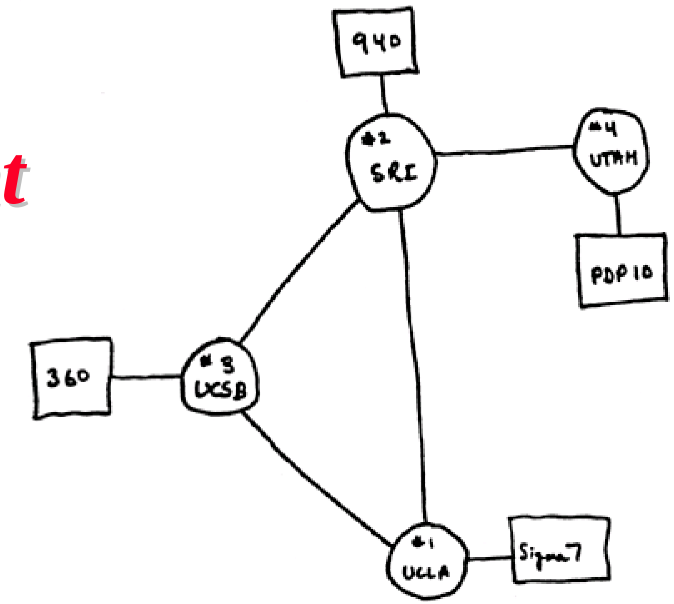
Internet 3.0:

Ten Problems with Current Internet Architecture and Solutions for the Next Generation



Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

These slides are available on-line at:

http://www.cse.wustl.edu/~jain/talks/in3_mc.htm

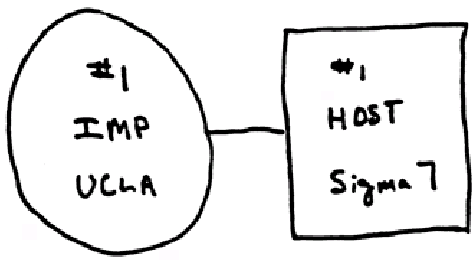
- What is Internet 3.0?
- Why should you keep on the top of Internet 3.0?
- What are we missing in the current Internet?
- Our Proposed Architecture for Internet 3.0: GINA

What is Internet 3.0?

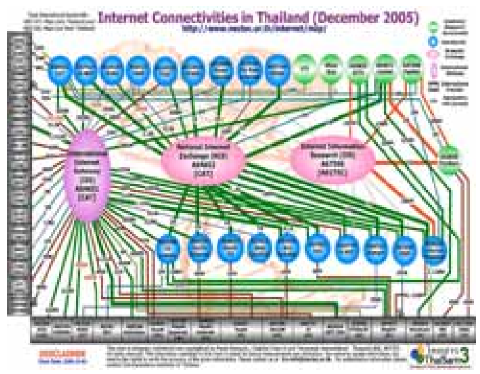
- Internet 3.0 is the next generation of Internet
- Named by me along the lines of “Web 2.0”
- Also known as “Global Environment for Networking Innovations” or GENI
  (Internet 3.0 is more intuitive then GENI)
- National Science Foundation is planning a $300M+ research and infrastructure program on GENI
  ⇒ Most of the networking researchers will be working on GENI for the coming years
- Ref: http://www.nsf.gov/cise/geni/

Internet Generations

- **Internet 1.0** (1969 – 1989) – Research project
  - RFC1 is dated April 1969.
  - ARPA project started a few years earlier
  - IP, TCP, UDP
  - Mostly researchers
  - Industry was busy with proprietary protocols: SNA, DECnet, AppleTalk, XNS



- **Internet 2.0** (1989 – Present) – Commerce ⇒ new requirements
  - Security RFC1108 in 1989
  - NSFnet became commercial
  - Inter-domain routing: OSPF, BGP,
  - IP Multicasting
  - Address Shortage IPv6
  - Congestion Control, Quality of Service,…



Ten Problems with Current Internet

- Assumes live and awake end-systems
  Does not allow communication while sleeping
  Many energy conscious systems today sleep.

- Identity and location in one (IP Address)
  Makes mobility complex.

- Location independent addressing
  ⇒ Most services require nearest server.
  ⇒ Also, Mobility requires location

- Single-Computer to single-computer communication ⇒ Numerous patches need for communication with globally distributed systems.

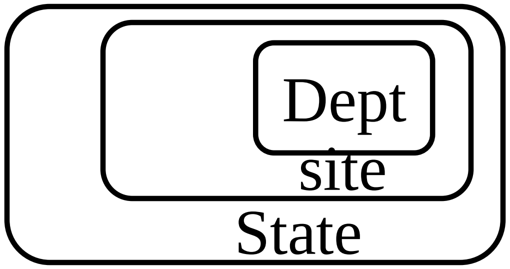
Problems (cont)

- No representation for real end system: the human.

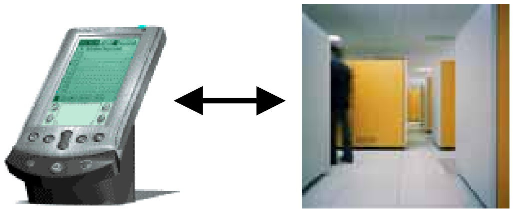
- Designed for research
  ⇒ Trusted systems
  Used for Commerce
  ⇒ Untrusted systems
- Control, management, and Data path are intermixed ⇒ security issues

Problems (cont)

- Difficult to represent organizational, administrative hierarchies with just two levels: domain and inter-domain



- Symmetric Protocols
  ⇒ No difference between a mote and a Google server.

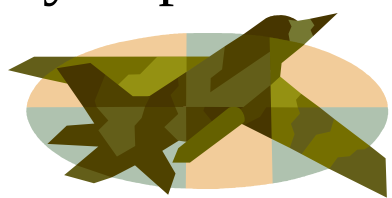


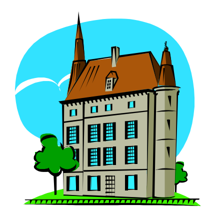
- Stateless ⇒ Can't remember a flow ⇒ QoS difficult.
  QoS is generally for a flow and not for one packet

Our Proposed Solution: GINA

- **Generalized Inter-Networking Architecture**
- Take the best of what is already known
  - Wireless Networks, Optical networks, …
  - Transport systems: Airplane, automobile, …
  - Communication systems: Wired Phone networks, Cellular networks,…
- Develop a consistent general purpose, evolvable architecture that can be customized by implementers, service providers, and users





Names, IDs, Addresses

Name: John Smith

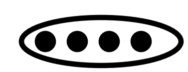
ID: 012-34-5678

Address:
1234 Main Street
Big City, MO 12345
USA

- Address changes as you move, ID and Names remain the same.
- **Examples**:
  - Names: Company names, DNS names (google.com)
  - IDs: Cell phone numbers, 800-numbers, Ethernet addresses, Skype ID, VOIP Phone number
  - Addresses: Wired phone numbers, IP addresses

Objects in GINA

- Object = Addressable Entity
- Current: End-Systems and Intermediate Systems
- GINA:
  - Computers, Routers/Firewalls….
  - Networks
  - Humans
  - Companies, Departments, Cities, States, Countries, Power grids
  - Process in a computer
  - Recursive ⇒ Set of Objects is also one object, e.g., Networks of Networks



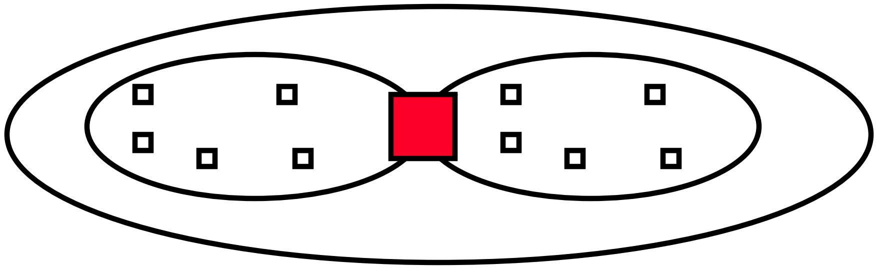
You can connect to a human, organization, or a department

Names, Ids, Addresses, and Keys

- Each Object has:
  - Name**s**: ASCII strings for human use
  - ID**s**: Numeric string for computer use
  - Addresse**s**: where the Object is located
    - Home Address, Current Address
  - Keys: Public, Private, Secret
  - Other attributes, Computer Power, Storage capacity
- Each object has one or more IDs, zero or more names, one or more addresses and zero or more other attributes

You connect to an ID not an address ⇒ Allows Mobility

Realms

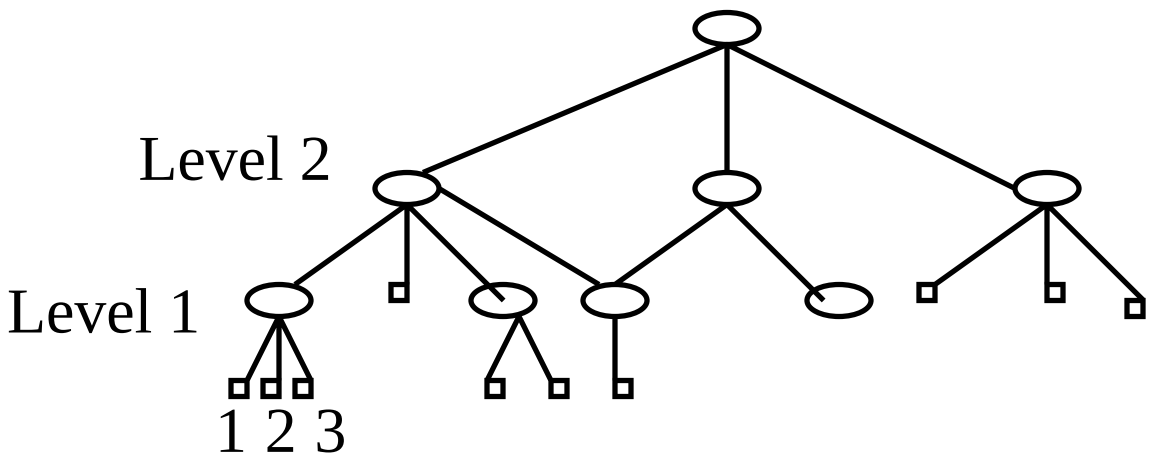


- Object names and Ids are defined within a realm
- An object may be a member of multiple realms.
  ⇒ One or more Ids in each realm of which it is a member
- Each realm has a set of exits. Objects with local realm Ids communicate to objects outside the realm only by simply communicating with server objects at the exit.
- Realms can be treated as single object and have Names, Ids, addresses. Realms are recursive.
- Boundaries: Technological, Governmental, ISP, Organizational

Realm = Organization

Hierarchy of IDs

- Universe is organized as a hierarchy of realms
- Each realm has a set of parents and a set of children
- Parent Ids can be prefixed to realm ids
- A child may have multiple parents ⇒ Hierarchy is not a tree
- Any path to the root of a level gives the ID for the object at that level, e.g., level2_id.level1_id…object_id = level2 id of object



Realm Hierarchy = Organizational Structure

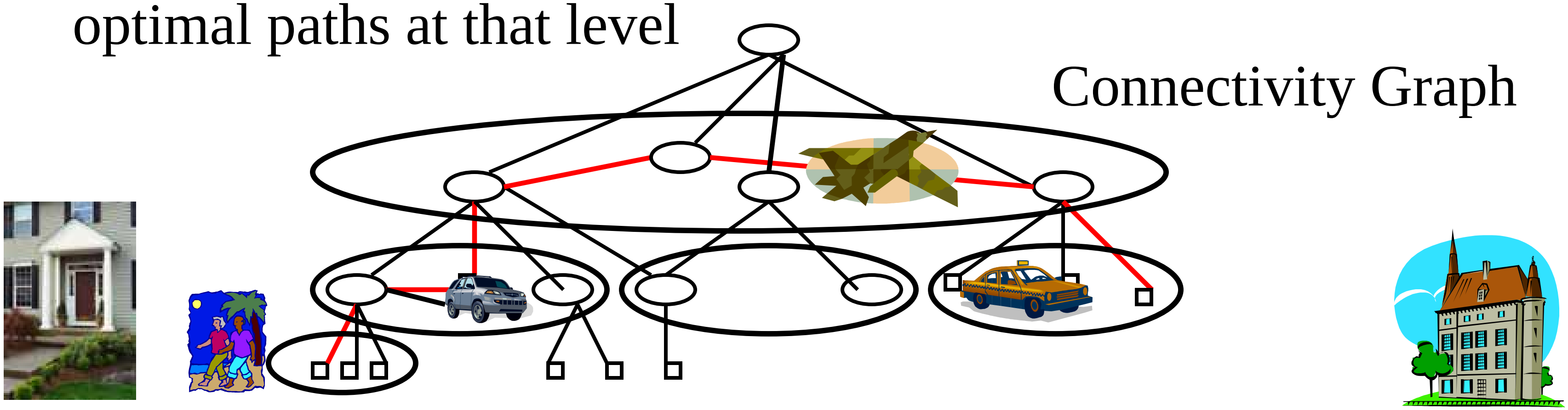
Object Addresses

- Address of an object indicates its *physical attachment point*
- Networks are organized as a set of *zones*
- Object address in the current zone is sufficient to reach it inside that zone
- Each object registers its names, addresses, IDs, and attributes with the registry of the relevant realms
- Zones are objects and have Ids, realms, addresses too
- An object's address at higher level zones is obtained by prefixing it with of addresses of ancestor zones

Zonal Hierarchy = Network Structure

Routing

- Based on connectivity
- Routing organized as paths through several levels of hierarchy
- At each level packets follow an optimal path from the entry point to that level to exit point in that zone
- Routing table exchanges at each level are used to find the optimal paths at that level



Highly scalable hierarchical routing

Server Objects

- Each realm has a set of server objects, e.g., forwarding, authentication, encryption,
- Some objects have built-in servers, e.g., an "enterprise router" may have forwarding, encryption, authentication services.
- Other objects rely on the servers in their realm
- Forwarding servers are located at the boundary of two realms
- Encryption servers encrypt the packets
- Authentication servers (AS) add their signatures to packets and verify signatures of received packets..
- Storage servers store packets while the object may be sleeping and may optionally aggregate/compress/transform/disseminate data. Could wake up objects.
- Persistent connections: Across system restarts, HW replacement, Object mobility

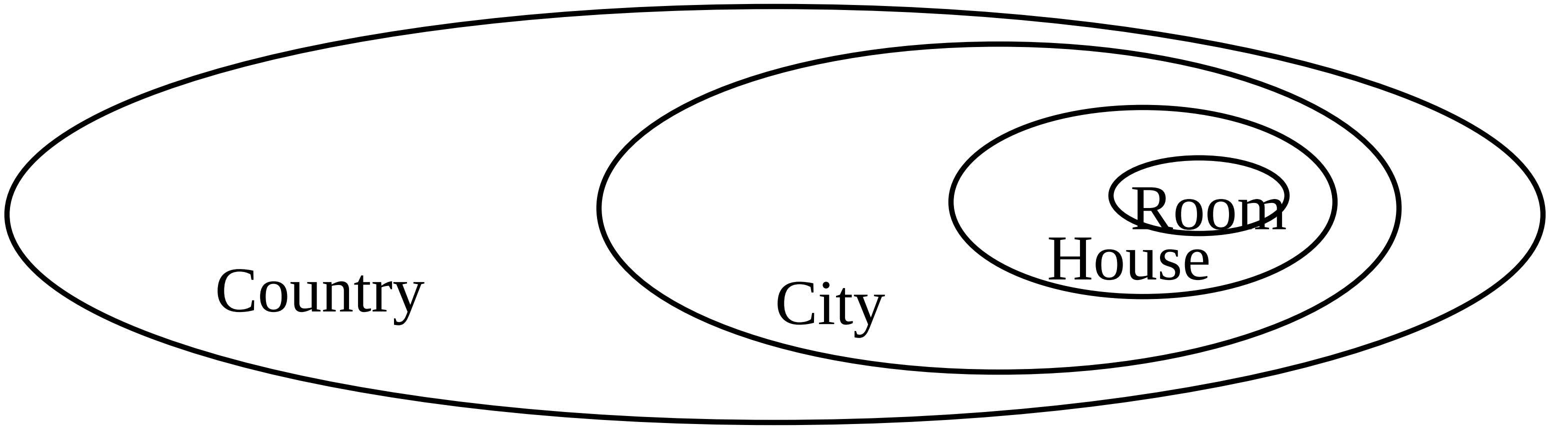
Servers allow simple energy efficient end devices

Packet Headers

- You have to know the name of the destination to be able to communicate with it.
- The destination name has to be up to the level where you have a common ancestor.
- The names can be translated to the ID of the destination by using registries at appropriate levels
- The packets contain either Ids or addresses of the destination
- Current level Ids are translated to address

Packets contain IDs $\Rightarrow$ Network handles mobility

Security



- Multi-level architecture. Gatekeepers on the entrance
- Authentication checked on entry to zone/realm.
  Not at every router.
- Authentication at multiple levels: country, city, home.
- Group Authentication: n-packets can be authenticated by one authentication
- VPN and firewalls are part of the architecture

Organizational control of security

Gatekeepers

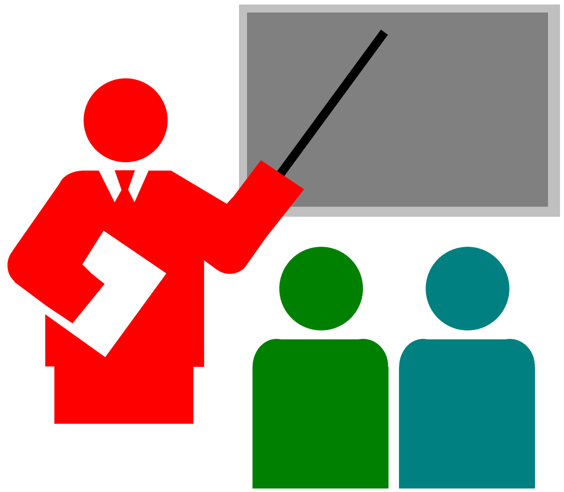
- Gatekeepers also enforce policies and do policing (Monitor bandwidth, type of traffic, contents)
- May provide storage for a limited time (Helps sleeping entities save energy)
- Add authentication headers (country, city, home, level)
- End systems can delegate the "TCP" responsibility on gatekeepers
- All services do not have to have reside in each gatekeeper.
- Gatekeepers may also delegate services to other servers
- Application-specific gatekeepers

Organizational control of all policies

Internet 1.0 vs. Internet 3.0

	Feature	Internet 1.0	Internet 3.0
1.	Energy Efficiency	Always-on	Green ⇒ Mostly Off
2.	Mobility	Mostly stationary computers	Mostly mobile *objects*
3.	Computer-Human Relationship	Multi-user systems ⇒ Machine to machine comm.	Multi-systems user ⇒ Personal comm. systems
4.	End Systems	Single computers	Globally distributed systems
5.	Protocol Symmetry	Communication between equals ⇒ Symmetric	Unequal: PDA vs. Google ⇒ Asymmetric
6.	Design Goal	Research ⇒ Trusted Systems	Commerce ⇒ No Trust Map to organizational structure
7.	Ownership	No concept of ownership	Hierarchy of ownerships, administrations, communities
8.	Sharing	Sharing ⇒ Interference, QoS Issues	Sharing *and* Isolation ⇒ Critical infrastructure
9.	Switching units	Packets	Packets, Circuits, Wavelengths, Electrical Power Lines, …
10.	Applications	Email and Telnet	Information Retrieval, Distributed Computing, Distributed Storage, Data diffusion

Summary



- Internet 3.0 is the next generation of Internet.
- It must be green (energy efficient), secure, allow mobility.
- Must be designed for commerce.
- Active industry involvement in the design essential. Leading networking companies must actively participate.
- Our proposal Generalized InterNet Architecture (GINA) addresses many issues.