The Domain Name System

Antonio Carzaniga

Faculty of Informatics
University of Lugano

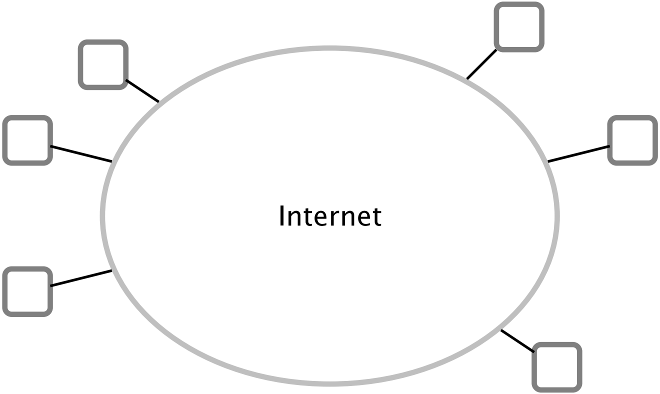
October 8, 2014

Outline

- IP addresses and host names
- DNS architecture
- DNS process
- DNS requests/replies

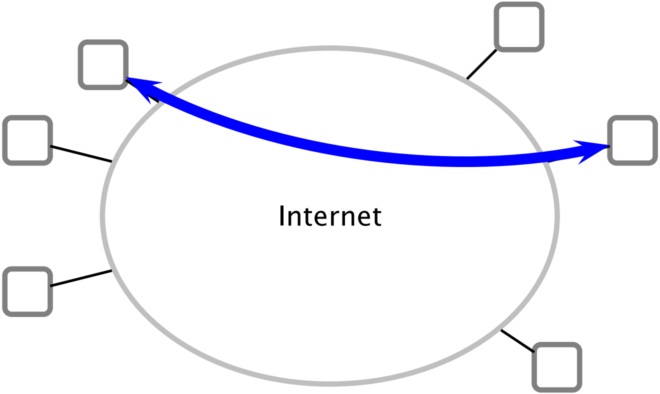
End Systems

Internet applications involve *end system communication*



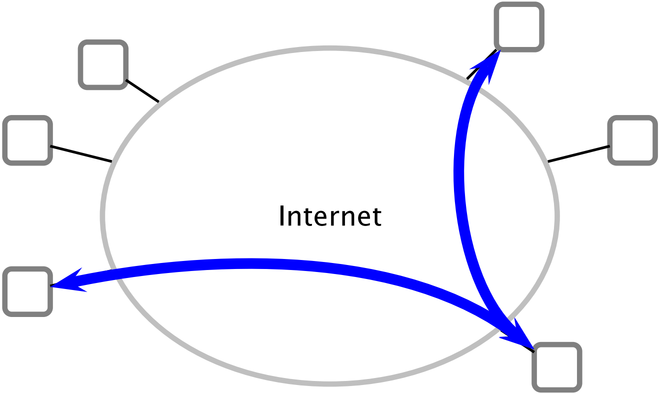
End Systems

Internet applications involve *end system communication*



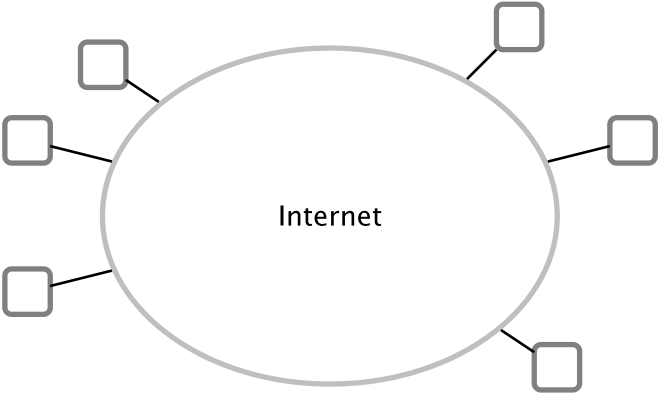
End Systems

Internet applications involve *end system communication*



End Systems

Internet applications involve *end system communication*



How does one end system *address* another end system?

IP Addresses

- An end system is *identified* and *addressed* by its *IP address*

IP Addresses

- An end system is *identified* and *addressed* by its *IP address*
  - 32 bits (4 bytes) in IPv4
  - e.g., 195.176.181.10

IP Addresses

- An end system is *identified* and *addressed* by its *IP address*
  - 32 bits (4 bytes) in IPv4
  - e.g., 195.176.181.10
  - 128 bits (16 bytes) in IPv6
  - e.g., fe80::211:43ff:fecd:30f5/64

IP Addresses

- An end system is *identified* and *addressed* by its *IP address*
  - 32 bits (4 bytes) in IPv4
  - e.g., 195.176.181.10
  - 128 bits (16 bytes) in IPv6
  - e.g., fe80::211:43ff:fecd:30f5/64

- *Advantages*

IP Addresses

- An end system is *identified* and *addressed* by its *IP address*
  - 32 bits (4 bytes) in IPv4
  - e.g., 195.176.181.10
  - 128 bits (16 bytes) in IPv6
  - e.g., fe80::211:43ff:fecd:30f5/64

- *Advantages*
  - computers (e.g., routers) are good at processing bits
  - especially in small packs of a size that is a power of two

IP Addresses

- An end system is *identified* and *addressed* by its *IP address*
  - 32 bits (4 bytes) in IPv4
  - e.g., 195.176.181.10
  - 128 bits (16 bytes) in IPv6
  - e.g., fe80::211:43ff:fecd:30f5/64
- *Advantages*
  - computers (e.g., routers) are good at processing bits
  - especially in small packs of a size that is a power of two
- *Disadvantages*

IP Addresses

- An end system is *identified* and *addressed* by its *IP address*
  - 32 bits (4 bytes) in IPv4
  - e.g., 195.176.181.10
  - 128 bits (16 bytes) in IPv6
  - e.g., fe80::211:43ff:fecd:30f5/64

- *Advantages*
  - computers (e.g., routers) are good at processing bits
  - especially in small packs of a size that is a power of two

- *Disadvantages*
  - not practical for use by *people*
  - i.e., not mnemonic
  - e.g., "look it up on 64.233.183.104!"

Host Names

- Goal: help the human users of the Internet
  - human-readable, mnemonic addresses, aliases

Host Names

- Goal: help the human users of the Internet
  - human-readable, mnemonic addresses, aliases
- Solution: *domain name system (DNS)*

Host Names

- Goal: help the human users of the Internet
  - human-readable, mnemonic addresses, aliases
- Solution: *domain name system (DNS)*
  - host names
  - e.g., www.google.com

Host Names

- Goal: help the human users of the Internet
  - human-readable, mnemonic addresses, aliases
- Solution: *domain name system (DNS)*
  - host names
  - e.g., www.google.com
- Primary function of the domain name system

$$name \rightarrow IP\ address$$

maps a name to an IP address

Host Names

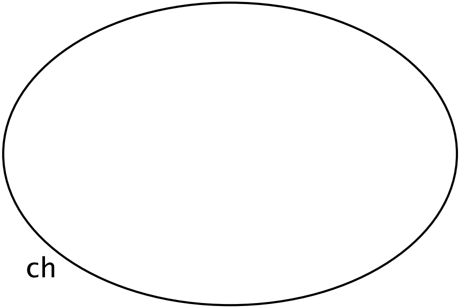
- E.g., `atelier.inf.usi.ch`

Host Names

- E.g., `atelier.inf.usi.ch`
- Hierarchical name space

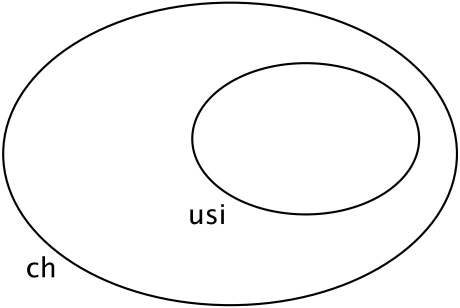
Host Names

- E.g., `atelier.inf.usi.ch`
- Hierarchical name space
- Top-level domain



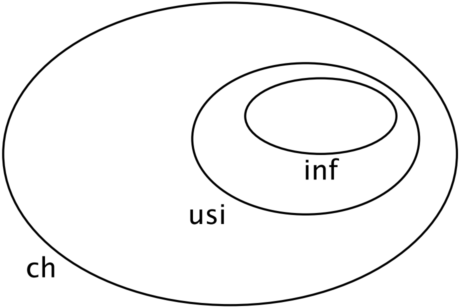
Host Names

- E.g., `atelier.inf.usi.ch`
- Hierarchical name space
- Top-level domain, ...



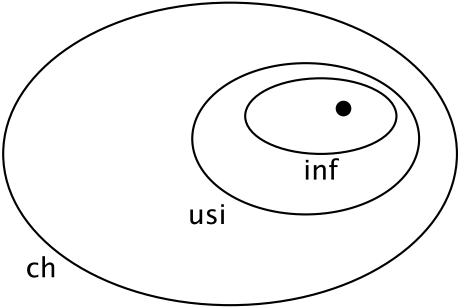
Host Names

- E.g., `atelier.inf.usi.ch`
- Hierarchical name space
- Top-level domain, ...



Host Names

- E.g., `atelier.inf.usi.ch`
- Hierarchical name space
- Top-level domain, ...



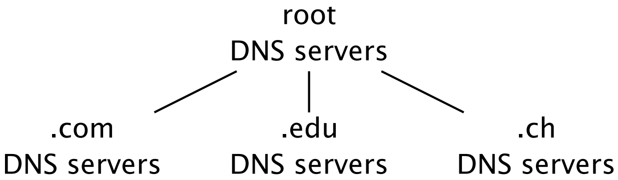
Architecture of DNS

Architecture of DNS

- Hierarchical architecture that mirrors the hierarchical structure of the namespace

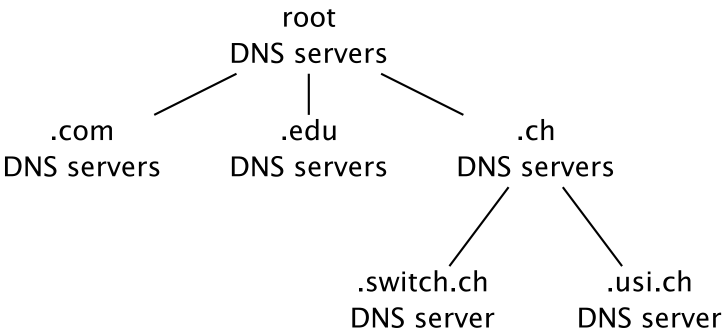
Architecture of DNS

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



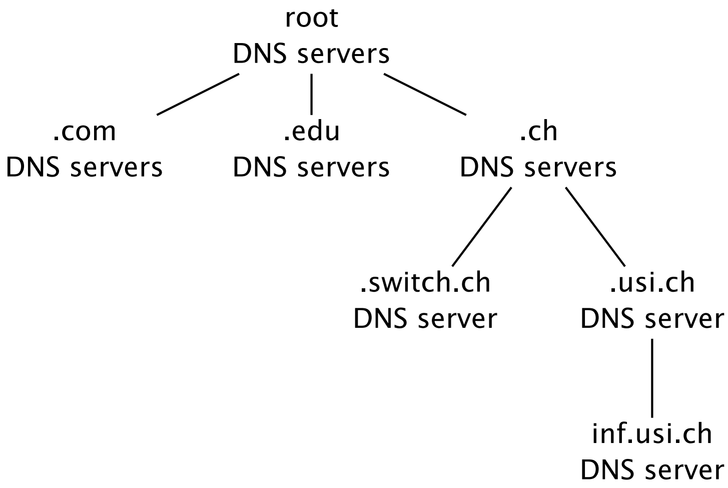
Architecture of DNS

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



Architecture of DNS

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



DNS Architecture

- *Root servers:* 13 “root” DNS servers know where the top-level servers are (labeled A through M)
  - see http://www.root-servers.org

DNS Architecture

- *Root servers:* 13 “root” DNS servers know where the top-level servers are (labeled A through M)
  - see http://www.root-servers.org

- *Top-level domain servers:* each one is associated with a top-level domain (e.g., .com, .edu, .ch, .org, .tv)

DNS Architecture

- *Root servers:* 13 “root” DNS servers know where the top-level servers are (labeled A through M)
  - see http://www.root-servers.org

- *Top-level domain servers:* each one is associated with a top-level domain (e.g., .com, .edu, .ch, .org, .tv)

- *Authoritative servers:* for each domain, there is an authoritative DNS server that holds the map of publicly-accessible hosts within that domain

DNS Architecture

- *Root servers:* 13 “root” DNS servers know where the top-level servers are (labeled A through M)
  - see http://www.root-servers.org

- *Top-level domain servers:* each one is associated with a top-level domain (e.g., .com, .edu, .ch, .org, .tv)

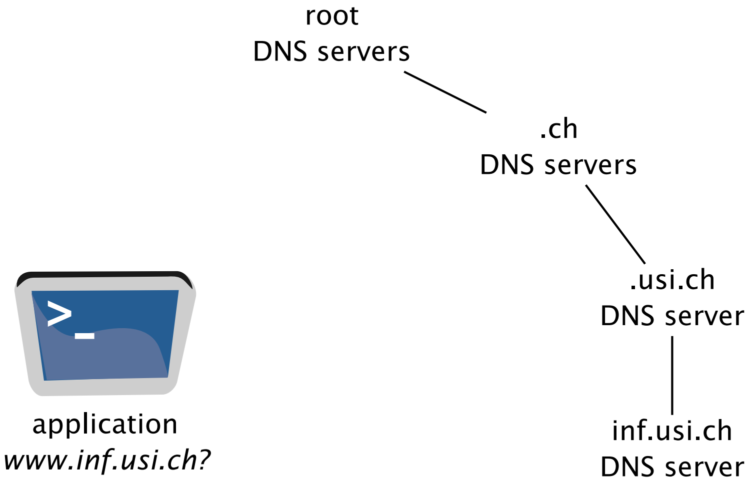
- *Authoritative servers:* for each domain, there is an authoritative DNS server that holds the map of publicly-accessible hosts within that domain

- Most root “servers” as well as servers at lower levels are themselves implemenented by a distributed set of machines

How DNS Works

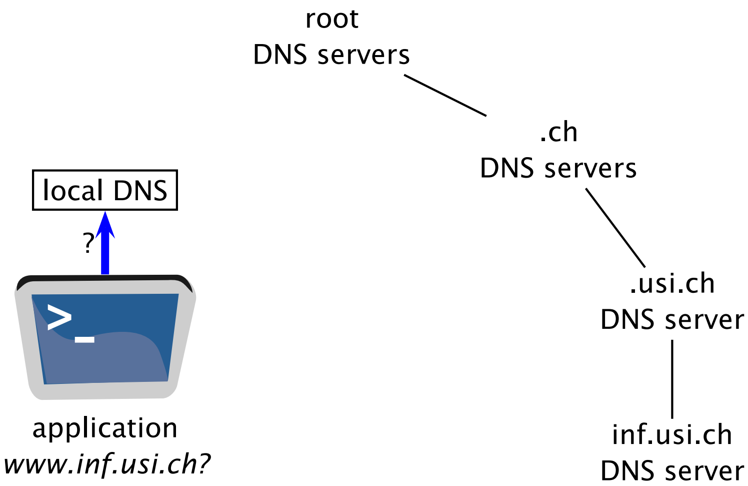
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



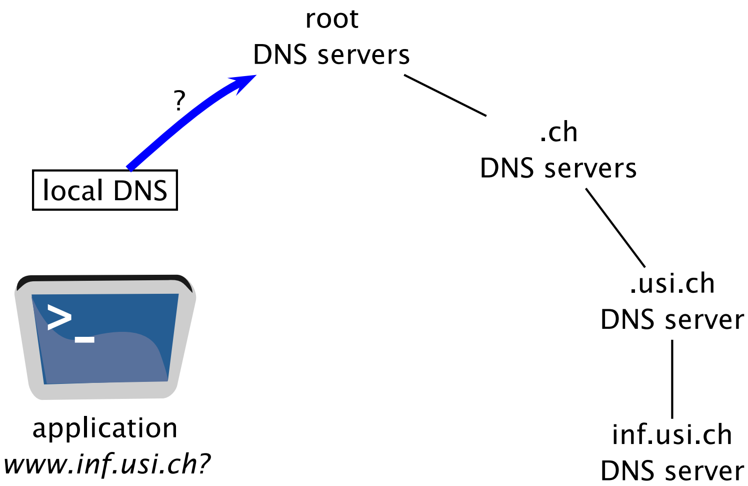
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



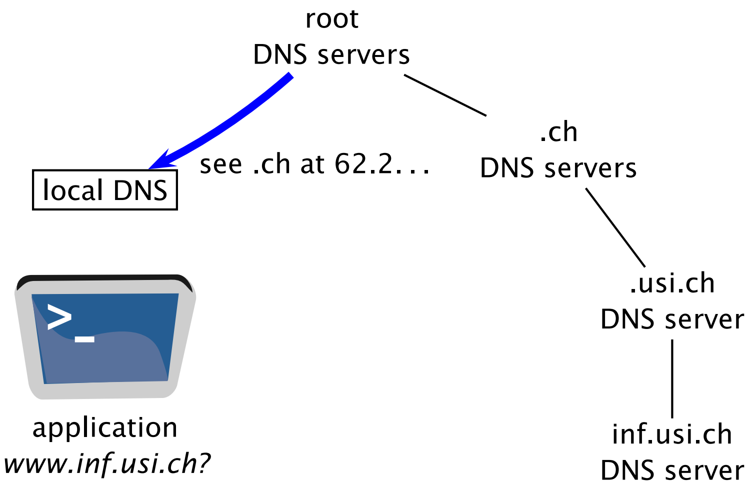
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



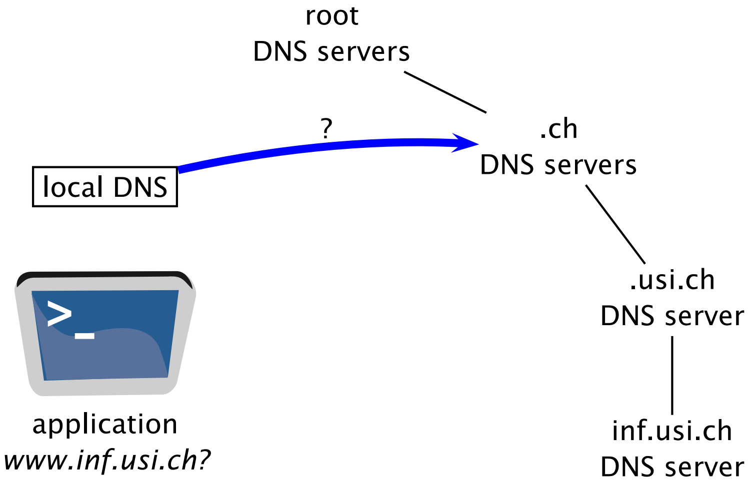
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



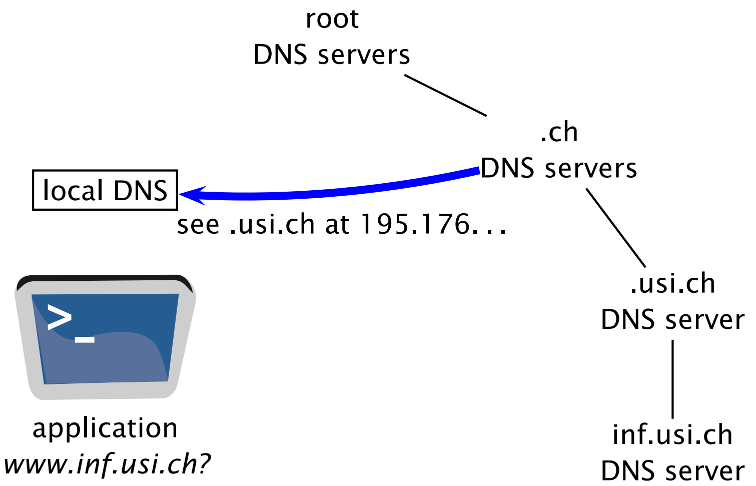
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



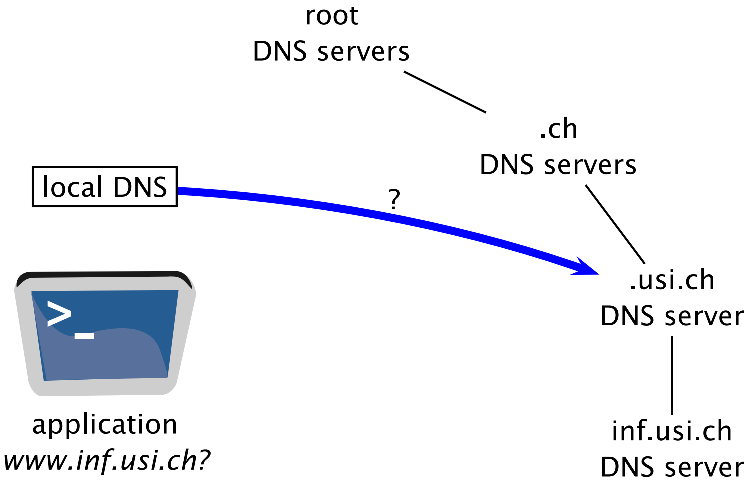
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



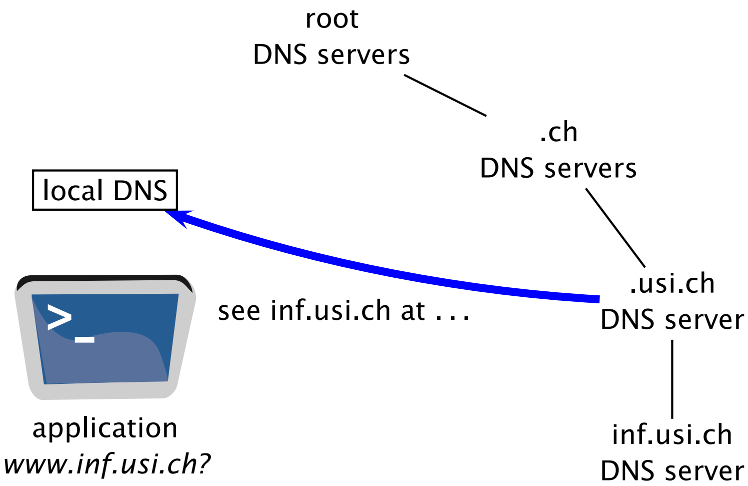
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



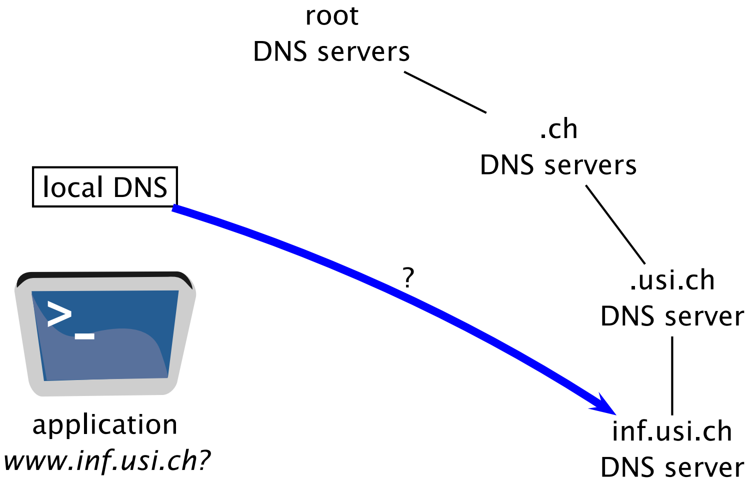
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



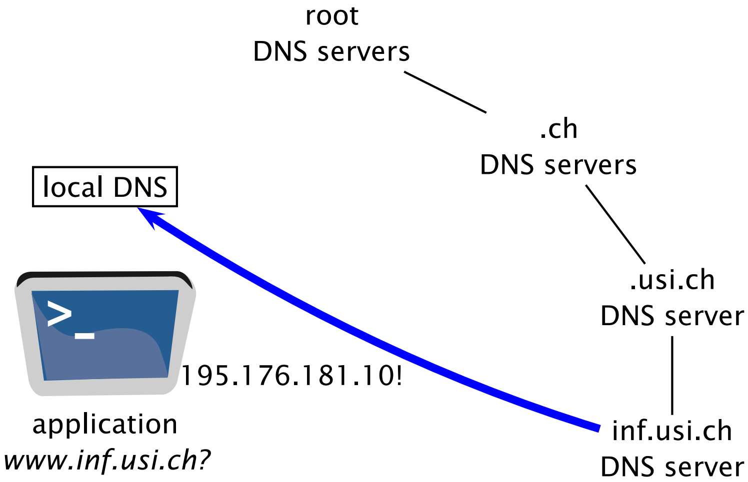
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



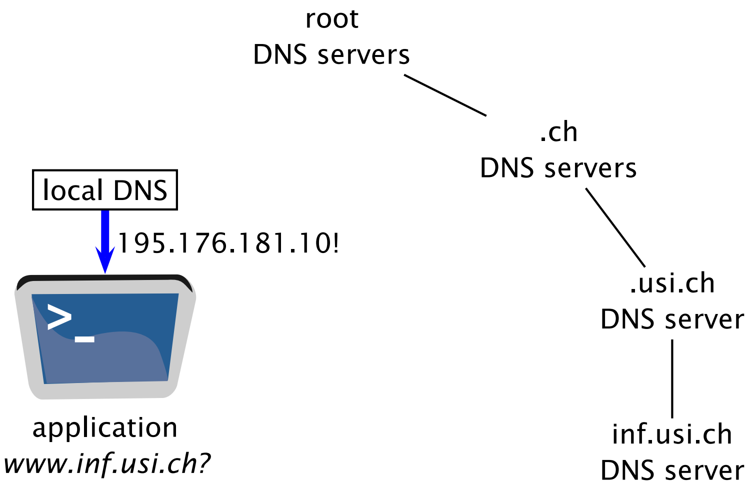
How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



How DNS Works

- Hierarchical architecture that mirrors the hierarchical structure of the namespace



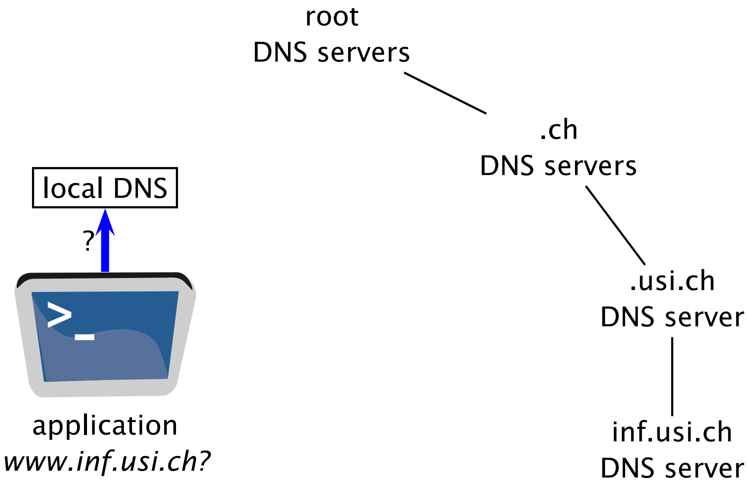
Recursive Queries

Recursive Queries

- A client/server can request a recursive query

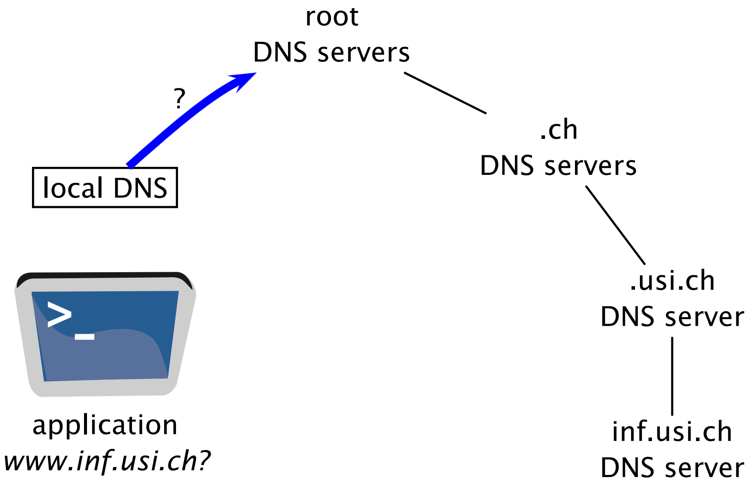
Recursive Queries

- A client/server can request a recursive query



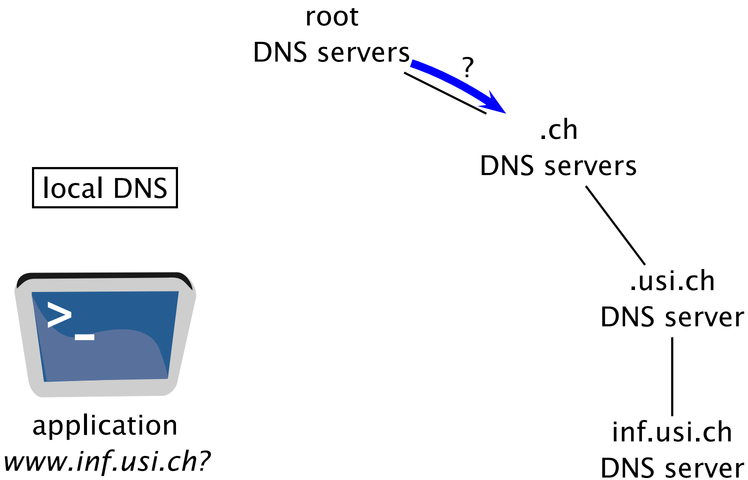
Recursive Queries

- A client/server can request a recursive query



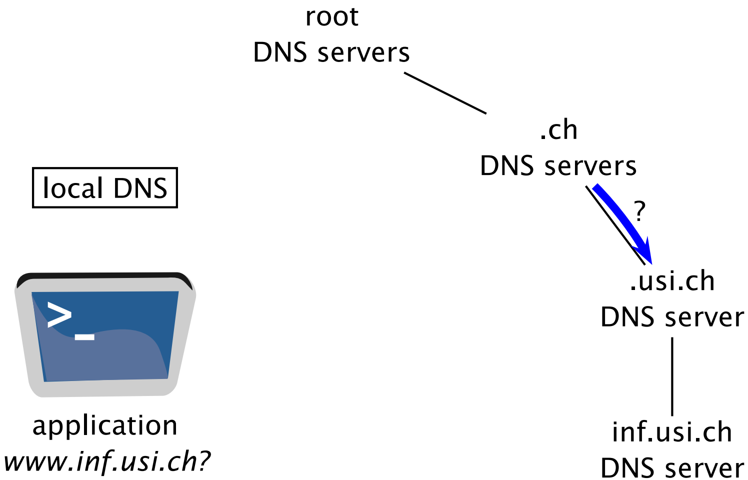
Recursive Queries

- A client/server can request a recursive query



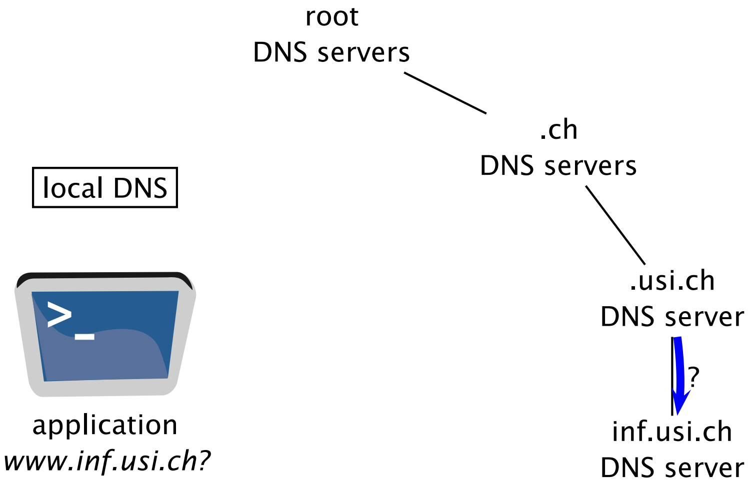
Recursive Queries

- A client/server can request a recursive query



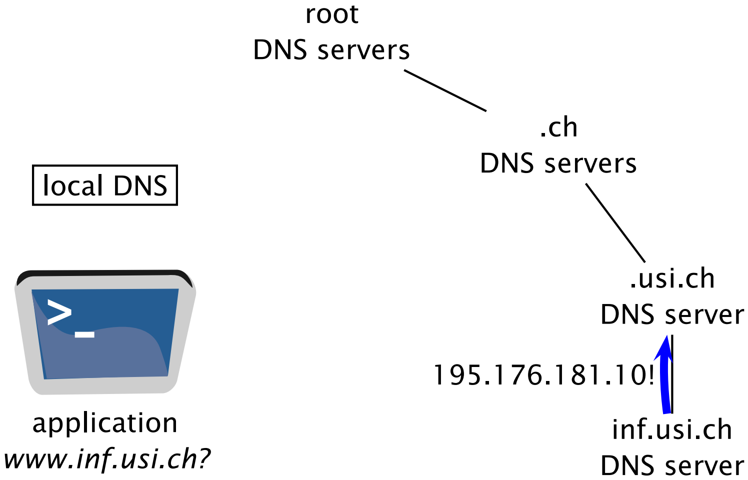
Recursive Queries

- A client/server can request a recursive query



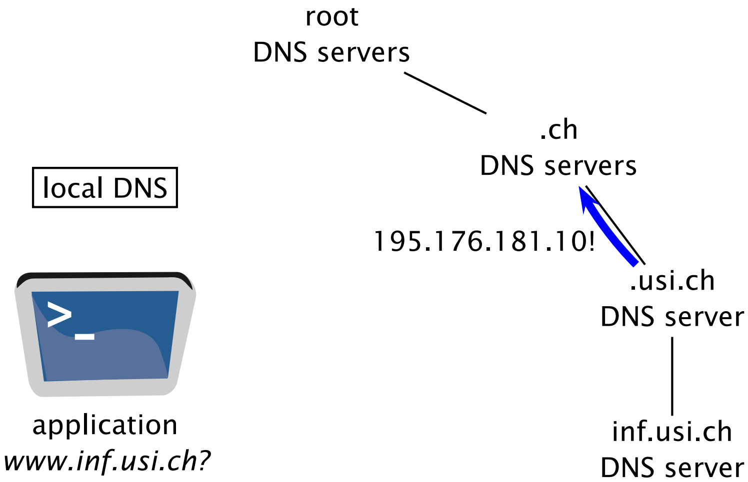
Recursive Queries

- A client/server can request a recursive query



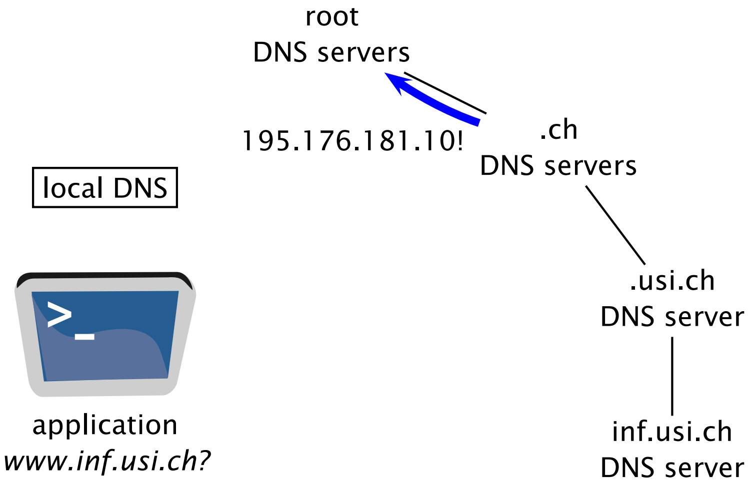
Recursive Queries

- A client/server can request a recursive query



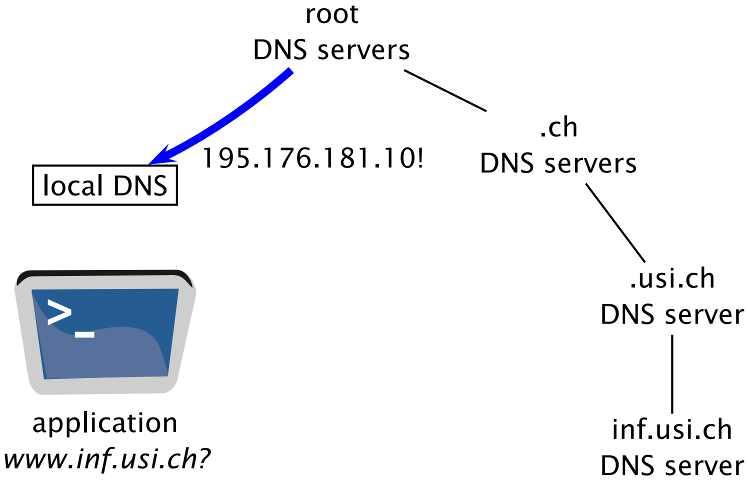
Recursive Queries

- A client/server can request a recursive query



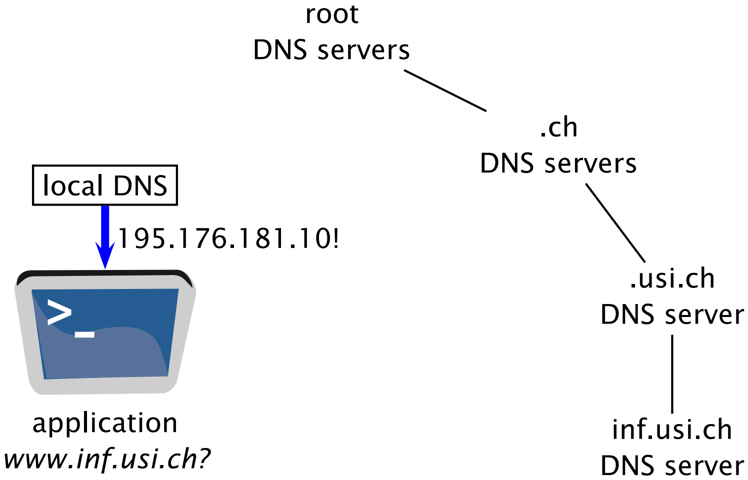
Recursive Queries

- A client/server can request a recursive query



Recursive Queries

- A client/server can request a recursive query



Observations on DNS

Observations on DNS

- A lot of messages just to figure out where to connect to!
  - DNS can indeed be a major bottleneck for some applications (typically, the Web)
  - it is also to a large extent a critical point of failure

Observations on DNS

- A lot of messages just to figure out where to connect to!
  - DNS can indeed be a major bottleneck for some applications (typically, the Web)
  - it is also to a large extent a critical point of failure
- It is a perfect demonstration of the “end-to-end principle”
  - it implements a (crucial) network functionality at the end-system level

Observations on DNS

- A lot of messages just to figure out where to connect to!
  - DNS can indeed be a major bottleneck for some applications (typically, the Web)
  - it is also to a large extent a critical point of failure
- It is a perfect demonstration of the "end-to-end principle"
  - it implements a (crucial) network functionality at the end-system level
- Any idea how to improve the performance and reliability of DNS?

DNS Caching

- Caching is clearly very important, as it can dramatically
  - improve the performance of DNS
  - reduce the load on the DNS infrastructure

DNS Caching

- Caching is clearly very important, as it can dramatically
  - improve the performance of DNS
  - reduce the load on the DNS infrastructure
- How does caching work in DNS?

DNS Caching

- Caching is clearly very important, as it can dramatically
  - improve the performance of DNS
  - reduce the load on the DNS infrastructure

- How does caching work in DNS?

- Same as always
  - a DNS server may cache a reply (i.e., the mapping) for a name *n*
  - if the server receives a subsequent request for *n*, it may respond directly with the cached address, even though the server is not the authoritative server for that domain

DNS Features

- DNS is essentially a “directory service” database
- The database contains *resource records (RRs)*

DNS Features

- DNS is essentially a “directory service” database

- The database contains *resource records (RRs)*

name	*value*	*type*	*ttl*
www.inf.usi.ch	195.176.181.10	A	...
research.inf.usi.ch	195.176.181.11	A	...
...	...	...	...

DNS Features

- DNS is essentially a "directory service" database
- The database contains *resource records (RRs)*

name	*value*	*type*	*ttl*
www.inf.usi.ch	195.176.181.10	A	...
research.inf.usi.ch	195.176.181.11	A	...
...	...	...	...

- *Name* and *value* have the intuitive meaning

DNS Features

- DNS is essentially a “directory service” database
- The database contains *resource records (RRs)*

name	*value*	*type*	*ttl*
www.inf.usi.ch	195.176.181.10	A	...
research.inf.usi.ch	195.176.181.11	A	...
...	...	...	...

- *Name* and *value* have the intuitive meaning
- What about *type*?

DNS Query Types

DNS Query Types

A this is the main mapping *host_name* → *address*, so *name* is a host name and *value* is its (IP) *address*

DNS Query Types

A this is the main mapping *host_name* → *address*, so *name* is a host name and *value* is its (IP) *address*

NS this is a query for a name server, so *name* is a domain name and *value* is the *authoritative name server* for that domain. For example,

name	*value*	*type*	*ttl*
usi.ch	one.ti-edu.ch	NS	...

DNS Query Types

A this is the main mapping *host_name* → *address*, so *name* is a host name and *value* is its (IP) *address*

NS this is a query for a name server, so *name* is a domain name and *value* is the *authoritative name server* for that domain. For example,

name	*value*	*type*	*ttl*
usi.ch	one.ti-edu.ch	NS	...

CNAME this is a query for a *canonical name*. The canonical name is the "primary" name of a host. A host may have one or more mnemonic *aliases*. For example,

name	*value*	*type*	*ttl*
www.google.com	www.l.google.com	CNAME	...

DNS Query Types (2)

DNS Query Types (2)

MX this is a query for the *mail exchange* server for a given domain, so *name* is a host or domain name and *value* is the name of the mail server that handles (incoming) mail for that host or domain. For example,

name	*value*	*type*	*ttl*
lu.usi.ch	spamfilter.usilu.net	MX	...

DNS Query Types (2)

MX this is a query for the *mail exchange* server for a given domain, so *name* is a host or domain name and *value* is the name of the mail server that handles (incoming) mail for that host or domain. For example,

name	*value*	*type*	*ttl*
lu.usi.ch	spamfilter.usilu.net	MX	...

... several other types

DNS Protocol

DNS Protocol

- DNS is a connectionless protocol
- Runs on top of UDP (port 53)

DNS Protocol

- DNS is a connectionless protocol
- Runs on top of UDP (port 53)
- DNS has *query* and *reply* messages
  - since DNS is connectionless, queries and replies are linked by an identifier

DNS Protocol

- DNS is a connectionless protocol
- Runs on top of UDP (port 53)
- DNS has *query* and *reply* messages
  - since DNS is connectionless, queries and replies are linked by an identifier
- Both queries and replies have the same format
  - *a DNS message can carry queries and answers*

DNS Message Format

DNS Message Format