- Authority is delegated from a parent and to a child

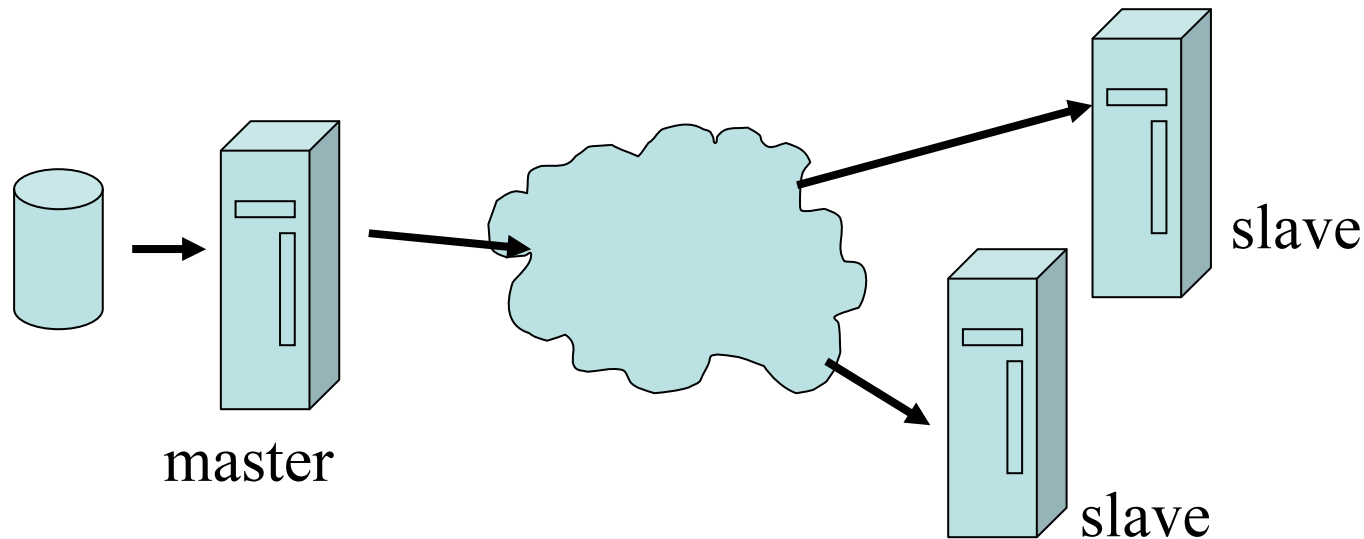


Concept: Name Servers

- Name servers answer 'DNS' questions.
- Several types of name servers
 - Authoritative servers
 - master (primary)
 - slave (secondary)
 - (Caching) recursive servers
 - also caching forwarders
 - Mixture of functionality

Concept: Name Servers contd.

- Authoritative name server
 - Give authoritative answers for one or more zones.
 - The master server normally loads the data from a zone file
 - A slave server normally replicates the data from the master via a zone transfer



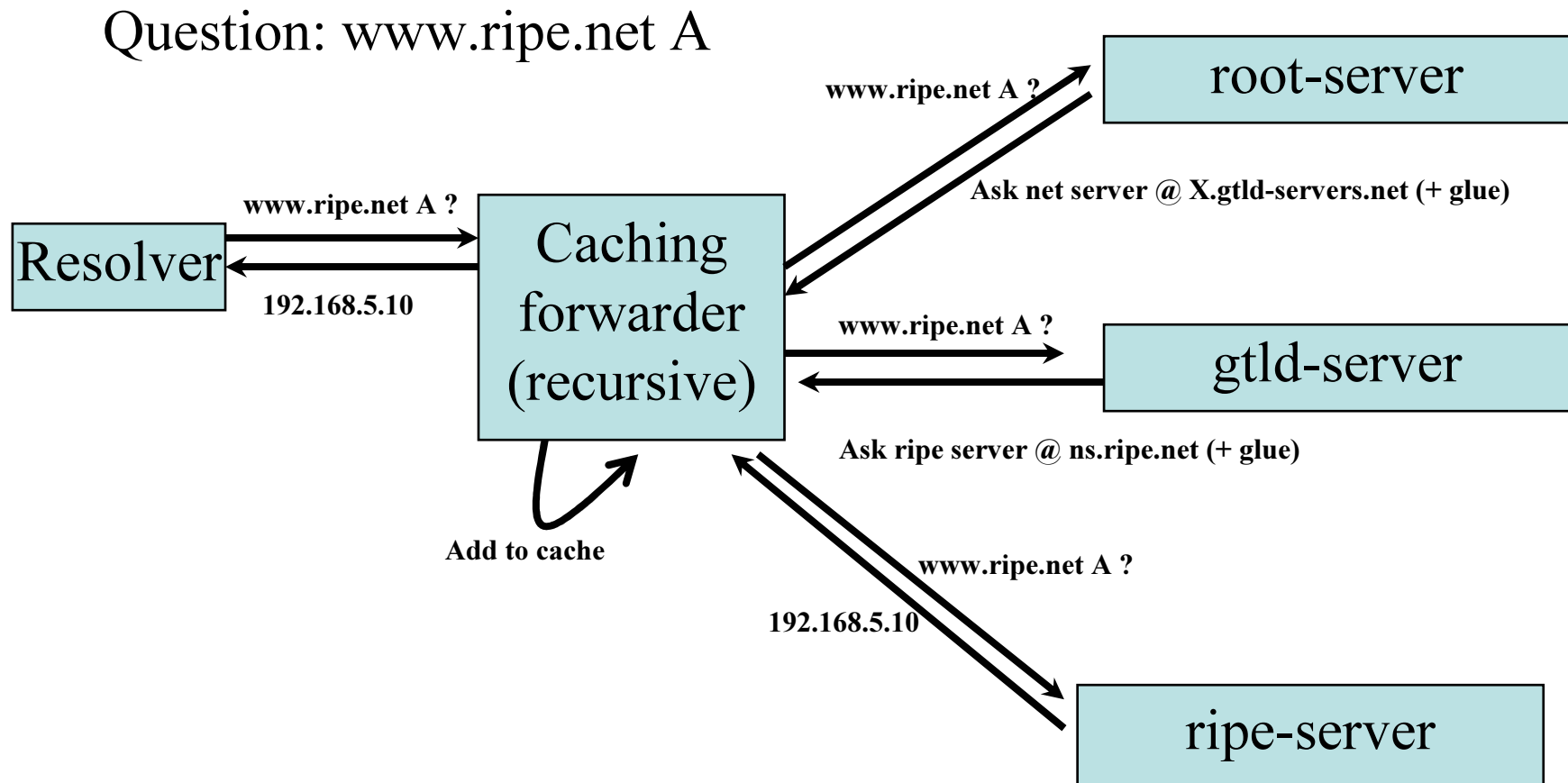
Concept: Name Servers contd.

- Recursive server
 - Recursive servers do the actual lookups; they ask questions to the DNS on behalf of the clients.
 - Answers are obtained from authoritative servers but the answers forwarded to the clients are marked as not authoritative
 - Answers are stored for future reference in the cache

Concept: Resolvers

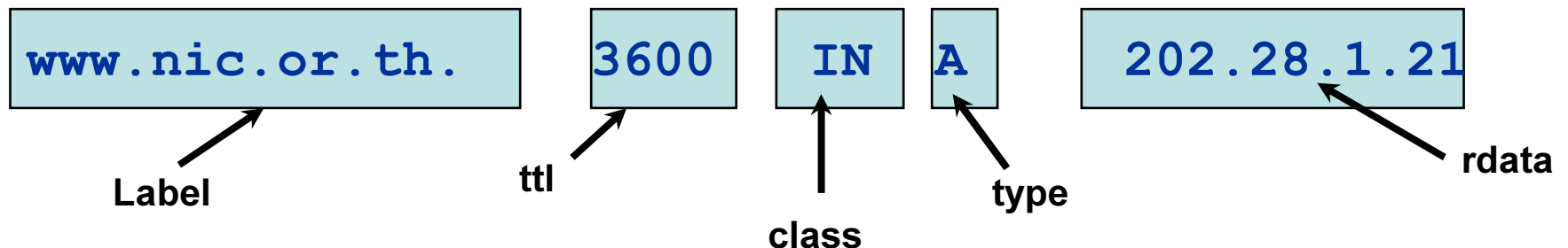
- Resolvers ask the questions to the DNS system on behalf of the application.
- Example applications:
 - Web browsers
 - Email
 - Etc.

Concept: Resolving process & Cache



Concept: Resource Records

- Resource records consist of it's name, it's TTL, it's class, it's type and it's RDATA
- TTL is a timing parameter
- IN class is widest used
- There are multiple types of RR records
- Everything behind the type identifier is called rdata



Example: RRs in a zone file

```
nic.or.th. 7200 IN      SOA      ns.in.th. admin.nic.or.th.
              (
                  2001061501      ; Serial
                  43200      ; Refresh 12 hours
                  14400      ; Retry 4 hours
                  345600      ; Expire 4 days
                  7200      ; Negative cache 2 hours
              )
nic.or.th.      7200      IN      NS      ns.thnic.net.
nic.or.th.      7200      IN      NS      ns.in.th.
```

www.nic.or.th.	3600	IN	A	202.28.1.21
host15.nic.or.th.	2600	IN	A	202.28.1.101

Diagram illustrating the components of a Resource Record (RR) for the example zone file:

- Label:** Points to the domain name (e.g., `host15.nic.or.th.`).
- ttd:** Points to the time-to-live value (e.g., `2600`).
- class:** Points to the class (e.g., `IN`).
- type:** Points to the record type (e.g., `A`).
- rdata:** Points to the record data (e.g., `202.28.1.101`).

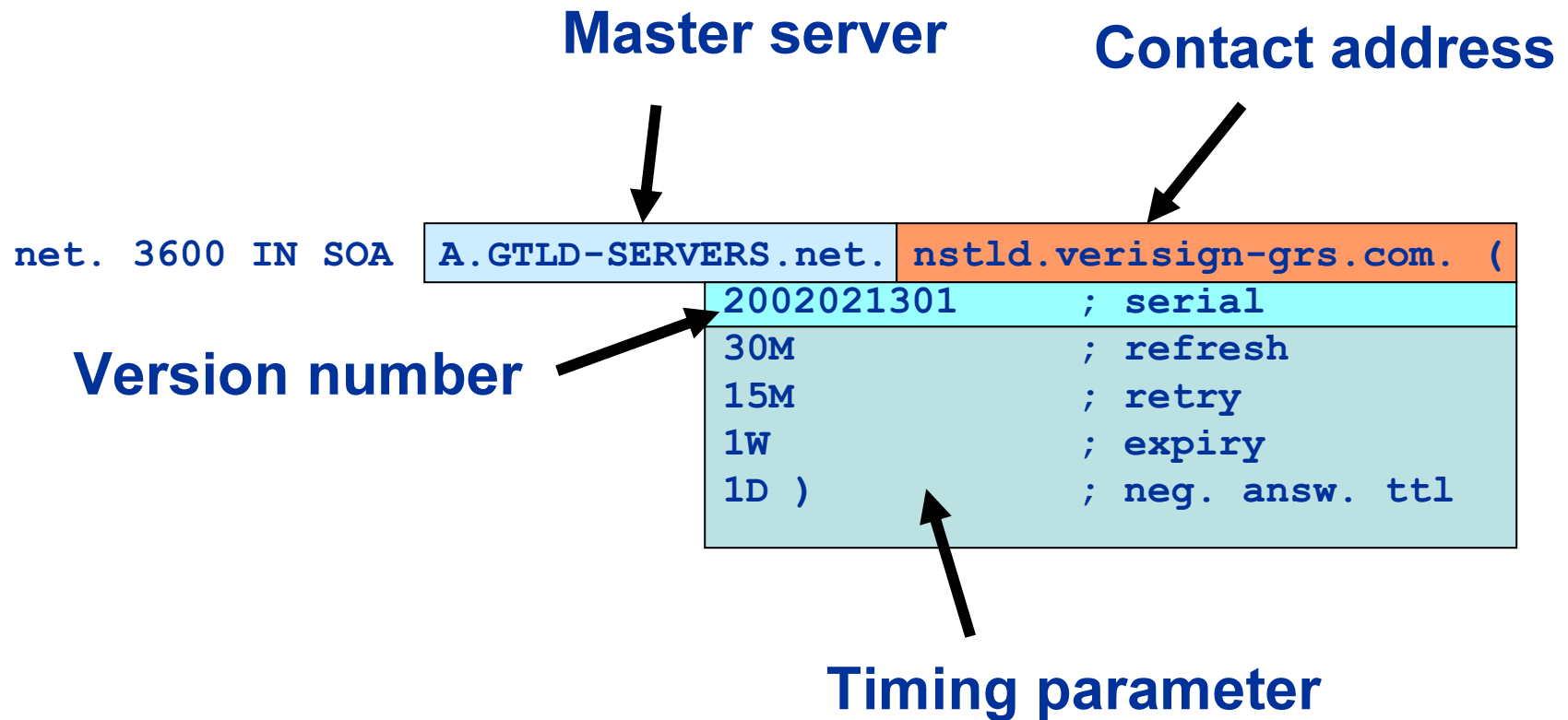
Resource Record: SOA and NS

- The SOA and NS records are used to provide information about the DNS itself.
- The NS indicates where information about a given zone can be found:

```
nic.or.th. 7200 IN  NS  ns.thnic.net.  
nic.or.th. 7200 IN  NS  ns.in.th.
```

- The SOA record provides information about the start of authority, i.e. the top of the zone, also called the APEX.

Resource Record: SOA

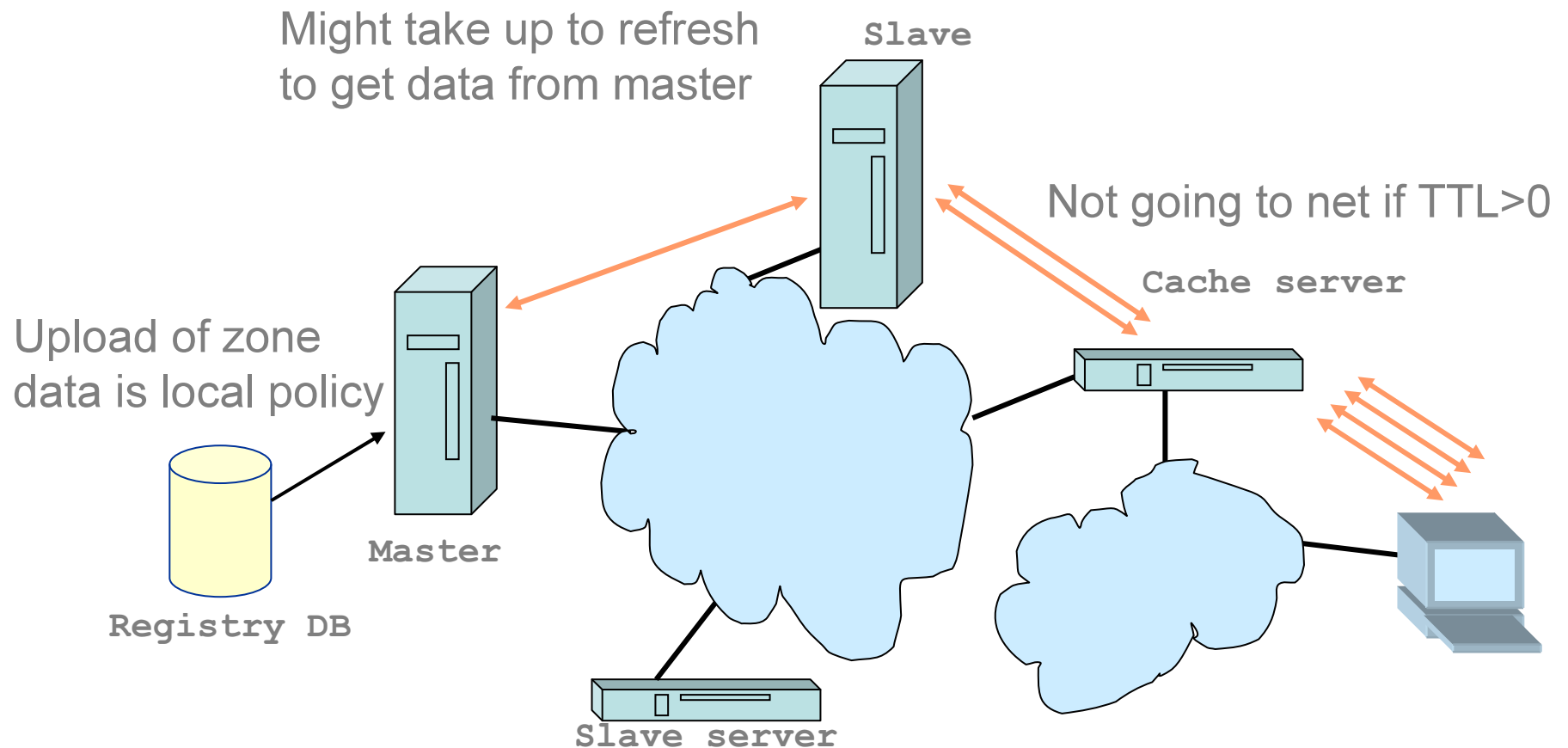


Concept: TTL and other Timers

- TTL is a timer used in caches
 - An indication for how long the data may be reused
 - Data that is expected to be 'stable' can have high TTLs
- SOA timers are used for maintaining consistency between primary and secondary servers

Places where DNS data lives

Changes in DNS do not propagate instantly!



To remember...

- Multiple authoritative servers to distribute load and risk:
 - Put your name servers apart from each other
- Caches to reduce load to authoritative servers and reduce response times
- SOA timers and TTL need to be tuned to needs of zone. Stable data: higher numbers

Selection Slave DNS Servers

- How many name servers for a zone
 - Multiple servers also spread the name resolution load, and improve the overall efficiency of the system by placing servers nearer to the resolvers
 - Minimum 2 in different networks, physical locations
 - 3 or more is recommended
 - Lot of is not very useful

How to find slave service

- Easy to find
 - Generally slave services run automatically
 - Many organizations are willing to do for you
 - Not expensive
- Not easy to find
 - Slave with special configuration
 - Using proprietary system to transfer between master and slave
- Make sure slave operator has tech skills, operates reliable host on reliable network

Restricting Zone Transfers

- Rationale
 - This is a policy issue
 - People have different opinions
- Some people think it's a good idea to restrict zone transfers to particular slaves

Authorizing Zone Transfers

- Two ways to authorize zone transfers from a master server:
 - Using a shared secret (password)
 - By source IP address

Shared Secret

- TSIG
 - Easy to configure
 - Need to agree a password with the people who operate your slave servers

```
key ns1-ns2.zone. {  
    algorithm hmac-md5;  
    secret "APlaceToBe";  
};  
Zone "nic.or.th." {  
    type master;  
    file ....;  
    allow-transfer {  
        key ns1-ns2.zone.;  
    };  
};
```

IP Address

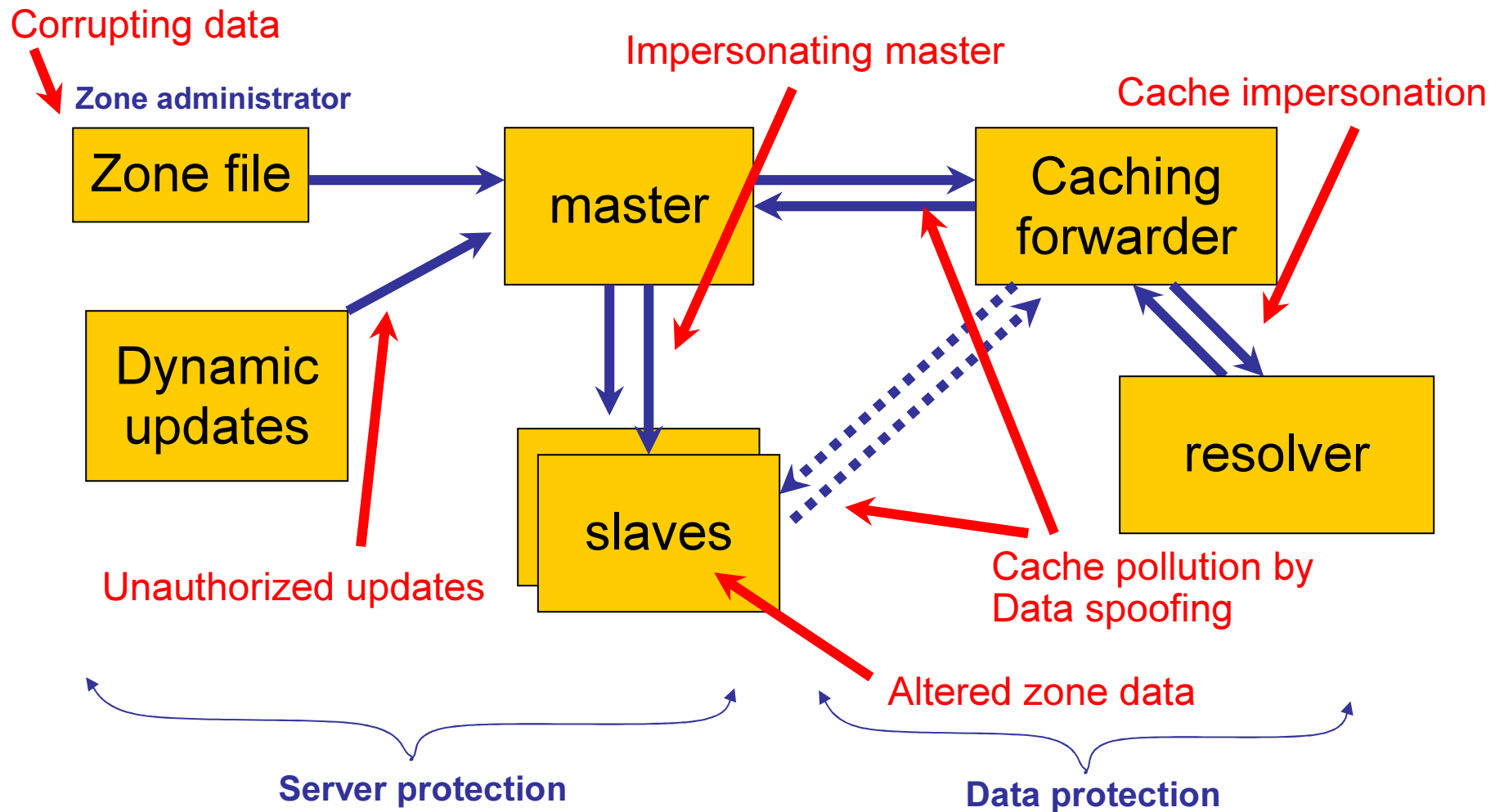
- Checking the source address is a weak way to authenticate anything, but sometimes it's better than nothing
- ACLs for Zone transfers

```
Zone "nic.or.th" {  
    Type master;  
    File "master/db.nic.or.th";  
    Allow-transfer {  
        192.41.170.2;  
        202.28.1.21;  
    };  
};
```

DNSSEC

- Why do we need DNSSEC
- What does DNSSEC provide
- How does DNSSEC work

DNS Vulnerabilities



DNS Protocol Vulnerability

- DNS data can be spoofed and corrupted between master server and resolver or forwarder
- The DNS protocol does not allow you to check the validity of DNS data
 - Exploited by bugs in resolver implementation (predictable transaction ID)
 - Polluted caching forwarders can cause harm for quite some time (TTL)
 - Corrupted DNS data might end up in caches and stay there for a long time
- How does a slave (secondary) know it is talking to the proper master (primary)?

DNSSEC protects

DNSSEC protects against data spoofing and corruption

- TSIG/SIG0: provides mechanisms to authenticate communication between servers
- DNSKEY/RRSIG/NSEC: provides mechanisms to establish authenticity and integrity of data
- DS: provides a mechanism to delegate trust to public keys of third parties
- A secure DNS will be used as an infrastructure with public keys
 - However it is NOT a PKI

Core Elements

DNSSEC is based on Public Key Cryptography

- Key pair: a private and a public key
- The private key can be used to create signatures
- The signature can be 'validated' with the public key.
- If the signature over a message validates the message must have been signed by the holder of the private keys.
- The message is not encrypted

A little about the IANA

- Jon Postel first started keeping track of numbers around 1969. (In a notebook)
- Added staff in the early 80's (Joyce Reynolds, now RFC editor)
- Started being called “the IANA” in the late 80's
- Became a function of ICANN in 1998

What does the IANA do?

- Registers Unique Identifiers and related information for use on the Internet.
- IP Addresses, AS #'s, Port #'s, many others... including TLD's.

Country Code Top Level Domains

- Referred to as a ccTLD
- codes are assigned from a table known as [ISO-3166-1](#)
- IANA current practices summary is called ICP1
 - <http://www.icann.org/icp/icp-1.htm>

Services for ccTLD's

- Mainly related to changes in the root zone or whois data.
 - Change of DNS servers
 - Change of IP addresses for DNS servers
 - Change of POC details (Point Of Contact)

Whois Data of ccTLDs

- <http://www.iana.org/root-whois/th.htm>

.th – Thailand

- <http://www.iana.org/root-whois/jp.htm>

.jp - Japan

Policy and ICANN

- ccNSO: <http://ccnso.icann.org/>
- “The ccNSO is the policy development body for a narrow range of global ccTLD issues within the ICANN structure”

Joining?

- <http://ccnso.icann.org/applications/clean-app.shtml>

Useful links

- ICP1 (IANA practices)
 - <http://www.icann.org/icp/icp-1.htm>
- Format, Content, and Technical Requirements for Requests to Change TLD Contact Information
 - <http://www.iana.org/cctld/contact-change-requests-09may01.htm>
- Template for changes
 - <http://www.iana.org/tld/cctld-template.txt>
- Step by step procedure
 - <http://www.iana.org/cctld/nameserver-change-procedures-13may03.htm>

A little about IDN

- IDN standard RFCs
 - RFC 3490, RFC 3491, RFC 3492
- ICANN issued a set of guidelines for the Implementation of Internationalized Domain Names in June of 2003
- Developed collaboratively by ICANN and leading Internationalized Domain Names (IDN) registries

<http://www.icann.org/general/idn-guidelines-20jun03.htm>

IDN issues

- Naming in some local languages generally use long words (more characters required)
 - ASCII: name not longer than 63
 - Thai: Maximum characters that can be used as a domain name is about 30-45. (Depends on naming)
- ASCII Compatible Encoding (ACE) is not human readable. More difficult to edit Zone file.

IDN Applications

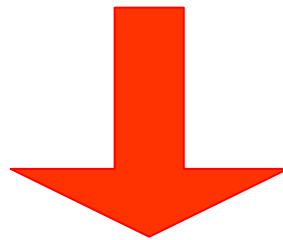
- Browsers
 - Netscape, Mozilla, Opera, Safari, Camino, Epiphany, Firefox, Konqueror and others
- E-Mail
 - FoxMail
- IDN SDKs and Language support
 - GNU libidn, JPNIC, VeriSign

ENUM

- ENUM is a mechanism for mapping telephone (E.164) numbers to Internet resource address(es)
 - 1 to many
- Internet resource address(es) are specified as URI(s).
- Mapping is
 - Registered using NAPTR records in DNS
 - referred to by DNS lookup
- End users (Applications) can select URI(s) according to their preference

ENUM Mapping

+66 2 524 66 17



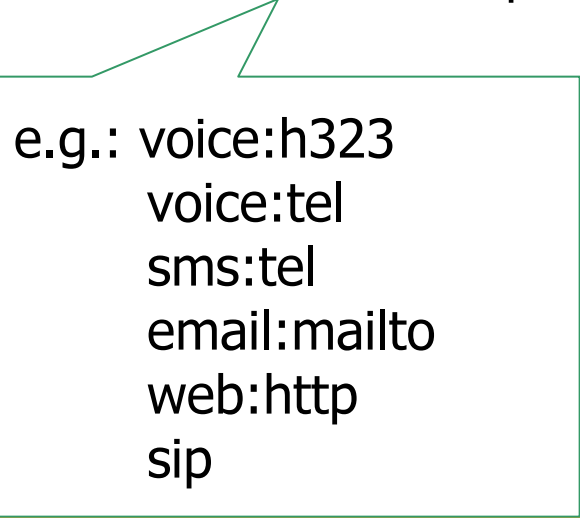
7.1.6.6.4.2.5.2.6.6.e164.arpa

ENUM Resolution

This domain-name is used to request NAPTR records from the DNS Database which may contain the end result or, if the flags field is blank, produces new keys in the form of domain-names from the DNS.

\$ORIGIN 7.1.6.6.4.2.5.2.6.6.e164.arpa.

;	type	order	pref	flag	enumservice	regular expression	replacement
IN NAPTR	10	100	"u"	"E2U+voice:sip"	"!^.*\$!sip:pensri@sip.ait.ac.th!"	.	



e.g.: voice:h323
voice:tel
sms:tel
email:mailto
web:http
sip



Thank You!