

# Serval

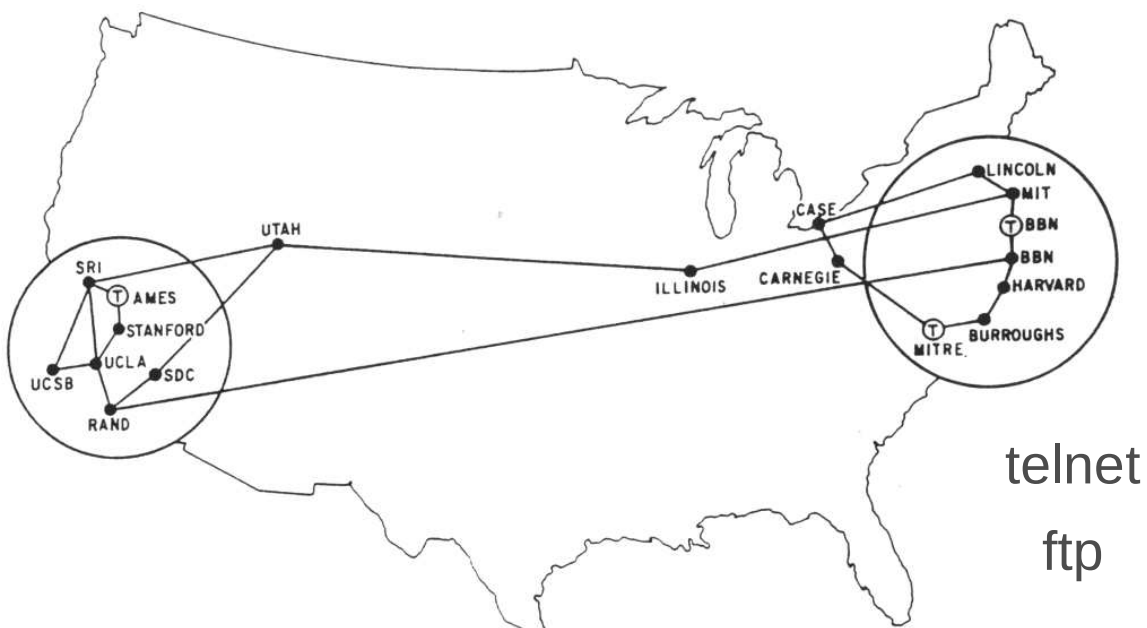


A Leap Toward Next-Generation Service-Centric Networking

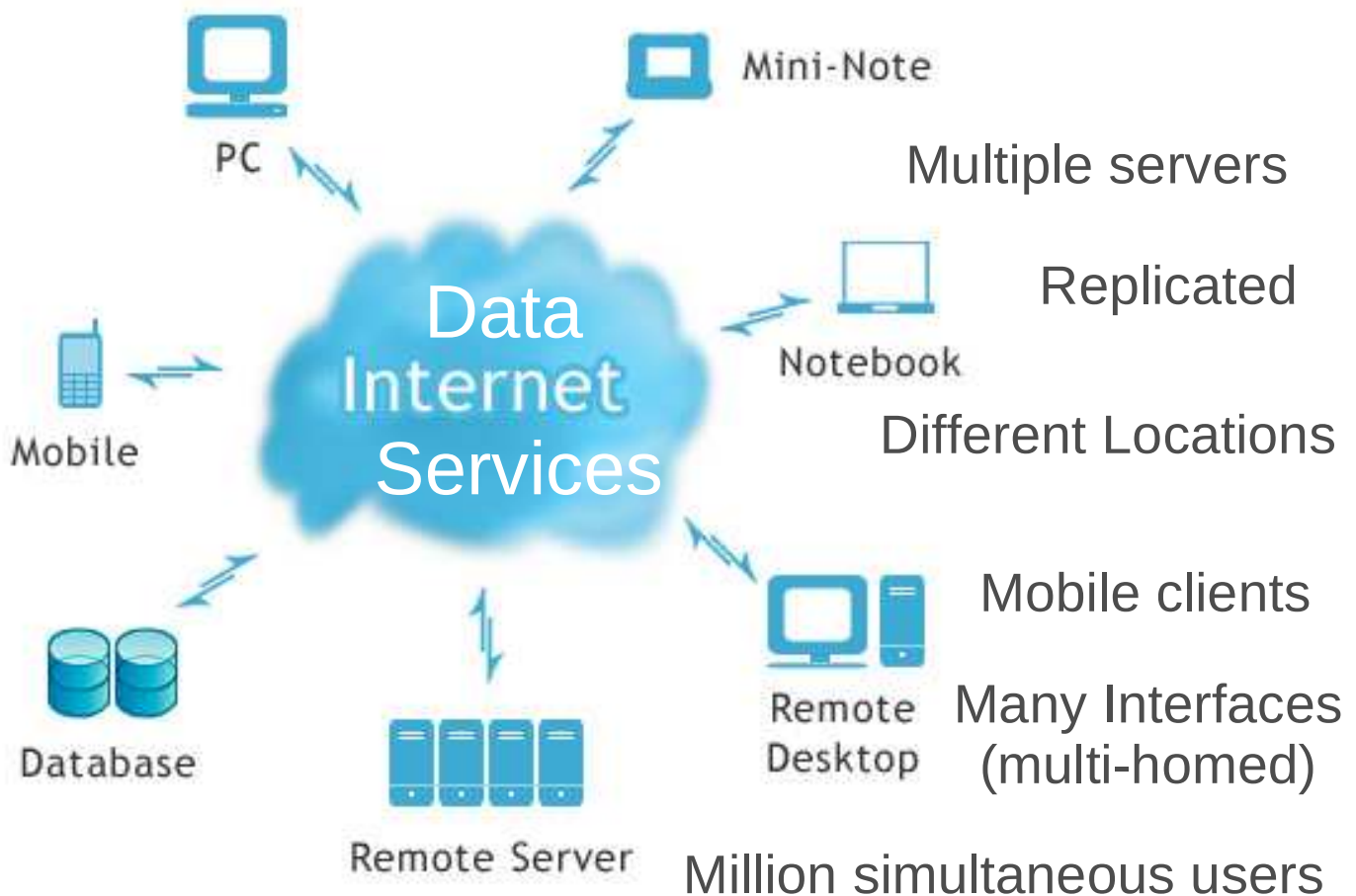
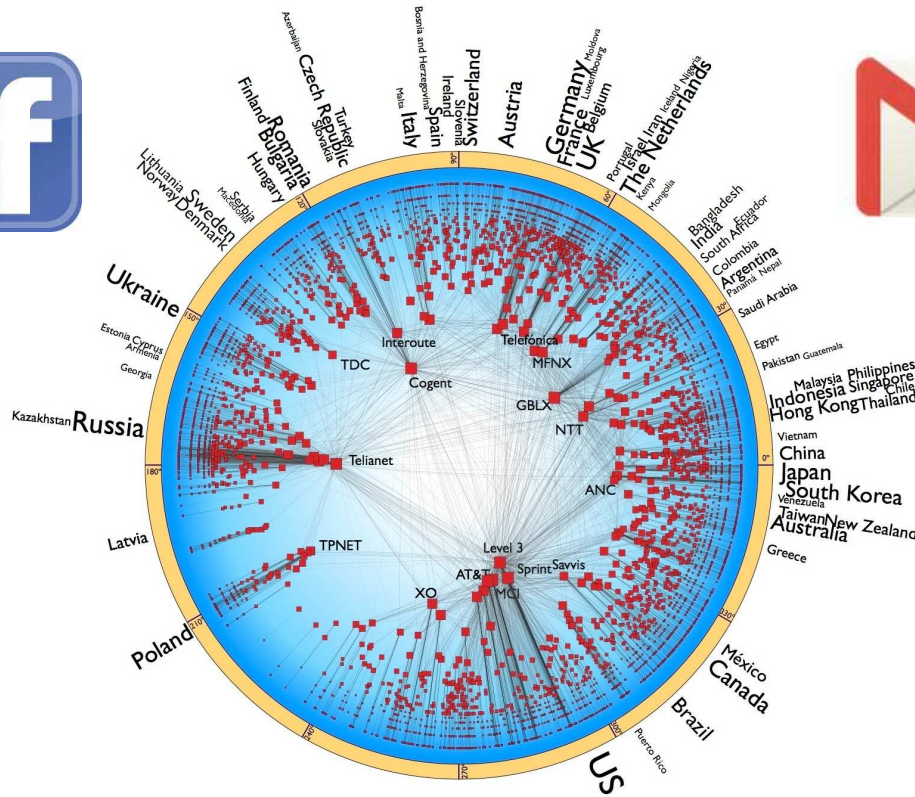
Isaakidis Marios  
Cyprus University of Technology



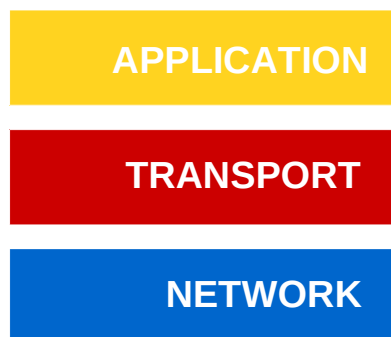
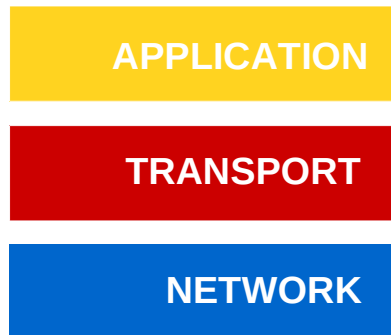
## Initial Internet topology



# The (hyperbolic) Internet today



# The Network Stack remains the same! ...creating problems



Hierarchical IP addressing  
**end-host mobility?**

APPLICATION

TRANSPORT

NETWORK

Demultiplexing

(remote IP, remote port, local IP, local port, protocol)

change IP address ? migration?

change interface?

load balance every  
single packet?

applications must know  
(remote IP – remote port)

APPLICATION

TRANSPORT

NETWORK

Applications “early-bind” on IP address  
and TCP/UDP ports.

out-of-band look-up mechanisms or  
a priori knowledge?

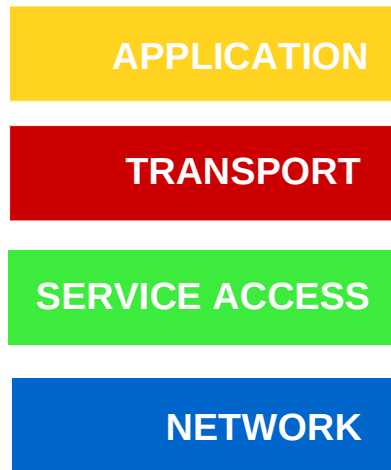
caching of ip-port?

multiple services? slow failover

specifies application protocol

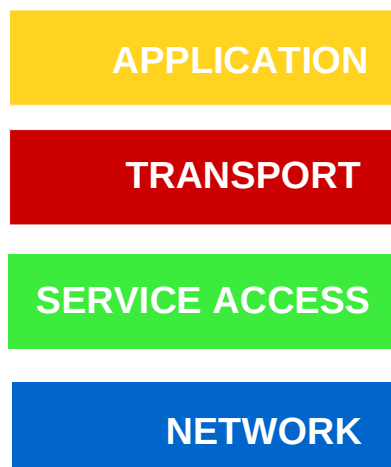
do we really need to know  
server's location?

# Proposal : Service Access Layer



Erik Nordström   Michael J. Freedman

# Proposal : Service Access Layer



**Connects to the server**  
**Maintains connectivity**

# Serval Abstractions : **Service Names**

- Human-readable, correspond to a real service (e.g. facebook)
- Reference a group of processes offering the same service
- Mapped to machine-readable 256-bit **serviceIDs** (through “DNS”, flooding or DHTs), allocated in blocks

## **Why?**

- Decouple services from their location
- Warrant late binding
- Enable service-level routing and forwarding
- Make load-balancing easier
- Help with building and managing services

# Serval Abstractions : **Flows**

- (dstIP, dstPort, srcIP, srcPort, protocol) -> flowID
- Do not encode the application protocol
- There is no need for port numbers
- Help when interacting with middleboxes
- Are network-layer oblivious (same for IPv4 and IPv6)
- Help maintain connectivity
- Reduce the overload of identifiers within and across the layers
- Are host-local and ephemeral
- Empower migration, mobility and multiple paths

# Serval Abstractions



**WHO?**

Service names

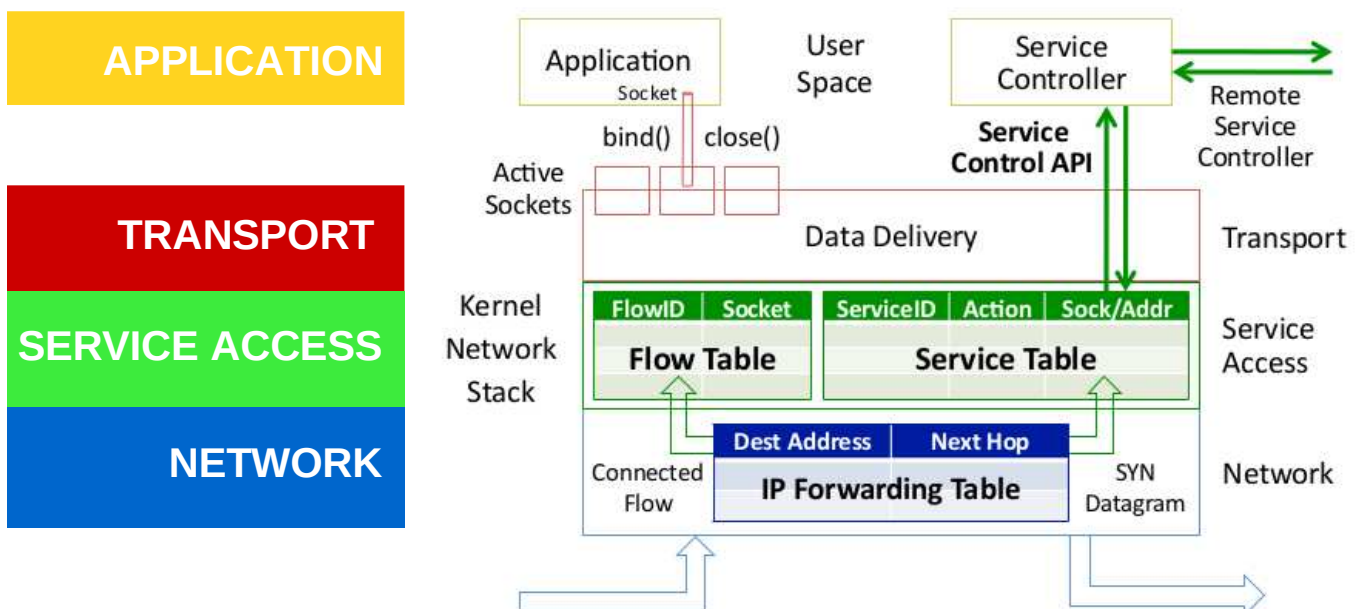
**HOW?**

Flows

**WHERE?**

Addresses

## Service Access Layer (SAL)



A clean service-level control/data plane split



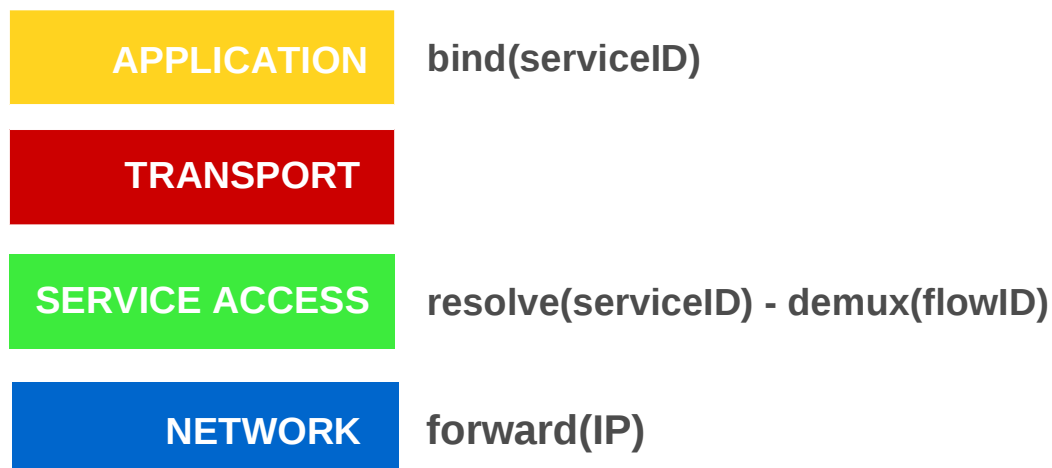
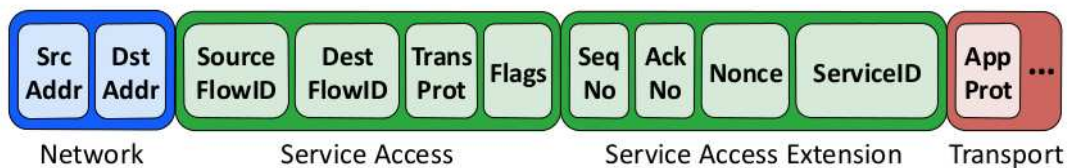
## Service Controller (control)

- Runs on the user-space level
- Manages **service resolution**
- Listens for service-related events
- Monitors service performance
- Communicates with other controllers

## Service Access Layer (data)

- Maintains the **service table** and **flow table**
- Communicates with applications using **Active Sockets**
- Late binds connections to services and maintains them across changes between interfaces
- Load balance between a pool of replicas

## Service Access Layer (SAL)



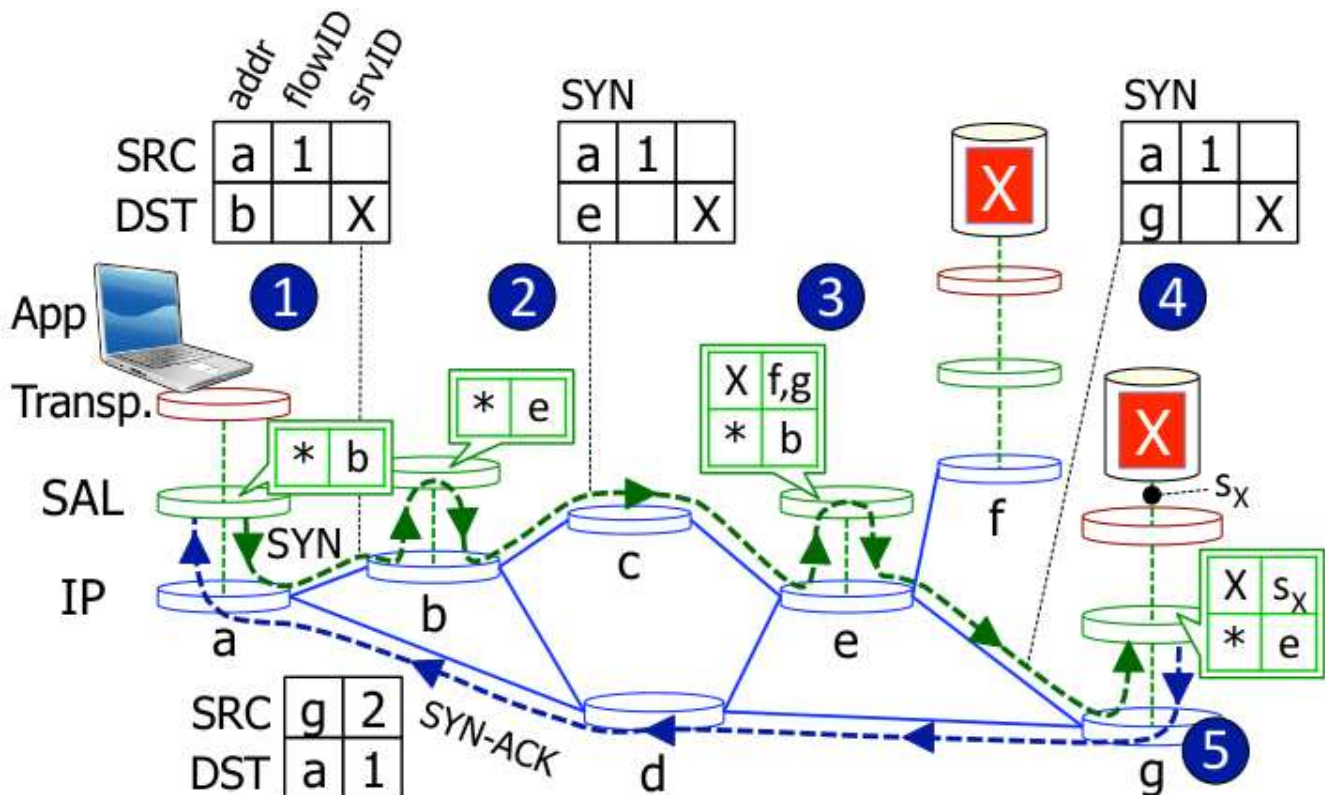


# Programming with the Serval Active Sockets API

PF_INET	PF_SERVAL
s = socket(PF_INET)	s = socket(PF_SERVAL)
bind(s, locIP:port)	bind(s, locServID)
//Datagram	//Unconnected Datagram
sendto(s, IP:port, data)	sendto(s, srvID, data)
//Stream	//Connection
connect(s, IP:port)	connect(s, srvID)
accept(s, &IP:port)	accept(s, &srvID)
send(s, data)	send(s, data)

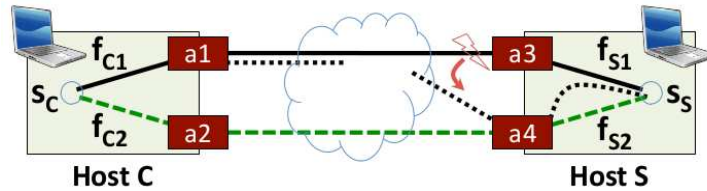
Firefox Browser needs only 70 changes o\_O

## A Service-Level Data Plane



# Multiple Flows and Migration

- Control messages are separate from the data stream
- By managing flows in a separate layer, Serval can support transport protocols other than TCP



- **Multihoming and Multipathing**
  - Serval can split a socket's data stream across multiple flows established and maintained by the SAL on different paths.
- (Virtual Machine) **Migration**
  - Since the transport layer is unaware of flow identifiers and interface addresses, the SAL can freely migrate a flow from one address, interface or path to another.

## ServalDHT

A leap Toward Freedom in  
Service-Centric Networking

# The future of Internet

“The **Open Internet** Is Threatened by UN's Closed-door Meeting in Dubai”

– Don Tapscott, 19h ago

“Who **oversees the internet?**”

Gulfnews.com, 23mins ago

“61 countries able to instantly **unplug from the Internet**”

– DAILY NEWS, 16h ago

“UN **Control of the Internet?** An Idea Whose Time Will Never Come”

– The Internationalist, 12h ago

“The Future Of **Internet Freedom**”

– Tom Ashbrook, 5h ago

## Declaration of Internet Freedom

- **Expression:** Don't censor the Internet.
- **Access:** Promote universal access to fast and affordable networks.
- **Openness:** Keep the Internet an open network where everyone is free to connect, communicate, write, read, watch, speak, listen, learn, create and innovate.
- **Innovation:** Protect the freedom to innovate and create without permission. Don't block new technologies, and don't punish innovators for their users' actions.
- **Privacy:** Protect privacy and defend everyone's ability to control how their data and devices are used.

**[don't censor the web]**

# Chord

- A scalable peer-to-peer lookup service
- Given a key, it maps the key onto a node
- Requires no special servers
- Adapts efficiently as nodes join and leave
- In an N-node system each node maintains information about  $O(\log N)$  other nodes and resolves all lookups via  $O(\log N)$  messages to other nodes.
- Needs routing information about only a few other nodes
- Unlike DNS, Chord can be used to find data not tied to a particular machine

# ServalDHT

- DHT – Distributed Hash Tables
- A decentralized DHT service resolution service (SRS)
- Gets serviceID and
  - either returns the (IP,Port) back to the client
  - or forwards the packet directly to the service provider
  - caches the (serviceID, IP) tuple for future use
- Flat namespace (with SHA-256 ( $n=256$ ) and one billion messages ( $p=10^9$ ) then the probability of collision is about  $4.3 \cdot 10^{-60}$ )
- Users register services with their public keys
- Incrementally deployable
- Can be utilized by independent networks for service (or data) identifier resolution
  - Along with the Internet, the two namespaces should not overlap!

# Serval vs ServalDHT

Serval	ServalDHT
Decouples the transport layer (IPs) from the application one (services)	The same + improves security utilizing cryptographic host identifiers
The service resolution relies on an hierarchical DNS-like service	The service resolution is based on a decentralized peer-to-peer service maintained by (users,) ISPs and tier {1,2,3} operators
serviceIDs are (annually) purchased by an IANA-like organization	Hashed service names get permanently registered as serviceIDs to a public key
The namespace remains hierarchical	ServalDHT's introduced namespace is flat
Prone to DDoS attacks	High robustness against targeted attacks
Requests are routed hierarchically	May cause router stretching
DNS can cache both data and hierarchy info	DHT SRS can proactively cache only (serviceID, IP) tuples

## ServalDHT – Any ideas?

- Flat namespace – block allocation? Last 3 bits?
- **Security? - Host Identity Protocol**
- Registration of service names to the right owners without an IANA-like organization?
- Router stretching?
- Test on PlanetLab?
- OpenFlow support?

# ServalDHT – Next steps

- Read papers relative to
  - Service-defined networks
  - DHTs (Chord, DHT vs DNS, DNS caching etc)
  - Host Identity Protocol
- Implement a DHT Service-Resolution-Service Demo
- Test on PlanetLab (?)
- Prepare the final thesis and a (ready for publication) paper
- Commit an RFC