COT 4600 Operating Systems Fall 2009

Dan C. Marinescu

Office: HEC 439 B

Office hours: Tu-Th 3:00-4:00 PM

Lecture 13

- Last time:
  - Review; discussion of midterm type problems
- Today:
  - Peer-to-peer systems
  - Remote Procedure Call (RPC)
  - Domain Name Service (DNS)
- Next Time:
  - Domain Name Service (DNS)
  - Network File System (NFS)

Peer-to-peer systems

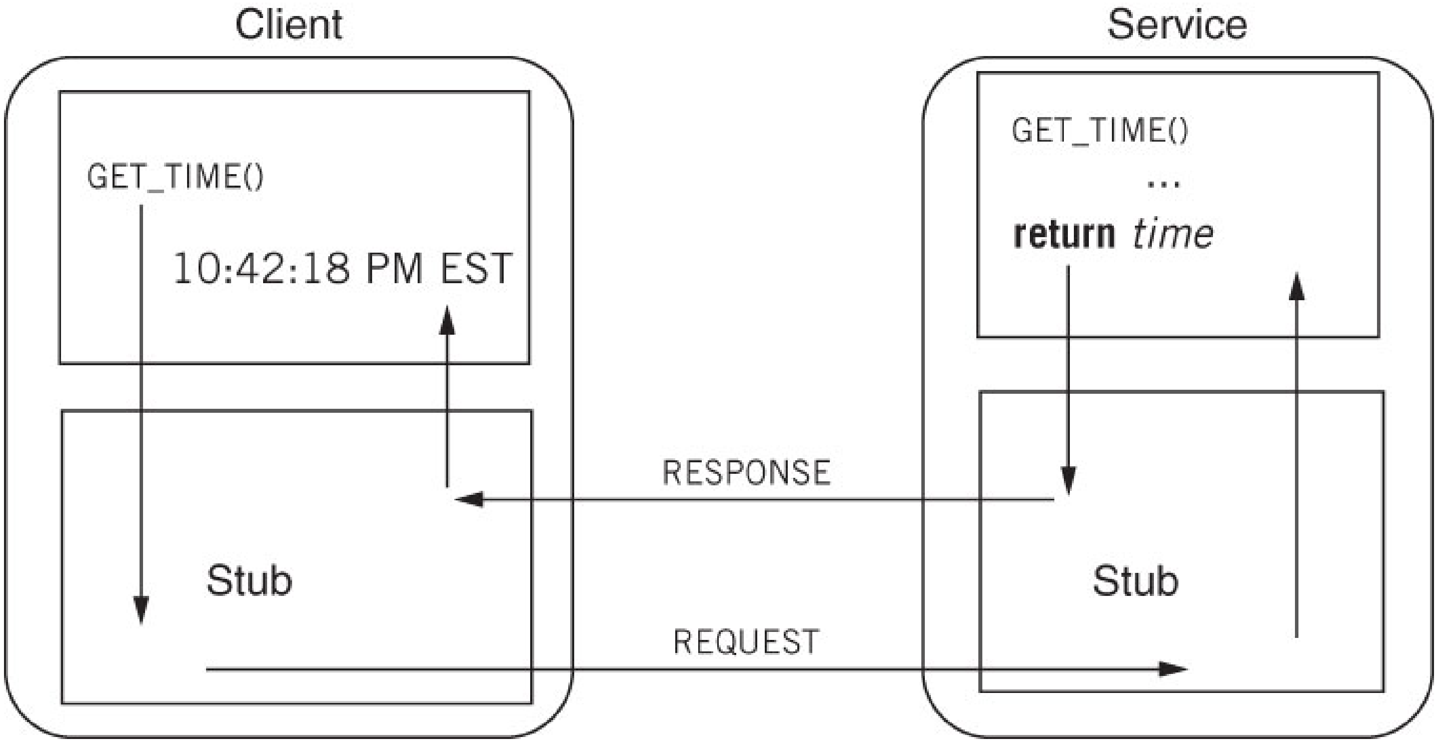
- Decentralized architecture without a trusted intermediary.
- Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model where servers supply, and clients consume.
- Peer-to-peer systems often implement an Application Layer overlay network on top of the native or physical network topology. Such overlays are used for indexing and peer discovery.
- Content is typically exchanged directly over the underlying IP network.
- Anonymous peer-to-peer systems implement extra routing layers to obscure the identity of the source or destination of queries.
- In *structured* peer-to-peer networks, connections in the overlay are fixed. They typically use distributed hash table-based (DHT) indexing, such as in the Chord system developed at MIT
- *Unstructured peer-to-peer* networks do not provide any algorithm for organization or optimization of network connections.
- Advantages→
  - use of spare resources at many sites
  - difficult to censor content
- Disadvantage
  - Finding information in a large peer-to-peer network is hard.

Remote procedure call (RPC)

- Support inter-process communication of remotely located processes and allows implementation of client-server systems (RFC 1831)
- Preserve the semantics of a local procedure call.
- To use an RPC a process may use a special service: PORTMAP or RPCBIND available at port 111. A new RPC service uses the *portmapper* to register. The portmapper also allows a service lookup.
- If the process knows the port number of the RPC it may call directly.
- RPC/TCP and also RPC/UDP
- Messages
  - must be well-structured; contain the identification of the specific RPC
  - are addressed to an RPC demon listening at an RPC port.
- A machine independent representation of data ➔ external data representation standard (XDR).

Stub

- Unburdens a user from implementation details of the RPC; it hides:
  - the marshalling of the arguments
  - the communication details
- The client calls the client stub which:
  1. marshals the arguments of the call into messages
  2. sends the message
  3. waits for the responds
  4. when the response arrives it un-marshals the results
  5. returns to the client


Client
GET_TIME()
10:42:18 PM EST
Stub
Service
GET_TIME()
...
return time
Stub
RESPONSE
REQUEST

RPCs differ from ordinary procedure calls

- RPCs
  - reduce the so called fate sharing between caller and callee
  - have a different semantics (see next slide)
  - take longer
  - global variables and RPC do not mix well

RPC semantics

- At least once → the client stub resends a message up to a given number of times until it receives a message from the server; is no guarantee of a response
  - the server may end up executing the a request more than once
  - suitable for side-effect free operations
- At most once → a message is acted upon at most once.
  - If the timeout set for receiving the response expires then an error code is delivered to the client.
  - The server must keep a history of the time-stamps of all messages. Messages may arrive out of order…..
  - suitable for operations which have side effects
- Exactly once → implement the at most once and request an acknowledgment from the server.

Intermediaries

- What if the sender and the receiver of a message are not active at the same time?
- Intermediaries support buffered communication and allow more flexibility→ the intermediary may decide how to sort messages
- The sender and the receiver may:
  - Push a message
  - Pull a message
- Example: the mail service:
  - The sender pushes a message into his/her outbox
  - The outbox pushes it to the inbox of the recipient
  - The recipient pulls it whenever s(he) wants
- The publish/subscribe paradigm → the sender notifies an event service when it produced a message. Recipients subscribe to the events and when the events occur the messages are delivered

Strategies for name resolution

1. Distribute to all parties a copy of the directory mapping names to physical /logical addresses. The strategy does not scale well:
    1. when the population is very large, e.g., the directory size is very large and the network traffic to distribute it would be horrendous
    2. the number of updates is proportional to the population and would add to the traffic
2. Central directory → easy to update but it does not scale well, “hot spot” contention.
3. Distributed directory → more sophisticated to implement but used successfully for DNS

IP addresses

- IP address serves two functions:
    - host identification and
    - location addressing.
- All communication in the Internet must use the IP protocol. The IP addresses are used by the IP protocol to route messages from source to the destination through the Internet
    - □ IPv4 →
        - uses 32-bit addresses; the address space is limited to 4,294,967,296 (2^{32}) possible unique addresses.
        - addresses for special purposes: private networks (~18 million addresses); multicast addresses (~270 million addresses).
        - addresses represented in dot-decimal notation e.g., 218.96.17.12).
    - □ IPv6 →
        - uses 64-bit addresses; the address space is limited to 2^{64} possible unique addresses.
        - No ‘’flag day”

Domain Name System

- Domain Name System (DNS → general-purpose name management system
  - Hierarchically structured
  - Maps user-friendly host names to IP addresses
- Domain Name Service (DNS)
  - A database editor generates tables of bindings and these bindings and then these tables are distributed to DNS servers
  - Propagation takes time, hours.
  - Supports both relative and absolute paths
- DNS architecture → a hierarchical distributed database and an associated set of protocols that define:
  - A mechanism for querying and updating the database.
  - A mechanism for replicating the information in the database among servers.
  - A schema of the database.
- DNS has a referral architecture somewhat complicated due to need to optimize.

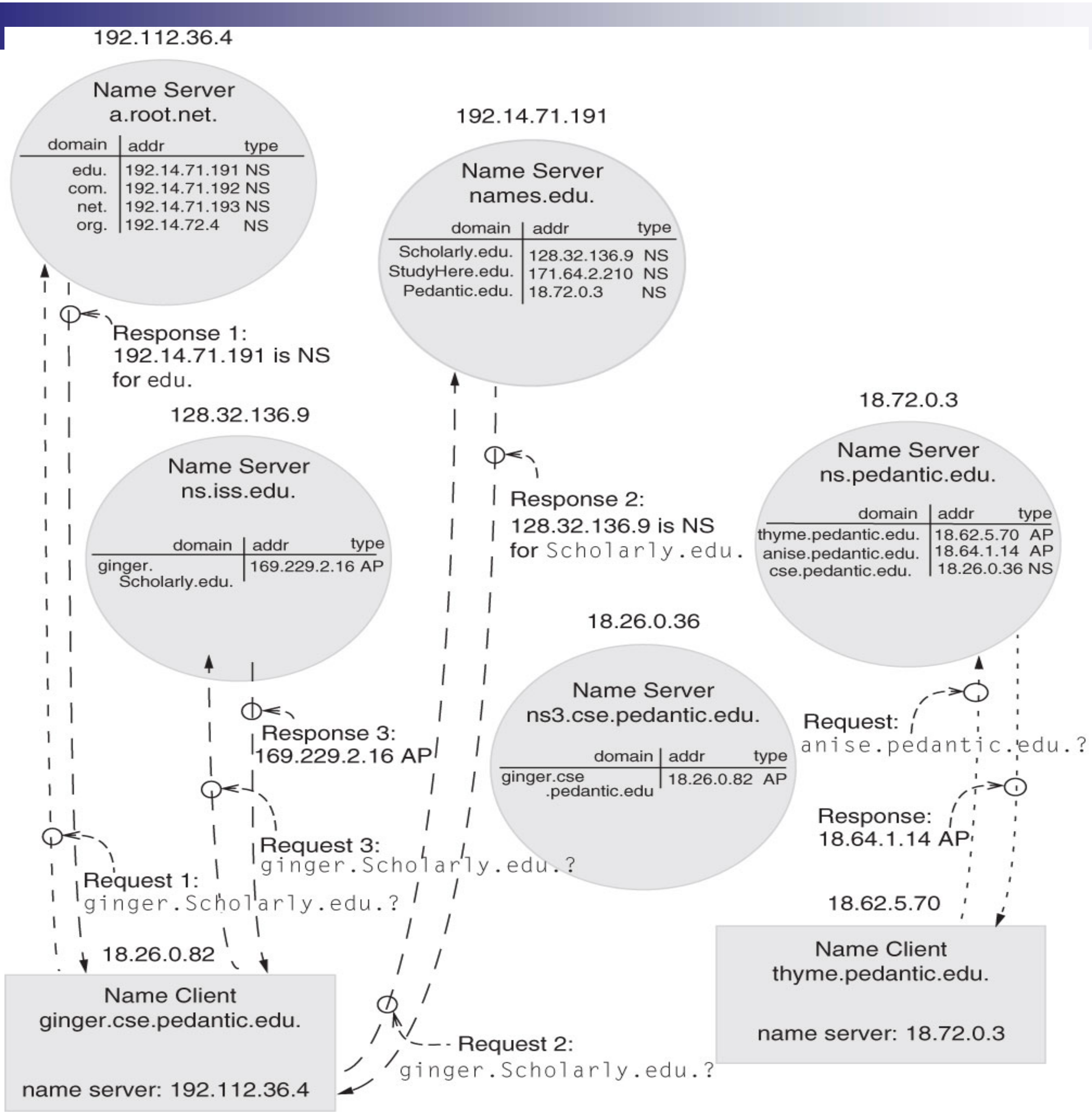
.

DNS Dictionary

- Domain name → an identification label that defines a realm of administrative autonomy, authority, or control in the Internet, based on the Domain Name System. The top-level domains (TLDs) are the highest level of domain names of the Internet; they form the DNS root zone. There are 20 generic top-level domains and 248 country code top-level domains
- Authoritative name server→ gives original, *first-hand*, definitive answers; holds either the name record or a referral record for the name
- Authoritative record →first hand information about a host name
- Naming authority → an Internet administrative authority allowed to add authoritative records to a name server
- Referral record → binds a hierarchical region of the DNS name space to another server that could help resolve the name
- Recursive name service →a DNS server takes upon itself to resolve a name rather than provide a referral record.
- Idempotent action → action that can be interrupted and restarted from the beginning any number of times and still produce the same result as if the action had run to completion without interruption

How DNS works

- A client sends a request to resolve a name to a Domain Name server
- The server examines the collection of the domains it is responsible for
  - If it finds the name record it returns the record
  - Else it searches a set of referral records
  - Starts with the most significant component of the requested domain name for the one that matches the most components and
    - If found it returns the name record
    - Else returns "not found"
- Example on the next slide (left diagram): the system ginger.cs.pedantic.edu tries to resolve the name ginger.Scholarly.edu
- Important → each host must have the address of a domain name server when it is connected to the Internet. This address could be :
  - provided by the ISP (Internet Service Provider)
  - hardwired into the browser
  - generated when the system was installed
  - selected by the user



192.112.36.4

Name Server
a.root.net.

domain	addr	type
edu.	192.14.71.191	NS
com.	192.14.71.192	NS
net.	192.14.71.193	NS
org.	192.14.72.4	NS

Response 1:
192.14.71.191 is NS
for edu.

192.14.71.191

Name Server
names.edu.

domain	addr	type
Scholarly.edu.	128.32.136.9	NS
StudyHere.edu.	171.64.2.210	NS
Pedantic.edu.	18.72.0.3	NS

Response 2:
128.32.136.9 is NS
for Scholarly.edu.

128.32.136.9

Name Server
ns.iss.edu.

domain	addr	type
ginger.Scholarly.edu.	169.229.2.16	AP

Response 3:
169.229.2.16 AP

18.72.0.3

Name Server
ns.pedantic.edu.

domain	addr	type
thyme.pedantic.edu.	18.62.5.70	AP
anise.pedantic.edu.	18.64.1.14	AP
cse.pedantic.edu.	18.26.0.36	NS

18.26.0.36

Name Server
ns3.cse.pedantic.edu.

domain	addr	type
ginger.cse.pedantic.edu	18.26.0.82	AP

Request 3:
ginger.Scholarly.edu.?

Request 1:
ginger.Scholarly.edu.?

18.26.0.82

Name Client
ginger.cse.pedantic.edu.

name server: 192.112.36.4

Request 2:
ginger.Scholarly.edu.?

Request:
anise.pedantic.edu.?

Response:
18.64.1.14 AP

18.62.5.70

Name Client
thyme.pedantic.edu.

name server: 18.72.0.3

The virtues of DNS

- Distributed responsibility → any DNS name server may act as a naming authority and
  - add authoritative records (see example on the previous slide, the right diagram)
  - create lower-level naming domains; e.g., UCF can create EECS, EECS can create ComputingFrontiers, etc.
- Robustness→
  - High level of replication of the name servers
    - There are some 80 replicas of the root name server
    - Each organization with a name server has 2-4 replicas
  - Stateless name servers → does not maintain any state, its public interface is idempotent
  - A DNS server is a dedicated computer running a relatively simple code, thus less likely to fail

More virtues and some failings of DNS

- Flexibility →
  - The same name may be bound to several IP addresses. Needed to
    - ensure replication of services
    - improve performance → see for example the content delivery services provided by akamai
  - Allows synonyms
    - a computer may appear to be in two different domains
    - Indirect names
- Lack of authentication → DNS does not use protocols to authenticate the response to a DNS request. One can impersonate a DNS server and provide a fake response.
- Does not guarantee accuracy →a DNS cache may hold obsolite information