

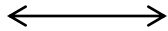
# DNS

- Domain Name System: DNS
- Objective: map names to IP addresses (i.e., high level names to low level names)
- Original namespace was flat, didn't scale..
- Hierarchical naming permits decentralization by delegating authority for parts of the namespace.
  - Scalability and performance advantages
  - Administrative benefits
- DNS is an example of a large scale client-server application.

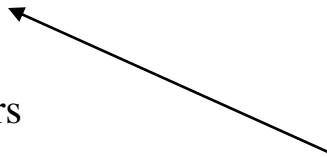
# DNS

- A domain name consists of a labels separated by periods

- `www.cs.clemson.edu`



Label: up to 63 characters

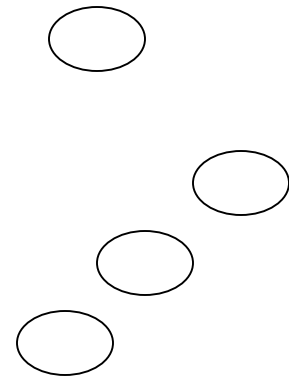


A domain name can have up to 255 labels

*RFC 2181: Clarifications to the DNS Specification: specifies a maximum of 63 octets per label and 255 octets per name.*

- A domain name has an inherent hierarchy

edu	top level domain	: edu
clemson	org. level	: clemson.edu
cs	department level	: cs.clemson.edu
www	host level	:



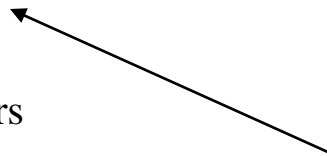
# DNS

- A domain name consists of a labels separated by periods

- `www.cs.clemson.edu`



Label: up to 63 characters



A domain name can have up to 255 labels

*RFC 2181: Clarifications to the DNS Specification: specifies a maximum of 63 octets per label and 255 octets per name.*

- A domain name has an inherent hierarchy

edu	top level domain	: edu
clemson	org. level	: clemson.edu
cs	department level	: cs.clemson.edu
www	host level	:

○ Unnamed root

○ .edu TLD server



- A domain name is fully qualified if it ends in the “.” to indicate root.

- Most platforms assume a name is a FQDN if it contains 2 or more labels

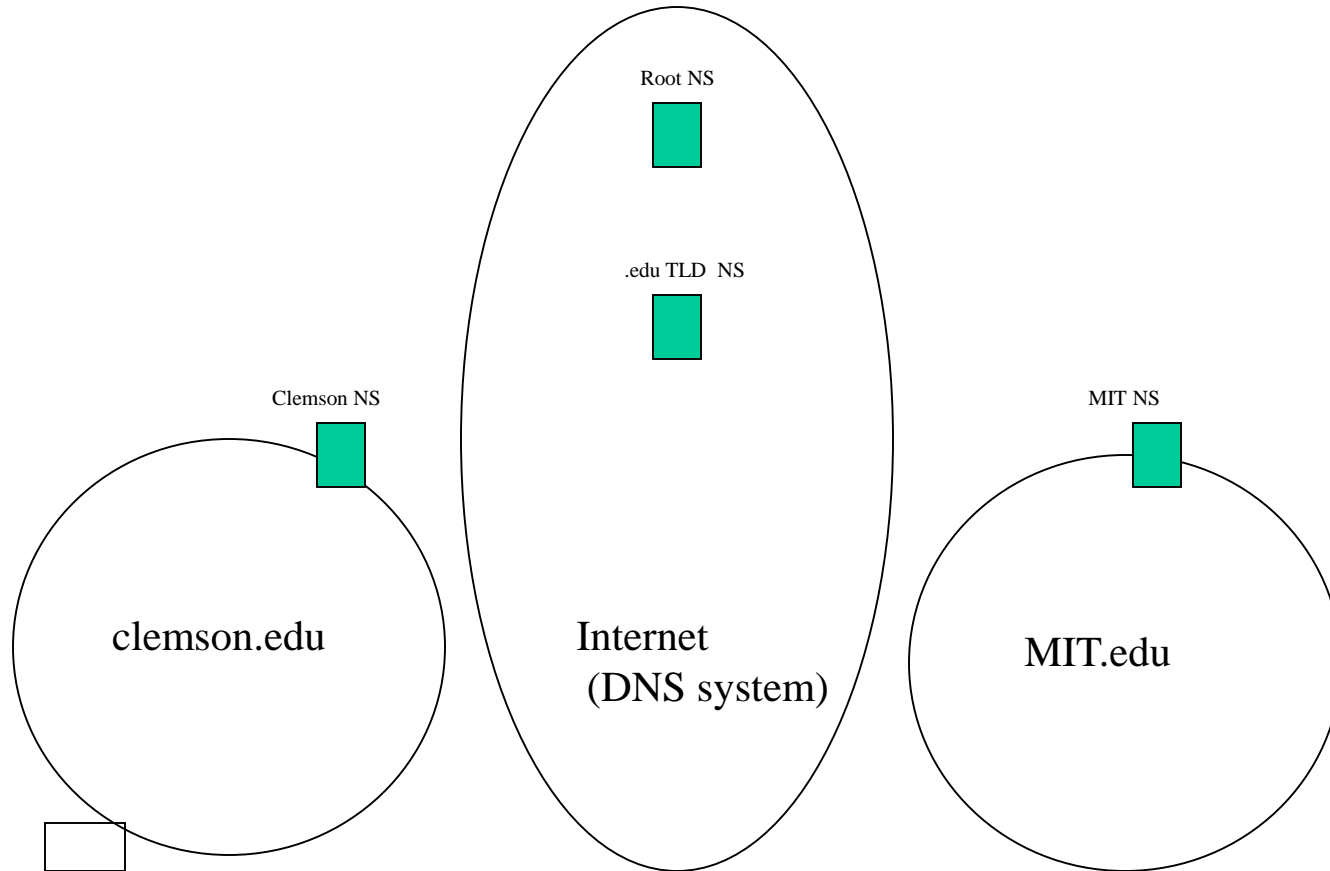
# DNS Key Concepts

- Client code called the ‘resolver’
  - Requires the IP address of at least one name server
- Nameservers;
  - Require at least the IP of the root NS
  - Maintains a cache
- Two types of DNS replies:
  - Authoritative: The answer comes from a NS that ‘owns’ the domain name – the name is in the NS’s zone of authority
  - Non-authoritative: The answer comes from a NS who had the binding in its cache
- Two types of DNS requests:
  - Iterative or recursive: The difference is in who ‘does the work’ to completely resolve the domain name

# DNS

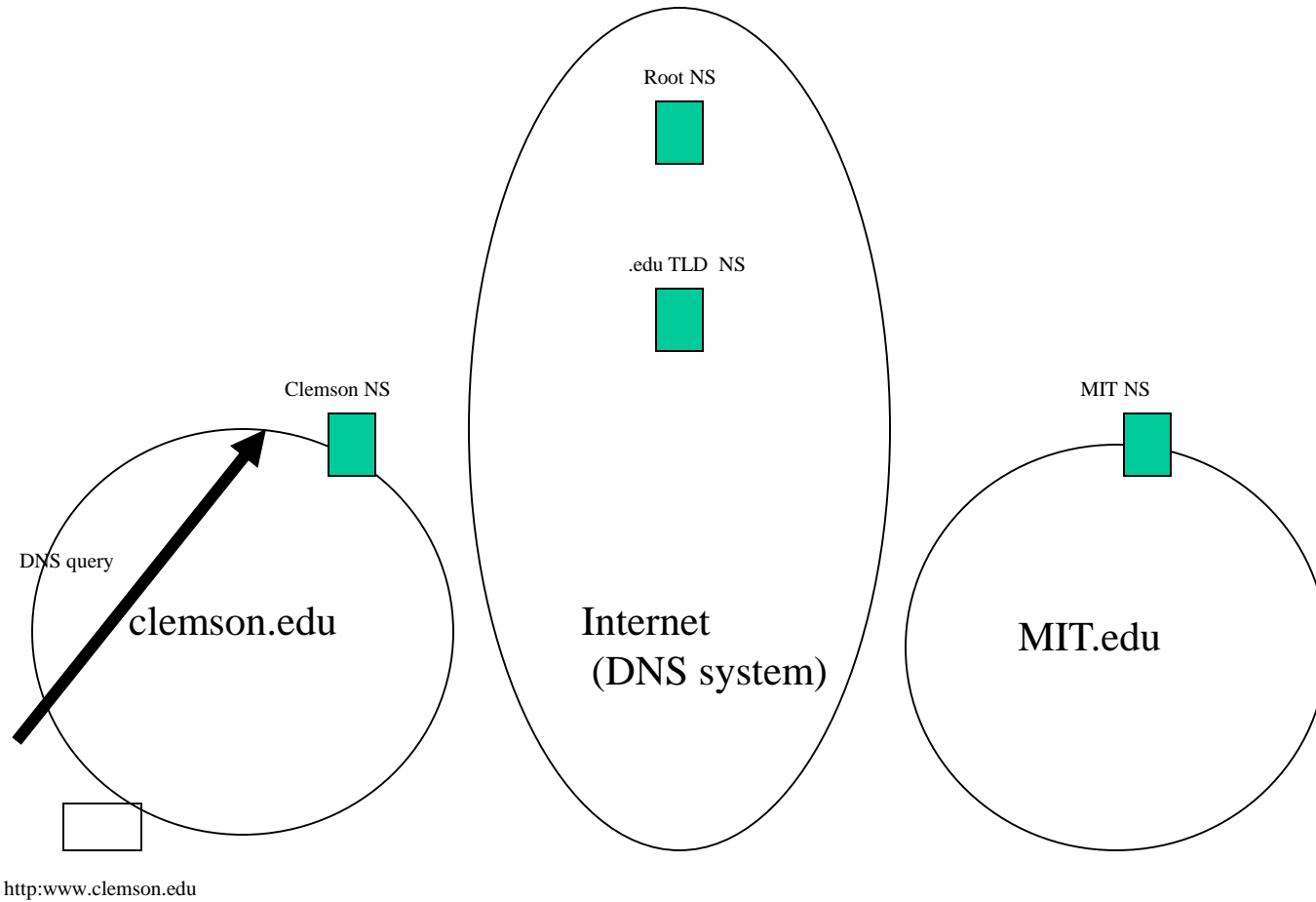
- Internet Domain Names System Managing organizations :  
IANA, ICANN, INTERNIC

Example: web browsing from a host on  
clemson.edu, what happens when you access  
www.mit.edu ?

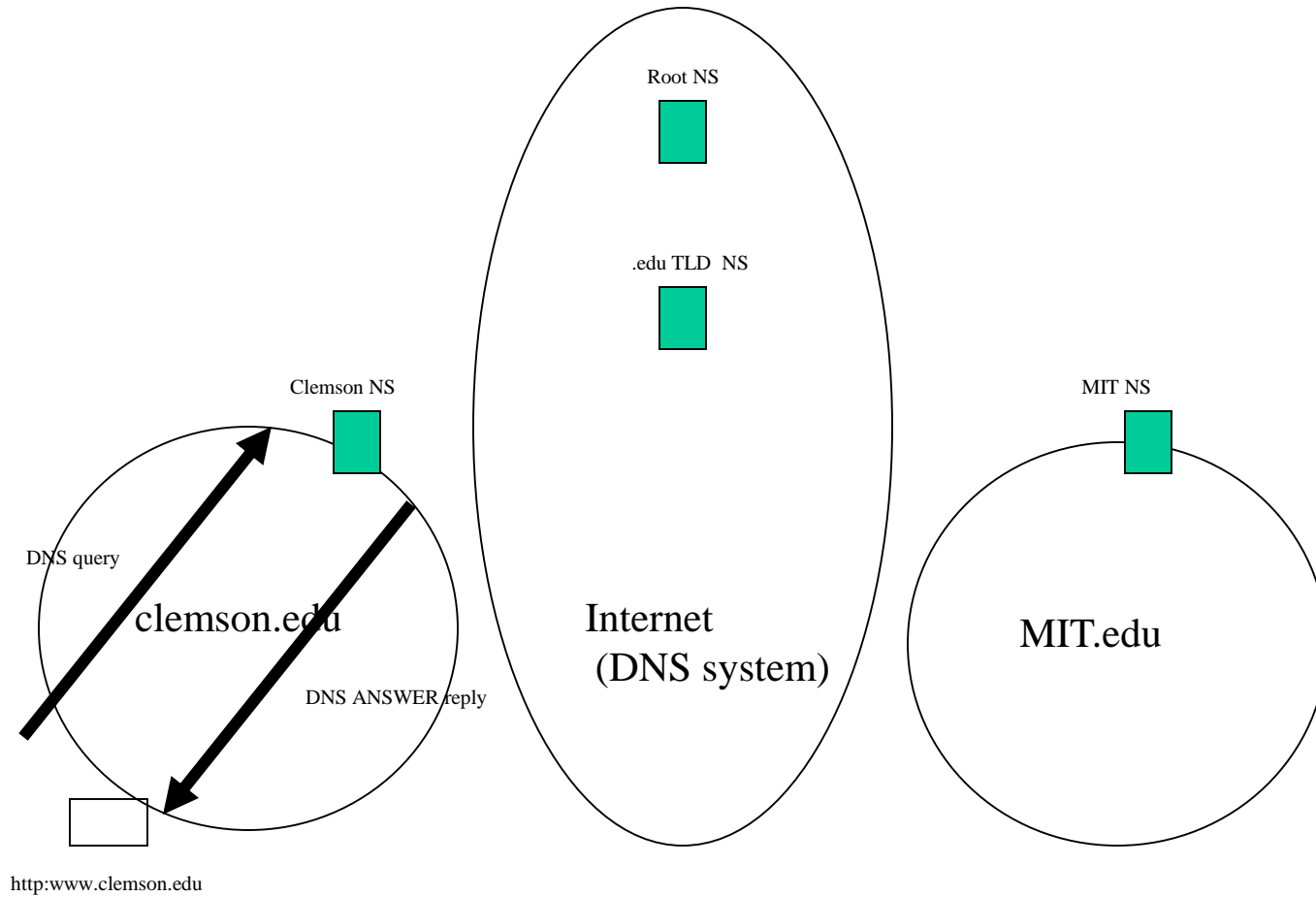


<http://www.clemson.edu>

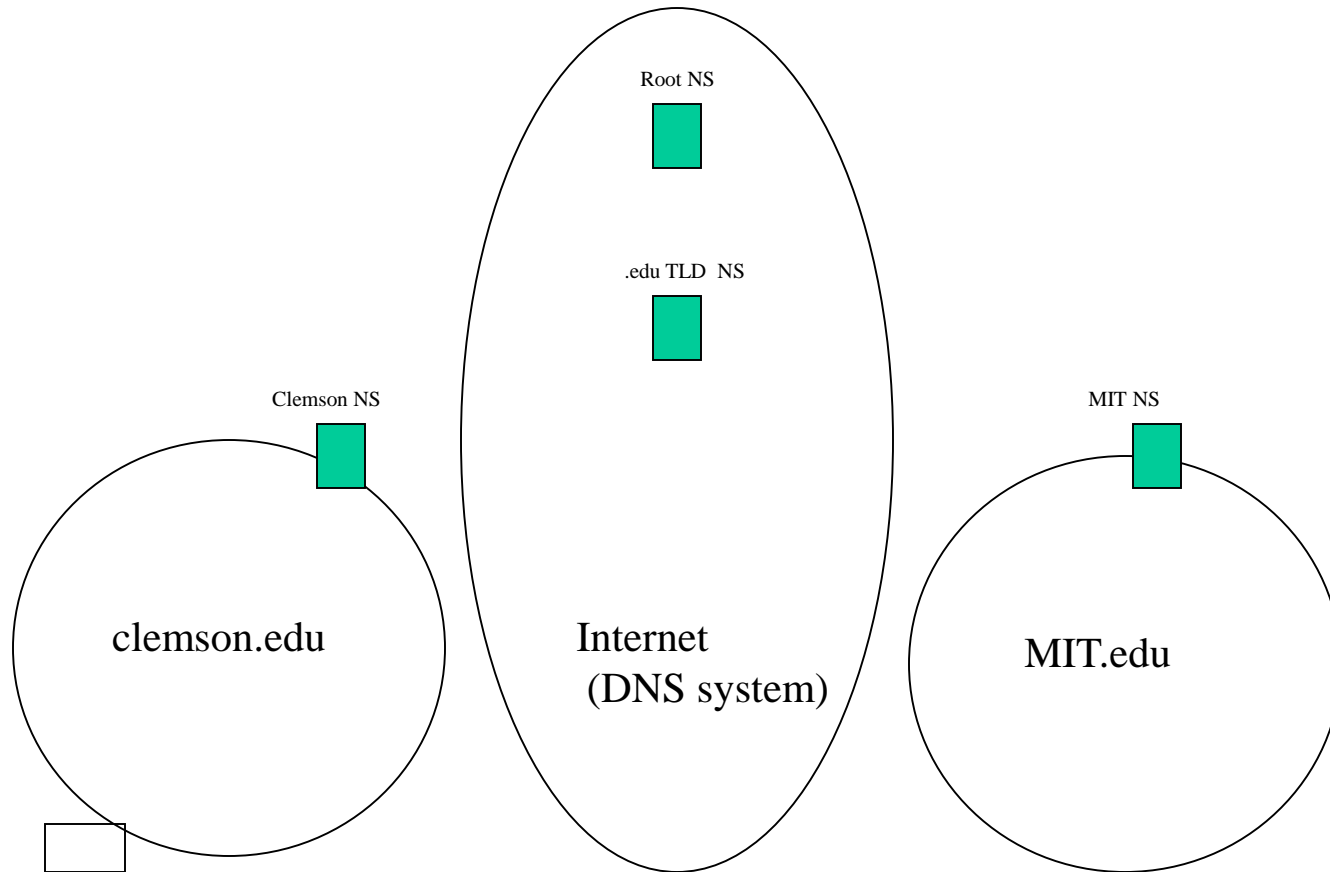
# In the local NS' s cache



# In the local NS' s cache

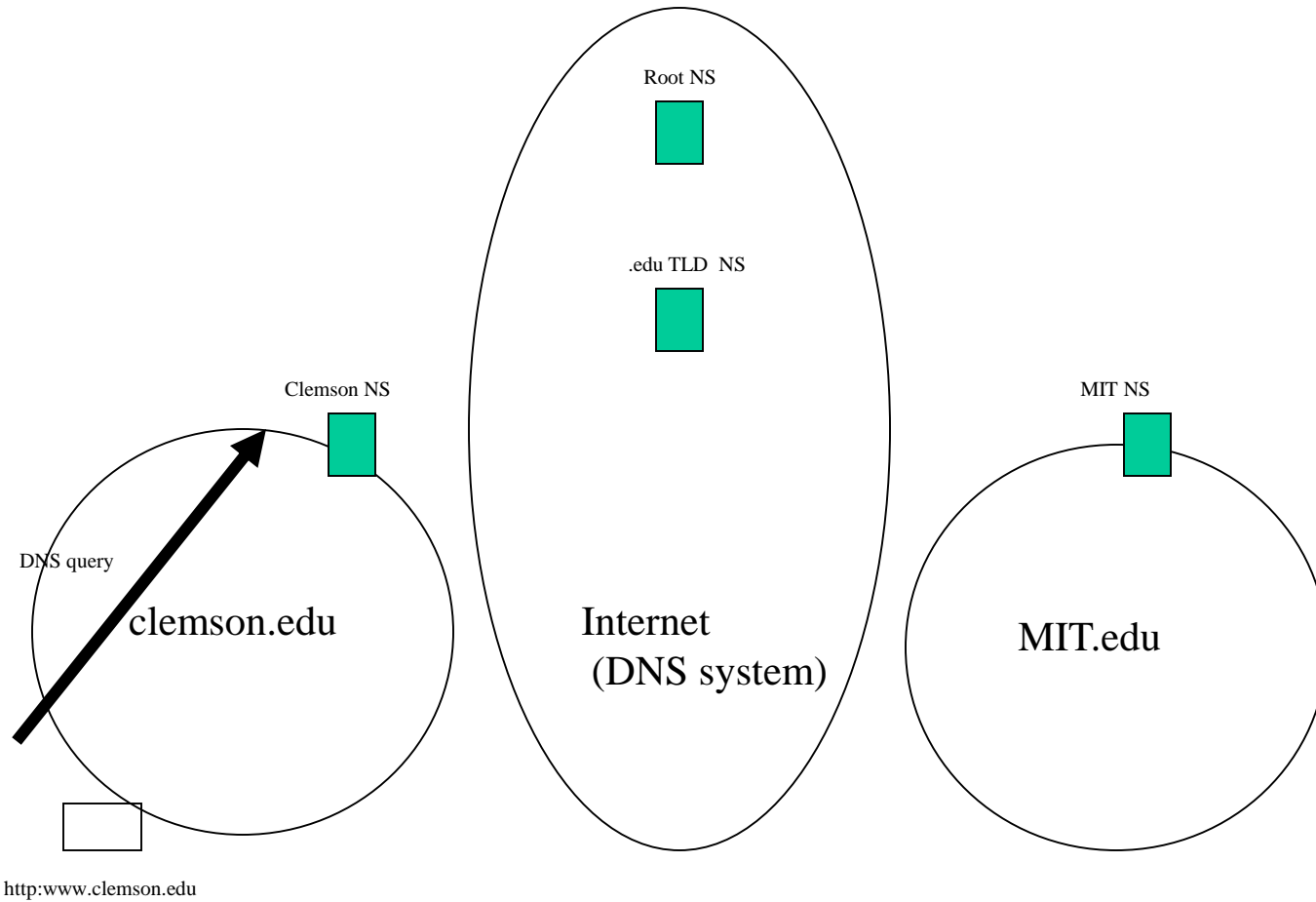


# Recursive request sent by client, not in cache

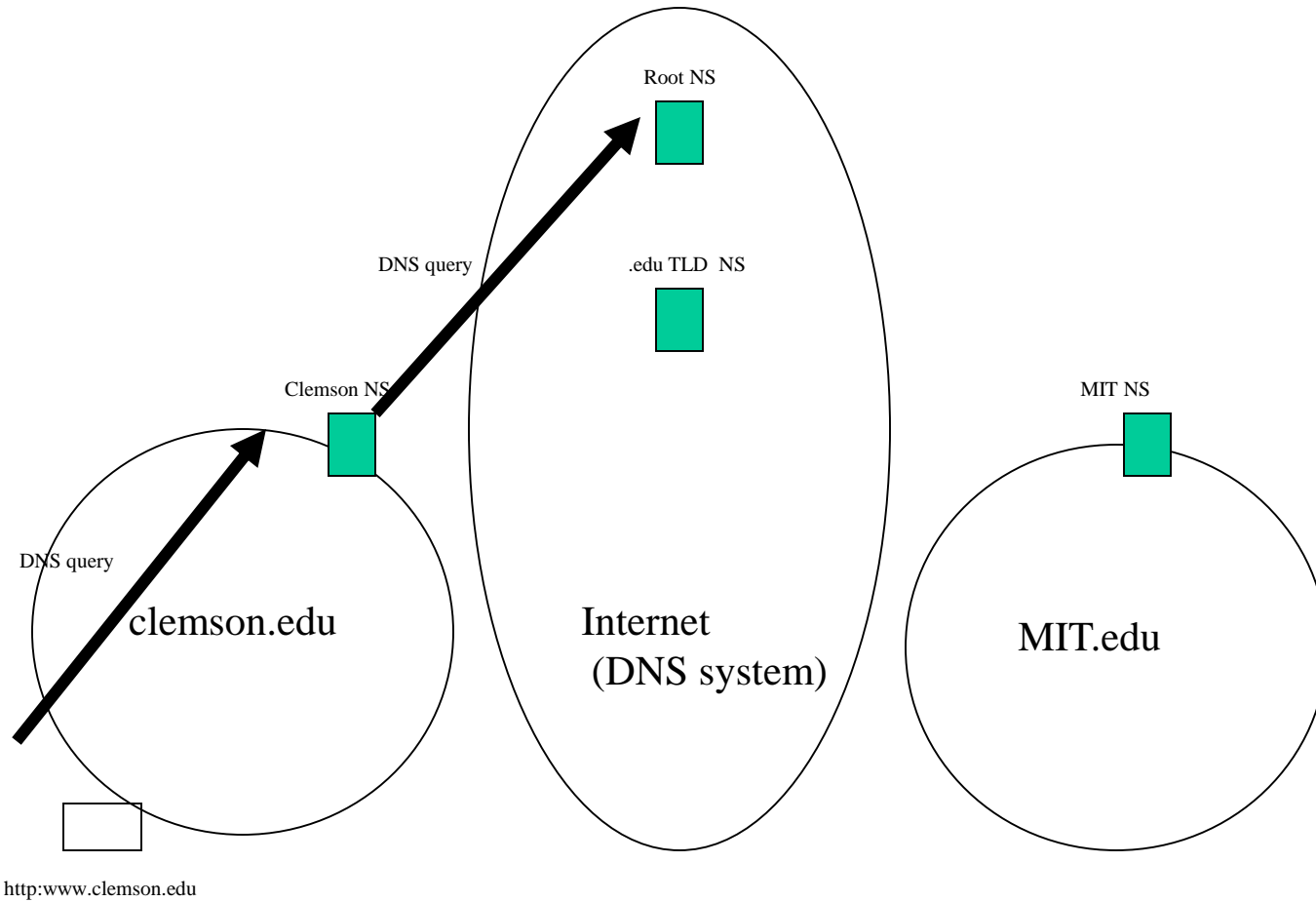


<http://www.clemson.edu>

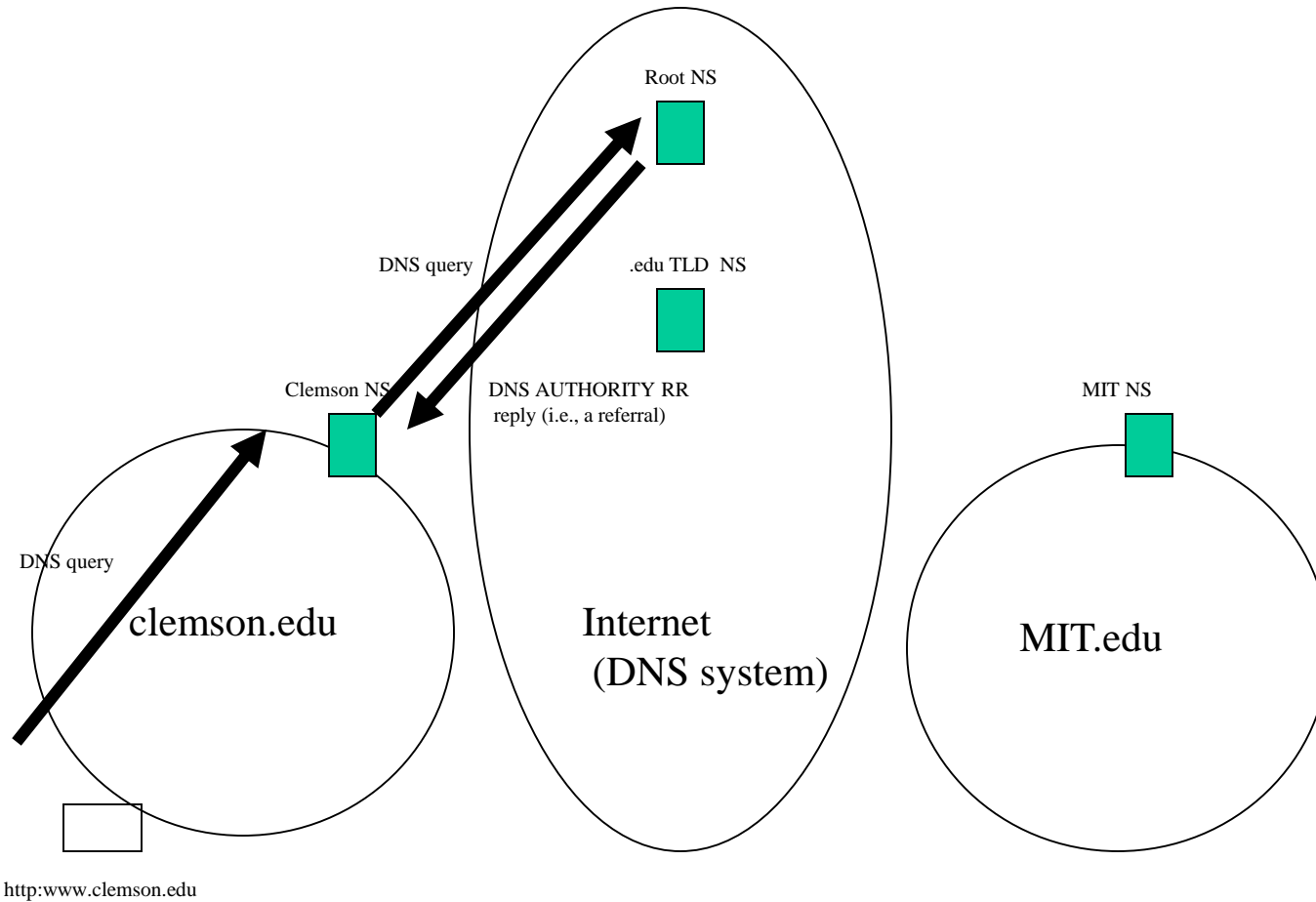
# Recursive request sent by client not in cache



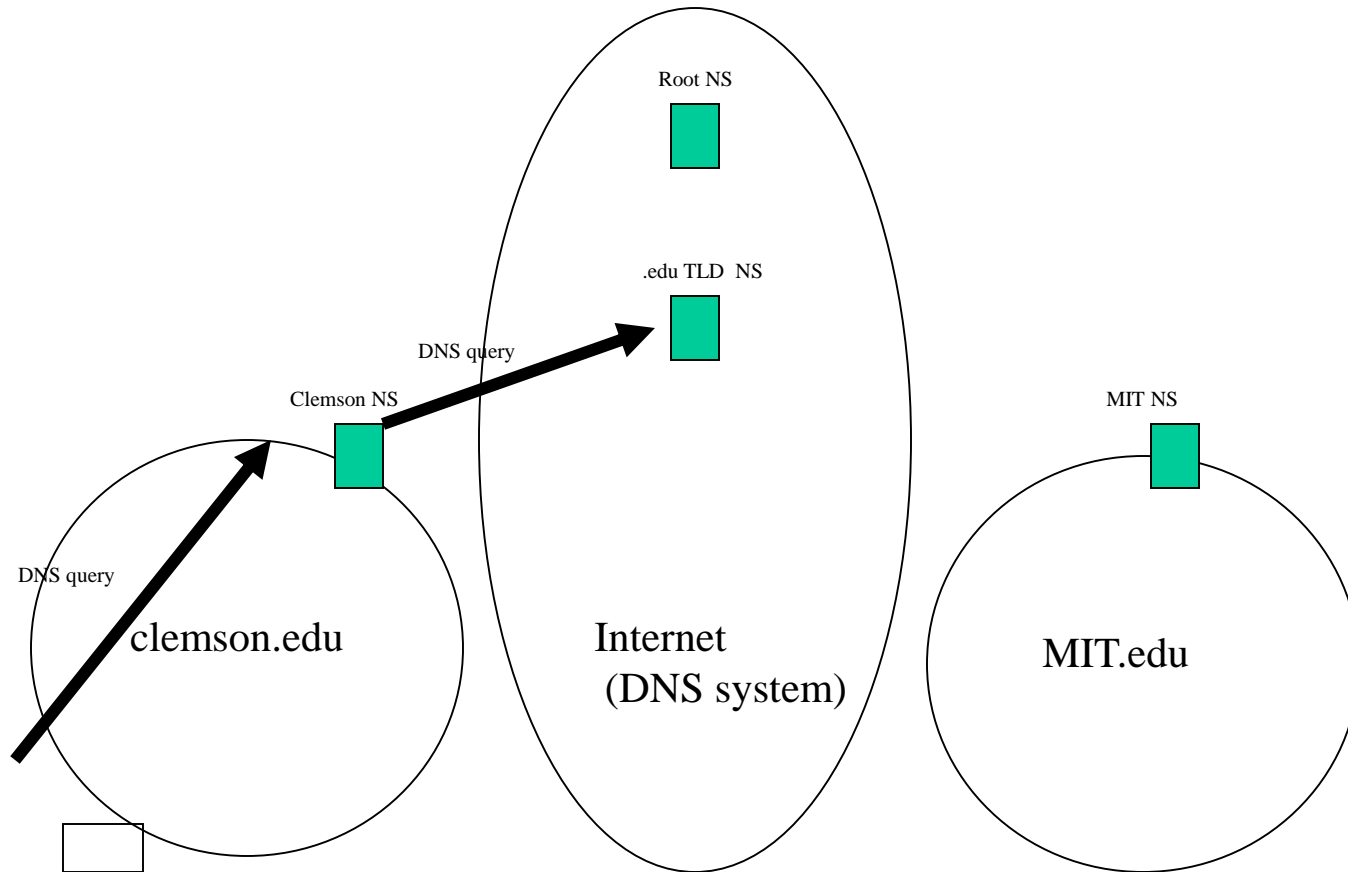
# Recursive request sent by client not in cache



# Recursive request sent by client

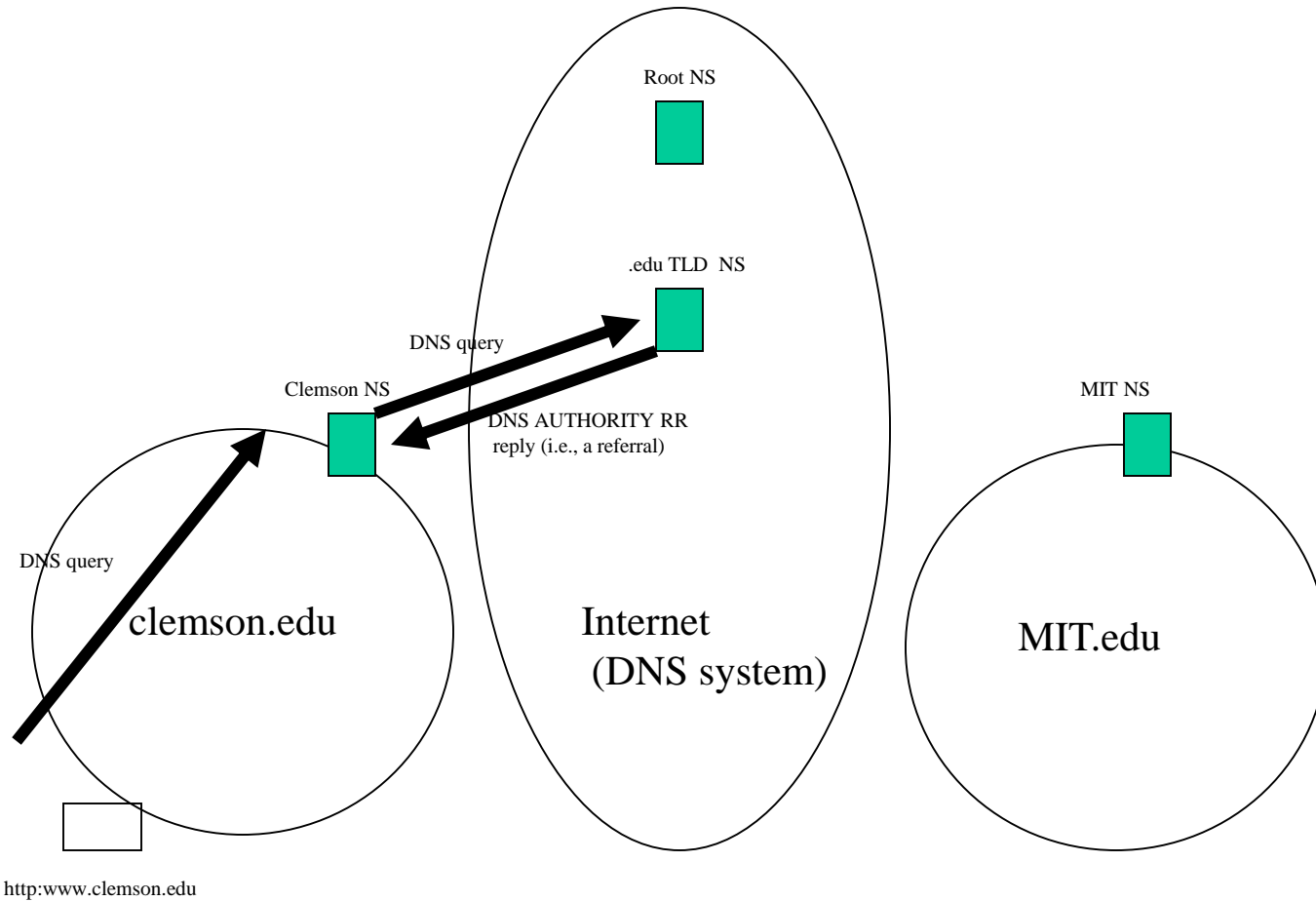


# Recursive request sent by client

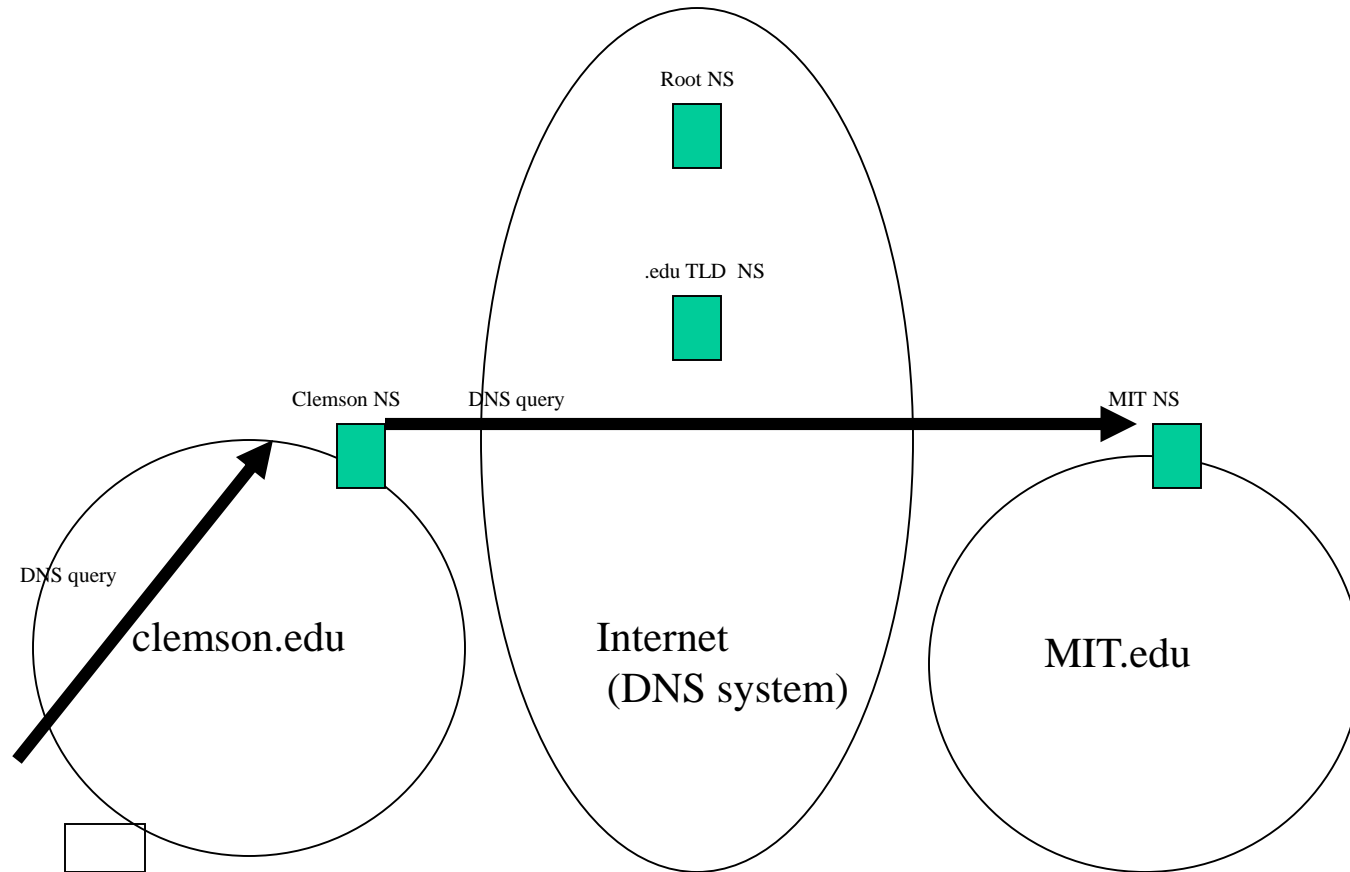


<http://www.clemson.edu>

# Recursive request sent by client

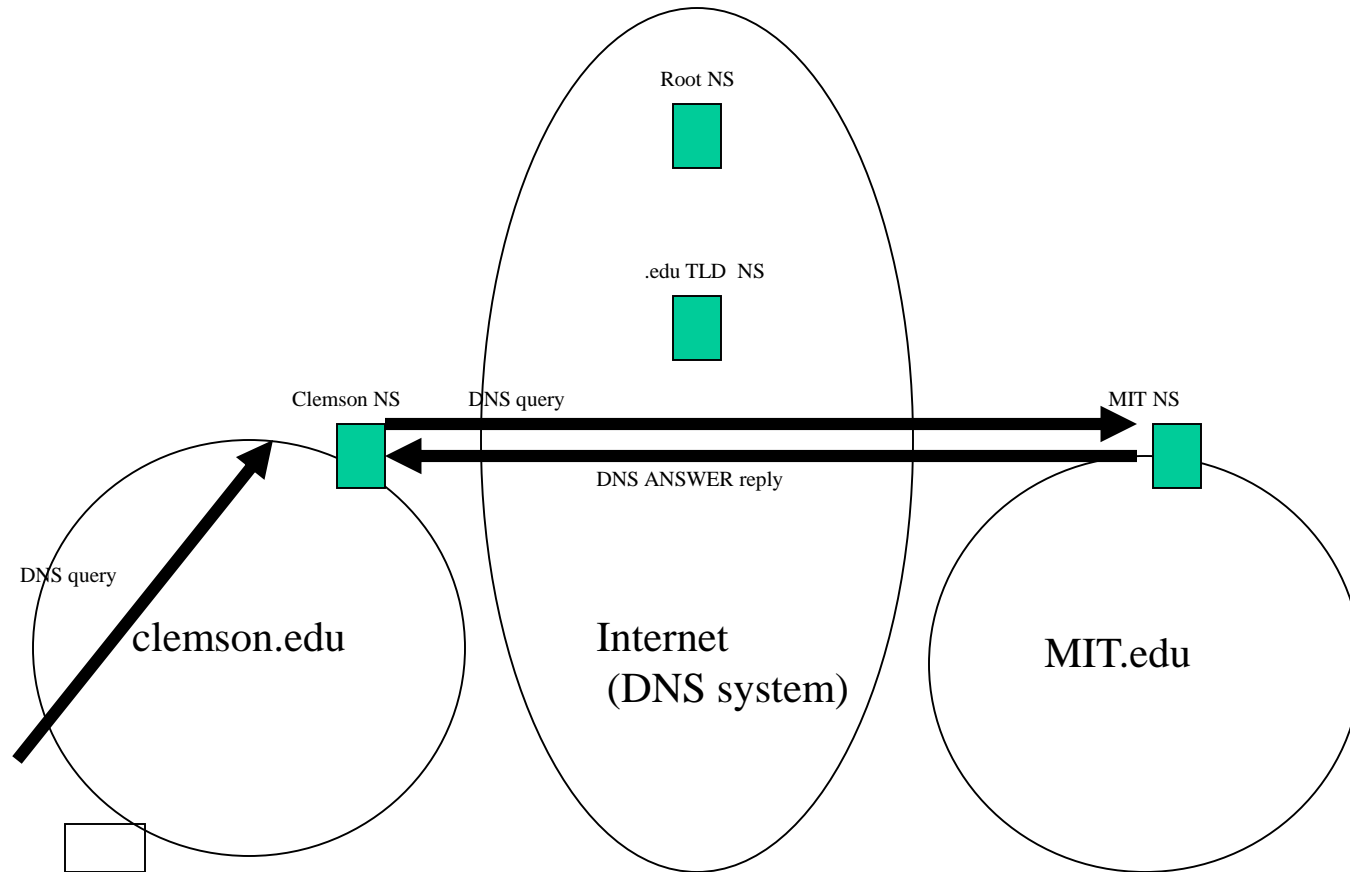


# Recursive request sent by client



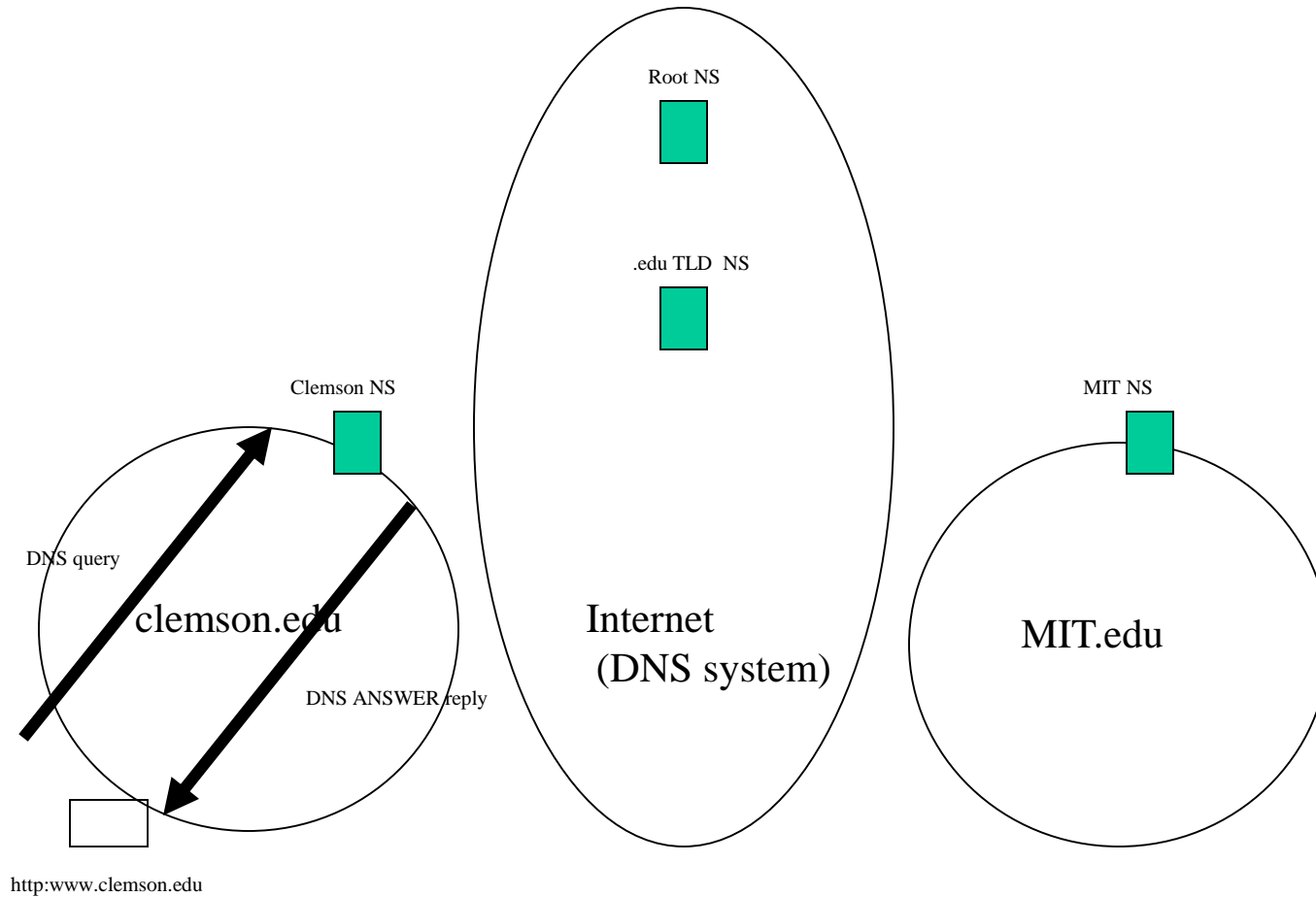
<http://www.clemson.edu>

# Recursive request sent by client



<http://www.clemson.edu>

Example: web browsing from a host on clemson.edu, what happens when you access www.mit.edu ?



# DNS

- Host asks local server to resolve `www.mit.edu` by sending a DNS query message.
- 3 possibilities
  - local NS has it cached and returns a non-authoritative resource record (RR),
  - if the local NS did not have the name cached:
    - And if the client asked for an iterative resolution the local NS returns a Name Server (NS) resource record.
    - And if the client asked for a recursive solution the local NS returns the resolved name.
      - Local NS sends a query to a root server which might return a ‘referral’ pointing to the `.edu` server.
      - The local NS sends a query to the `.edu` server which might return a referral pointing to the MIT NS.
      - Finally, the local NS queries the `.mit.edu` server which returns an “A” record (the 32 bit IP address)

# DNS

- Two aspects of DNS:
  - Local host resolver
  - DNS query/response protocol

Local host resolver:

- `GetHostByName()`, `GetHostByAddr()`
- Handles abbreviations
- On Unix, `/etc/resolv.conf` tells the resolver the name server and the default domain

# DNS: Sockets Support

struct hostent \*thehost;

char \*h\_name

This is the “official” name of the host.

char \*\*h\_aliases

These are alternative names for the host, represented as a null-terminated vector of strings.

int h\_addrtype

This is the host address type; in practice, its value is always either AF\_INET or AF\_INET6, with the latter being used for IPv6 hosts. In principle other kinds of addresses could be represented in the database as well as Internet addresses; if this were done, you might find a value in this field other than AF\_INET or AF\_INET6. See Socket Addresses.

int h\_length

This is the length, in bytes, of each address.

char \*\*h\_addr\_list

This is the vector of addresses for the host. (Recall that the host might be connected to multiple networks and have different addresses on each one.) The vector is terminated by a null pointer.

char \*h\_addr

This is a synonym for h\_addr\_list[0]; in other words, it is the first host address.

# DNS: Sockets Support

```
/* Construct the server address structure */
memset(&echoServAddr, 0, sizeof(echoServAddr));
echoServAddr.sin_family = AF_INET;
echoServAddr.sin_addr.s_addr = inet_addr(servIP); //converts from dotted decimal to 32 bit IP
                                                    //returns -1 if NOT dotted decimal

/* If user gave a name ip, we need to resolve it */
if (echoServAddr.sin_addr.s_addr == -1)
{
    thehost = gethostbyname(servIP);
    echoServAddr.sin_addr.s_addr = *((unsigned long *) thehost->h_addr_list[0]);
}

echoServAddr.sin_port = htons(echoServPort);
```

## : Internet Addressing, Naming

- DNS query/reply protocol
  - show message format
  - Query types: most common Type A and Type PTR
  - Responses in the answer, authority, and additional information sections are called Resource Records

### Protocol aspects:

- typically UDP, sometimes TCP
- If using UDP, DNS must do its own recovery

# DNS

- Domain name registration services ?

- Domain registrations** are leased to the registrant in yearly increments.

- A domain name **registry** is the operator for a particular Top-Level Domain (TLD). For example, VeriSign is the registry operator of '.com' and '.net' domains, PIR is the registry operator of '.org', and NeuStar is the registry operator of '.us' and '.biz'.

- A **registrar**, also known as a Domain Name Registration Service, is a company that inserts your domain name into a directory of all the domain names and their corresponding computers on the Internet. A domain name registrar is a go-between for the registries and the end-user. See, for example: [www.mydomain.com](http://www.mydomain.com)

- Much like a retailer sells good from a wholesaler, a registrar sells domain registrations directly to the registrants, maintains information for the registrations such as nameserver delegation and WHOIS information (see below), makes changes to the registry on behalf of the registrants, and provides customer service and support. While a registry is usually governed by a single entity, many registrars exist to sell registrations to customers on their behalf, creating a competitive market.

- Domain hosting ??

- Different from domain name registration. A company ([www.drmartinrocks.com](http://www.drmartinrocks.com)) might chose NOT to operate its own name server. It can pay a monthly fee for a company like [www.mydomain.com](http://www.mydomain.com) to do this.

# DNS

- Dynamic DNS:

- is a system which allows the domain name data held in a name server to be updated in real time. The most common use for this is in allowing an Internet domain name to be assigned to a computer with a varying (dynamic) IP address. This makes it possible for other sites on the Internet to establish connections to the machine without needing to track the IP address themselves. A common use is for running server software on a computer that has a dynamic IP address, as is the case with many consumer Internet service providers.

- To implement dynamic DNS it is necessary to set the maximum caching time of the domain to an unusually short period (typically a few minutes). This prevents other nodes on the Internet from retaining the old address in their DNS cache, so that they will typically contact the name server of the domain for each new connection.

- See <http://www.dyndns.com/services/dns/dyndns/>