

Internet Engineering

241-461

Robert Elz

kre@munnari.OZ.AU

kre@coe.psu.ac.th

<http://fivedots.coe.psu.ac.th/~kre>

TCP Header

Source Port	Destination Port		
Sequence Number			
Acknowledgment Number			
Header Len	Reserved	Urgent	Window Size
Checksum		Urgent Pointer	

- ... Contains some data not examined yet

TCP Header

Source Port	Destination Port		
Sequence Number			
Acknowledgment Number			
Header Len	Reserved	Urgent	Window Size
Checksum		Urgent Pointer	

- ... Contains some data not examined yet

◊ Reset

- The TCP error message
- Sent whenever TCP error occurs
 - > Segment in invalid state
 - > NOT missing packet
 - > NOT out of order packet
 - > NOT duplicate packet

TCP Timeouts

◊ Recall ...

Ngx

TCP Timeouts

◊ Recall ...

- retransmit
- after timeout

TCP Timeouts

◊ Recall ...

Ngx

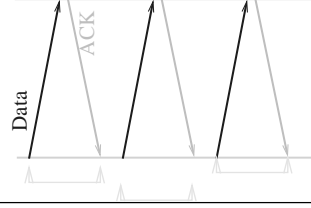
TCP Timeouts

- ◊ Recall ...
 - retransmit
 - after timeout
- ◊ How long?
 - 1 second ?
 - 2 seconds ?
 - 3 seconds ?
 - Long enough
 - ▷ no unnecessary retransmit
 - ▷ Short enough
 - ▷ avoid delays
- ◊ Time to wait varies
 - Different peers (near or distant)
 - Different network state (idle or congested)
- ◊ Measure Round Trip Time

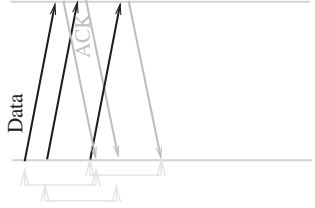
Round Trip Time Estimation

- ◊ Useful to know when to retransmit
 - if have not received ACK within the RTT
 - * (plus a bit)
 - ▷ then assume packet lost
- ◊ But how to measure the RTT?
 - Measure delay between packet and its ACK
 - ▷ easy
- But
 - ▷ Send packet
 - * wait ... wait ... wait (nothing)
 - ▷ Retransmit packet
 - ▷ ACK arrives
- Which packet was acknowledged?
 - ▷ The initial packet
 - * acknowledged slower than expected
 - ▷ Or the retransmit?

TCP RTT Measurement



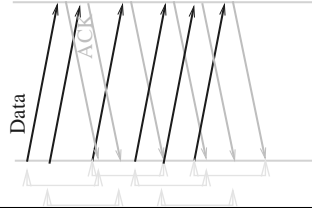
TCP RTT Measurement



TCP RTT Measurement



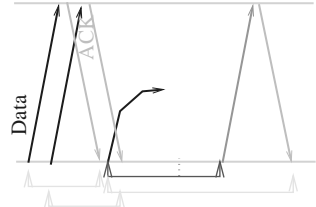
TCP RTT Measurement (2)



TCP RTT Measurement (2)



TCP RTT Measurement (3)



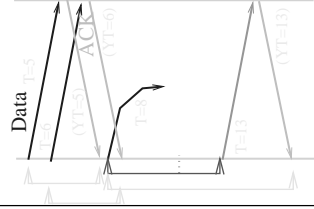
TCP RTT Measurement (3)



TCP Timestamp Option

- ◇ Each TCP can add timestamp option
- ◇ Peer TCP sends back timestamp received
 - with each ACK
- ◇ Allows TCP to determine which packet was ACK'd
- ◇ Better than that
 - no need to remember when packets were sent
 - returning timestamp contains that information
- ◇ Also used for long delayed old packet detection
 - extends the sequence number space

TCP RTT Measurement (tstamp)



TCP RTT Measurement (tstamp)

The Domain Name System

- ◊ Kurose & Ross: Computer Networking
 - Chapter 2 (2.5)

James F. Kurose & Keith W. Ross
Computer Networking

A Top-Down Approach Featuring the Internet
(2nd, 3rd or 4th Edition)
Addison Wesley

Contents

- ◊ The Domain Name System
 - Domains
 - Zones
- ◊ The DNS Database
- ◊ DNS Protocols
- ◊ DNS Message Formats
- ◊ DNS Limits
- ◊ Zone Transfer
- ◊ Mapping Addresses to Names

Domain Name System

- ◊ Translates Names to Addresses
 - (Finds where something is located)
- ◊ Also translates Addresses to Names
 - (Finds what is located there)
- ◊ Does not discover names
 - We have to know what name we want
 - The DNS is NOT a directory service
- ◊ Does provide other information
 - about known names

DNS Terminology

The DNS Name

Tr

(the

dat

DNS Terminology

The DNS Name

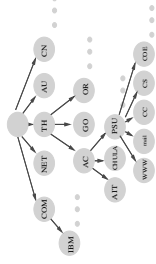
Tr

- ◇ Domain
 - Fully Qualified Domain Name
- ◇ Zone
 - Delegation
- ◇ Resource Record
 - Resource Record Set
- ◇ DNS Server
- ◇ DNS Cache
- ◇ DNS Resolver
 - Stub Resolver
 - Back End Resolver

DNS

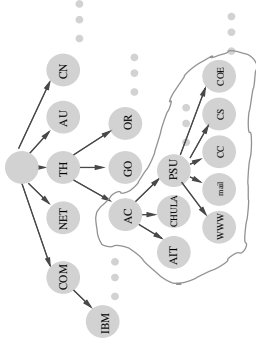
Sc

The DNS Name Tree



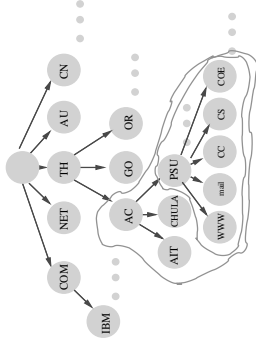
- ◇ A tree of domains
- ◇ The root at the top of the tree
 - There is exactly one Root
 - The root has no name - empty label
- ◇ Read from the bottom up
 - Produces Left -> Right answer
- ◇ Insert a ' ' each time a link is crossed
- ◇ Name that ends in ' ' (and empty root domain)
 - Fully Qualified Domain Name (FQDN)

A Domain

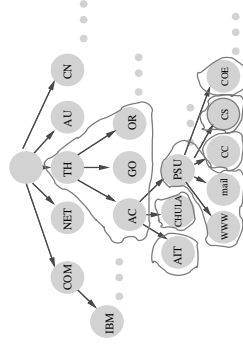


- ◇ Each domain is
 - A sub-domain of some other domain
 - Except the root domain
- ◇ That other domain is called its parent
- ◇ Any domain can have sub-domains
 - Sub-domains are part of the domain
- ◇ Domain named by name of its highest node
 - All subdomain names end .domain.name.

A Domain



Zone



- ◇ Domains can be divided into Zones
- ◇ Division at any sub-domain boundary
 - If domain is excluded
 - Entire domain is excluded
 - Cannot cross existing zone boundaries
 - But zone division can be removed
 - Domain is either
 - in a zone of its own
 - or in same zone as its parent
- ◇ Zones are independant DNS units

Zones and Domains

- ◇ A Zone is almost a Domain
 - Both named by domain name
 - ▷ Of highest node in tree contained
 - If no sub-domains are split
 - ▷ Domain and Zone are the same thing
 - But if a new zone is formed
 - ▷ Domain remains unchanged
 - ▷ The two zones are independant
 - Every node is in exactly one zone
 - ▷ Usually in many domains

Contents

- ◇ The Domain Name System
- ◇ The DNS Database
 - Resource Records
 - RR Sets
- ◇ DNS Protocols
- ◇ DNS Message Formats
- ◇ DNS Limits
- ◇ Zone Transfer
- ◇ Mapping Addresses to Names

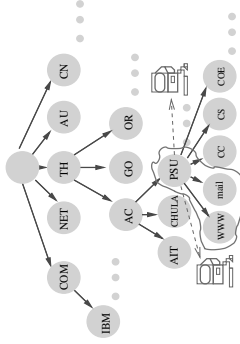
DNS database

- ◇ Keyed by domain name
- ◇ Contains Resource Records (RRs)
- ◇ RRs have a Type and a Class
- ◇ Class is not used much
 - Internet Class is almost everything
- ◇ Type specifies what kind of data is in the RR
 - A - Address (IPv4)
 - AAAA - 128 bit (IPv6) address
 - MX - Mail Exchange
 - LOC - Location
 - (many)

RRSets

- ◇ Group of RRs with
 - same class type & domain name
 - are an RRset
- ◇ RRset has a Time to Live
 - measured in seconds
- ◇ DNS Query gives domain name
 - ▷ required type (& class)
 - Gets RRset and its TTL
 - Or gets No Such Name
 - Or gets No Data of that Type
- ◇ Answer can be remembered
 - ▷ until TTL expires
 - Saves on DNS queries

DNS Servers



- ◇ Each zone has servers
 - to answer queries about it
- ◇ Each server has information
 - about its zone(s)
- ◇ There are redundant servers for reliability
- ◇ Servers can serve many zones
 - Related or unrelated zones

Contents

- ◇ The Domain Name System
- ◇ The DNS Database
- ◇ DNS Protocols
- ◇ DNS Message Formats
- ◇ DNS Limits
- ◇ Zone Transfer
- ◇ Mapping Addresses to Names

DNS Protocols

- ◊ UDP based (in general)
- ◊ Mostly lookup protocol
 - Don't care if transactions are repeated
 - Don't need to deal with duplicate packets
- ◊ Lost packets handled by retransmit
 - Not necessarily to same server

A DNS Query

- ◊ Given a Domain Name, request information
- ◊ Information is typed
- ◊ Can ask for
 - Address (A)
 - Or IPv6 Address (AAAA)
 - Name Server (NS)
 - Mail eXchange (MX)
 - and much more.
- ◊ Assuming no other information, the query
 - Starts at the root of the tree
 - Gets referred to next lower subdomain
 - Until it reaches the domain of the name desired
- ◊ Each server replies with nameserver names
 - names of servers for the next lower sub-domain
 - (and usually their addresses)

Client Queries Root Server



Root Server Reply



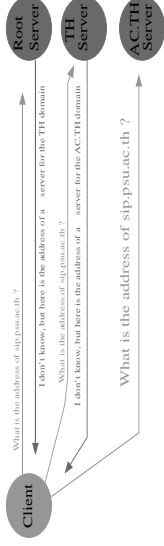
Client Queries TH Server



TH Server Reply

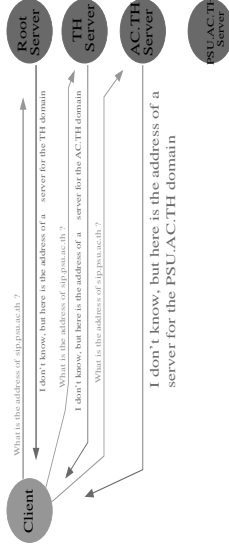


Client Queries AC.TH Server



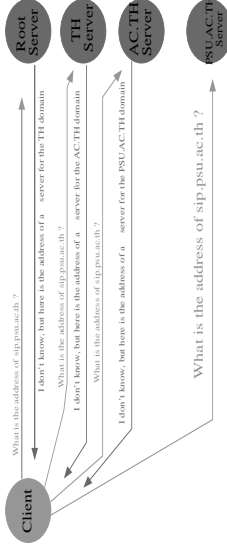
- ◊ Client called Resolver
 - because its job is to
 - resolve domain names
 - into the required information

AC.TH Server Reply

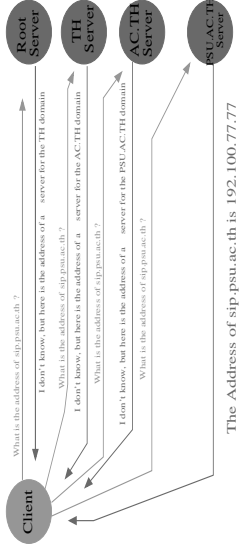


- ◊ Resolver is a library function
 - Included as part of every application
 - That requires name resolution

Query PSU.AC.TH Server

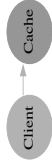


PSU.AC.TH Server Reply



- ◇ This is a complex process
 - Much code required
 - Many packets sent and received
 - All data lost when application exits

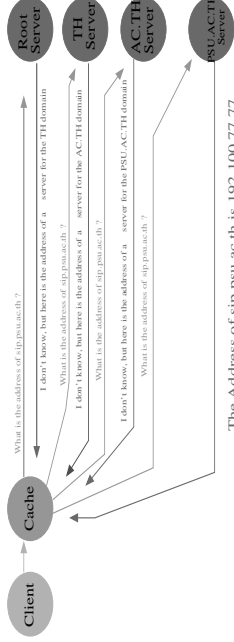
The Way it Really Works



What is the address of sip.psu.ac.th ?

- ◇ Client contains stub resolver
 - Able to ask question
 - Expects to get final answer returned
- ◇ Stub Resolver uses DNS Cache
 - or Back-End Resolver
 - To do the resolution work for it.

Cache Finds Answer

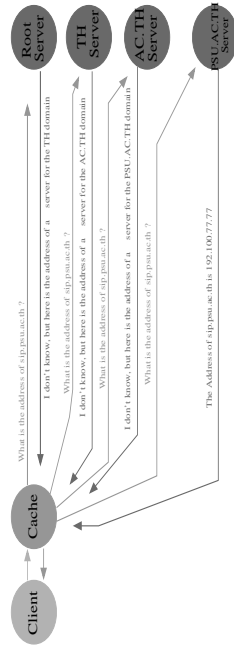


What is the address of sip.psu.ac.th ?

(if required)

- ◇ Back End Resolver (Cache) resolves name
 - Using its memory of earlier answers

Cache Tells Client



What is the address of sip.psu.ac.th ?
The Address of sip.psu.ac.th is 192.100.77.77

- ◇ **Cache remembers each answer**
 - ▷ in case it is required again
 - Every purple arrow is an answer
 - Each of these is remembered
- ◇ **Query for www.ait.ac.th**
 - starts at AC.TH server