Name and Addressing in Future Internet

Jun Bi

Tsinghua University

2010.11.03

Outline

- Background
- Requirements of New ID Space
- Identifier Designs, Proposals and Solutions
- Design Suggestions

Background

Namespaces in the Internet Today

- Namespace in global scope
  - MAC addresses: used to identify interfaces in layer two
  - IP addresses: used for data transmission between hosts
  - Domain names: used by applications
  - TCP/UDP ports (together with IP addresses): used to identify processes in application layer
- IP addresses are overloaded
  - Problems in mobility, security and so on mostly regarding IP addresses

Name and Addressing in today's Internet

- IP addresses
  - identify interfaces connected to the Internet
  - Classful scheme: A, B, C, D before year 1992.
  - Class-less Inter-Domain Routing (CIDR, RFC1519)
- Domain names
  - to help people identify the hosts when they change IP addresses.
  - Old solution: A centrally-maintained **`hosts.txt`** file, distributed to all hosts
  - DNS (*Paul Mockapetris,* RFC1034/1035)
    - Distributed database; Replication; Caching
    - A critical piece of today's Internet infrastructure

Today's Multihoming and Mobility

- Multihoming
  - host multihoming
    - A host can have multiple physical interfaces with several logical internet addresses
  - site multihoming
    - A site can connect to more than one providers to reach the Internet
- Mobility
  - dynamic update of host IP addresses
  - IP don't support mobility inherently

ID/Locator overloading

- IP addresses identify attachment points in the network (Locator)
  - IP address space is topology-dependent
- IP addresses are also used to identify TCP connections (addr+port), even the node Identifier by applications
  - It is OK when attachment points of nodes are unique and permanent

ID/Locator Split

- ID spaces should be independent to attachment points
  - Separate the semantic of identifier from IP addresses
- ID spaces will be topology-independent
  - The relationship between an entity and its identifier does not have to change when the entity changes its point of attachment in the network
- ID/Locator split may benefit the routing scalability (the Core-Edge Elimination, CEE, approaches), while it is more for the mobility and security problem

What does mean an ID and Loc?

Discussions in RRG and ITU

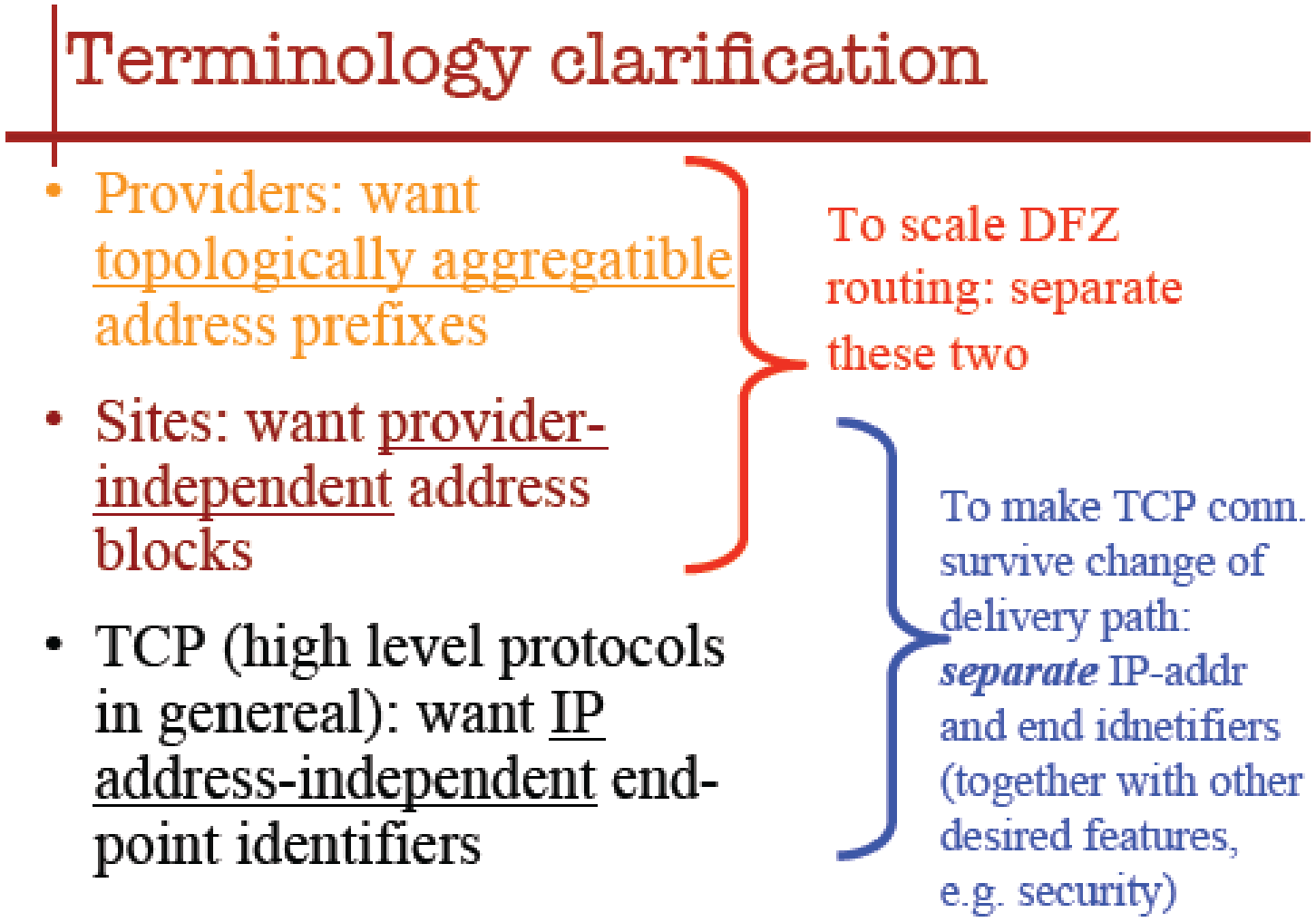
- Locator
  - Topology-dependent
  - Layer 3 by default
  - RRG: name for a point of attachment within the topology at a given layer
  - ITU: Topological name for an interface or a set of interfaces
- ID
  - Topology-independent
  - RRG: name of an object at a given layer
  - ITU: Node ID used at the transport and higher layers to identify the node as well as the endpoint of a communication session

Some Definitions (from ITU-T Rec.)

- Identity
  - A series of digits, characters and symbols or any other form of data used to identify subscriber(s), user(s), network element(s), function(s), network entity(ies) providing services/applications, or other entities (ITU-T Rec. Y.2091 and Y.2720)
- Identifier
  - Information about an entity that is sufficient to identify that entity in a particular context (ITU-T Rec. Y.2720)
- Identification
  - A process to identify the UPT user or the UTP service provider (ITU-T Rec. F.851 and F.852)

- Reference to avoid further confusion

Routing Scalability and ID/Loc split



Source: Lixia Zhang at HIP RG Meeting, IETF68, 2007-3-23

Activities

- IRTF RRG, ITU-T
- In Europe
  - Naming and Addressing in a Future Internet Workshop
  - Dagstuhl Seminars, Wadern, Germany, March 1-4 2009
  - http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=09102
- In Asia
  - Workshop on Identifiers in Future Internet （WIFI）Sponsored by AsiaFI and FIF
  - Seoul, Feb. 24 2010
  - http://www.asiafi.net/meeting/2010/WIFI/main.htm

Questions on Identifiers in FI

- Do we really need identifiers?
  - Problems and requirements from applications/industries on mobility, multi-homing, and security
- What need to be identified and how to design identifiers?
- Architecture design and deployment on identifiers, locator, and mapping between them

Requirements of new ID space

Requirements of new ID space

- Site multi-homing
- Host multi-homing
- Mobility support
- Get around NAT
- Communication authentication
- Multi-device support

Site Multi-homing

- One site accesses multiple ISPs simultaneously
  - Provider-Independent (PI) address: may break aggregation of IP addresses, and cause exploding of global routing table
  - Provider-Allocated (PA) address: multiple addresses, sessions will break when switching between them
- A name space which is topology-independent is needed to identify the hosts in multi-homed site
  - PA address can be used, without destroying routing scalability
  - Upper layer use ID to keep session survivability

Host Multi-homing

- One host accesses the Internet through multiple interfaces
  - Multiple IP addresses in one host, without an unique ID to identify it in communications
  - Sessions will break once data changes incoming interface
- Multi-homed hosts need a new name space to identify themselves
  - Which is independent of their locations

Mobility Support

- Similar to multi-homing
  - New attachment points appear and hosts get new addresses, old ones disappear
- But more dynamic than multi-homing
  - Hosts don’t know new addresses before moving
  - Data transmissions may break before new attachment points appear
- A name space which remains unchanged during moving is required

Get Around NAT

- NAT is widely deployed
  - Hosts behind NAT do not have globally unique IP addresses
  - Which breaks E2E connectivity and brings problems
- This can be handled by a new name space which identifies all the hosts in global scope

Communication Authentication

- Security is a big problem in the Internet
  - E2E authentication is an effective way
  - But binding between IP and hosts are neither unique nor permanent
  - Capacity of authentication based on IP is limited
- New name space may benefit authentication mechanisms
  - Bound ID to a pair of public and private keys, it will be stable when host changes its locations

Multi-device Support

- Multi-device owned by single user will become increasingly popular
  - Why users need to communicate with each other using the host ID or IP address which is connected to a device. Users actually want to communicate with users
  - What about services switched between devices own by a single user?

- Need a new name space which is not bound to locators or devices

Identifier Designs, Proposals and Solutions

Outline

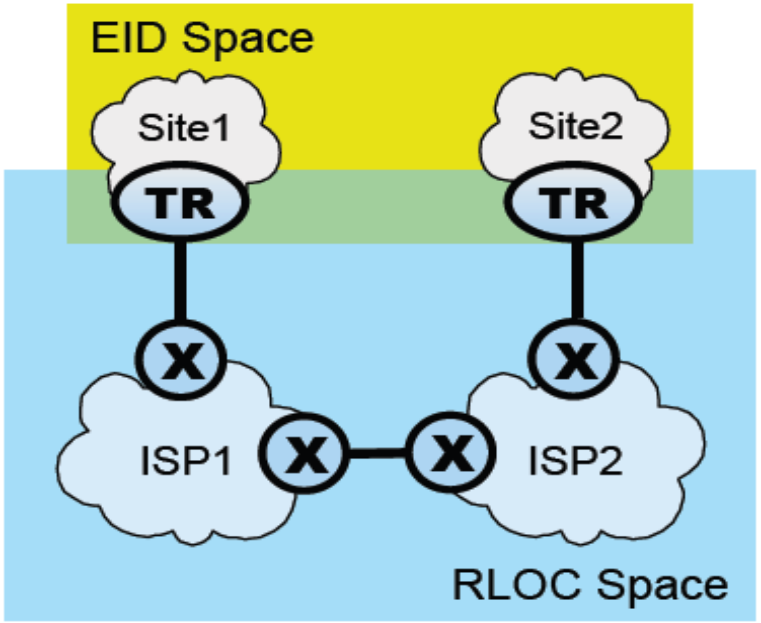
- Some typical proposals
  - Host Identifier
  - Session Identifier
  - Service Identifier
  - Data Identifier
- Other solutions
  - ID design in mobile Internet
  - Some proposed solutions in China

What does ID identify?

- Host Identifier
  - ID for hosts or nodes in the network layer
- Session Identifier
  - ID for sessions in the transport layer
- Service Identifier
  - ID for services in the application layer
- Data Identifier
  - ID for data or content in the network

Host Identifier – Mapping in the network

- Look ID-Loc mapping up in the network
  - Modify network devices, no changes to hosts and apps
- Core-Edge Separation (CES): LISP, Ivip …
  - EID globally unique, identify hosts in the edge
  - RLOC are aggregatable, used for routing in the core
  - Mappings are stored in the boundary
- Not a complete ID/Loc split proposal
  - Routing on EIDs in the edge

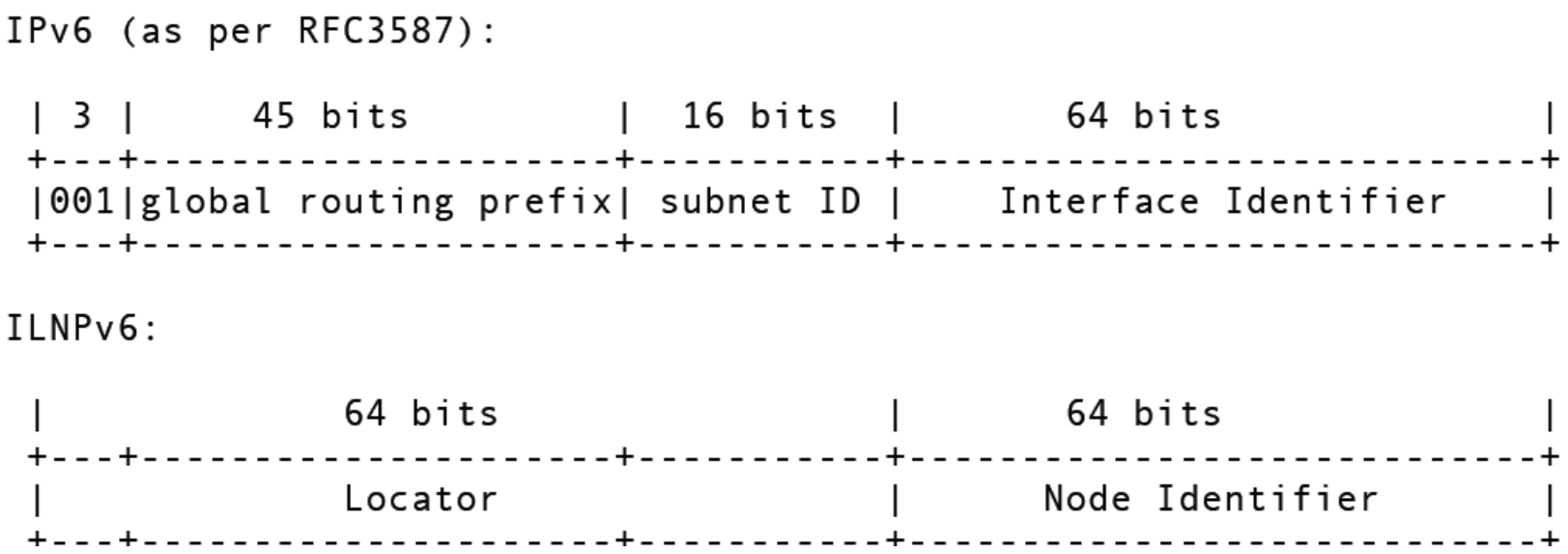


Host Identifier – Mapping in hosts

- Look ID-Loc mapping up by hosts
  - No changes to network devices, while host-change is required
  - Also have to provide mapping service in global scope
- Related proposals
  - ILNP
  - GLI-split
  - HIP
  - RANGI
  - MobileMe

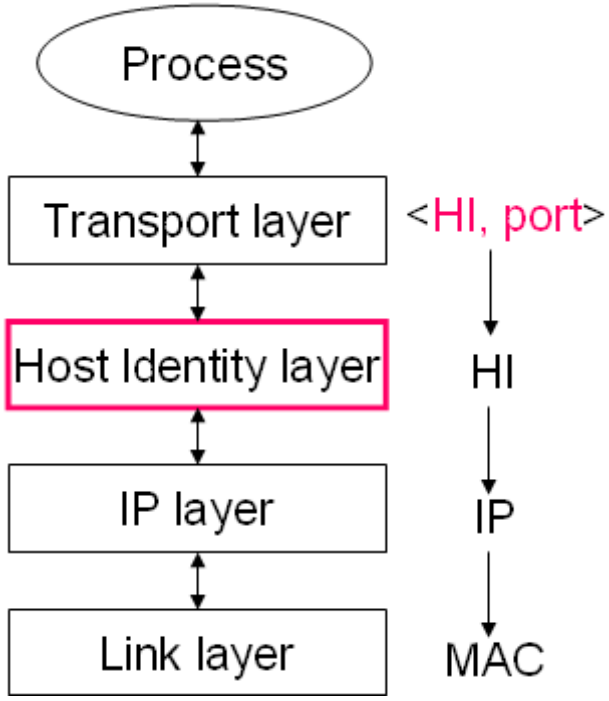
Host Identifier – Mapping in hosts, using IPv6

- ILNP，GLI-split
  - Divide IPv6 address into Identifier part and Locator part (8+8)
  - Session states only contain Identifier part
  - Locator part is used for routing



Host Identifier – Mapping in hosts, with new ID layer

- HIP，RANGI，MobileMe
  - New “Identifier layer” in the stack
  - Use Locators under ID layer
  - Use Identifiers above ID layer



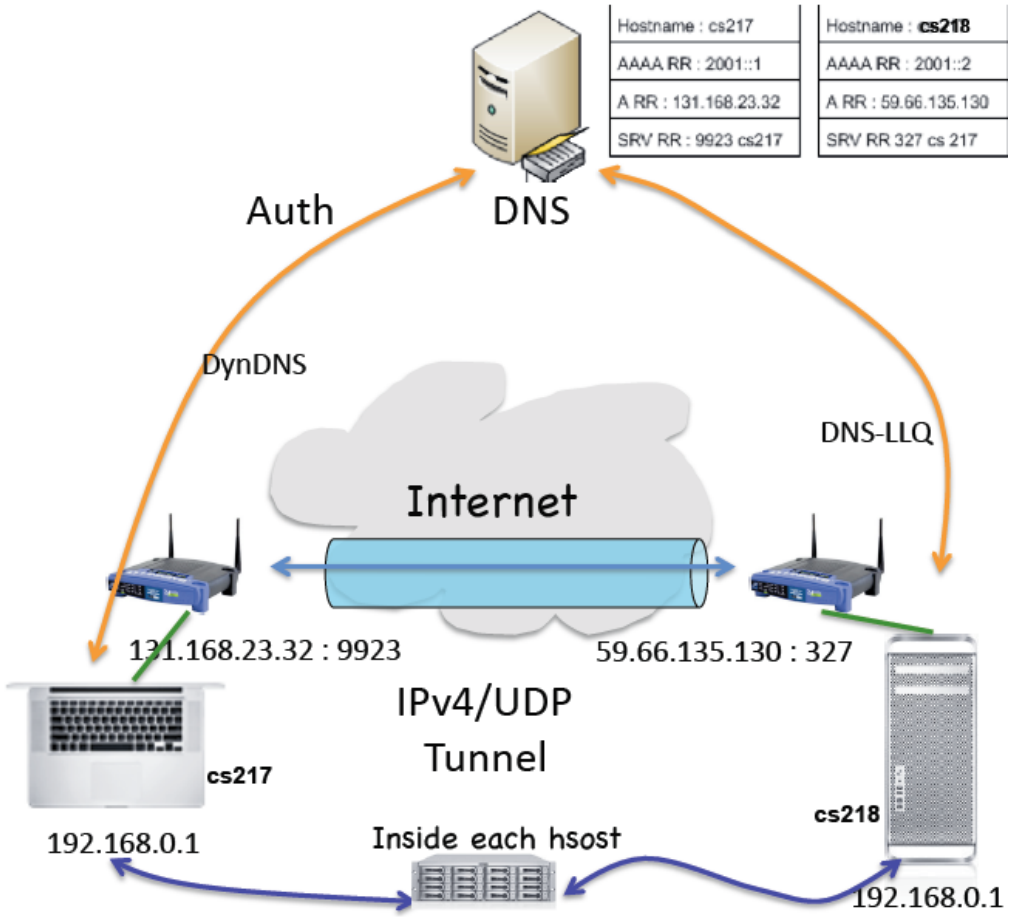
- HIP (Host Identity Protocol)
  - Public Key for Host Identity (HI)
  - Using hash of HI, compatible with IP addresses
  - HI + port for sessions
  - Map HI to IP addresses in the network layer
  - Need global mapping service

MobileMe

- MobileMe
  - Provide Back to My Mac (BTMM) service
  - Identify user devices in global scope, to keep them all in sync
  - Mobility, NAT traversal, some security enhancements
  - No new protocol introduced

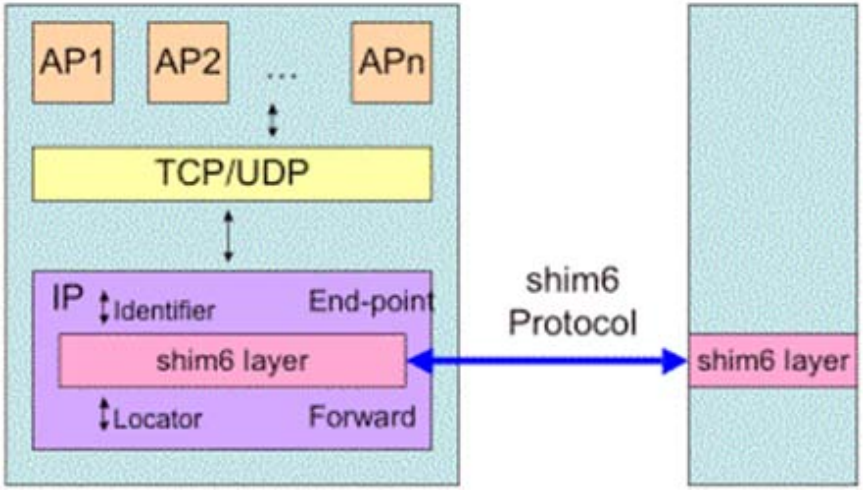
MobileMe

- Implemented In Mac OS
  - Hosts generate IPv6 ULA address and register in the DNS system
  - Mapping from Host ID (IPv6 address) to Loc (IPv4 address) is maintained
  - Port will also be record in DNS to traverse NAT



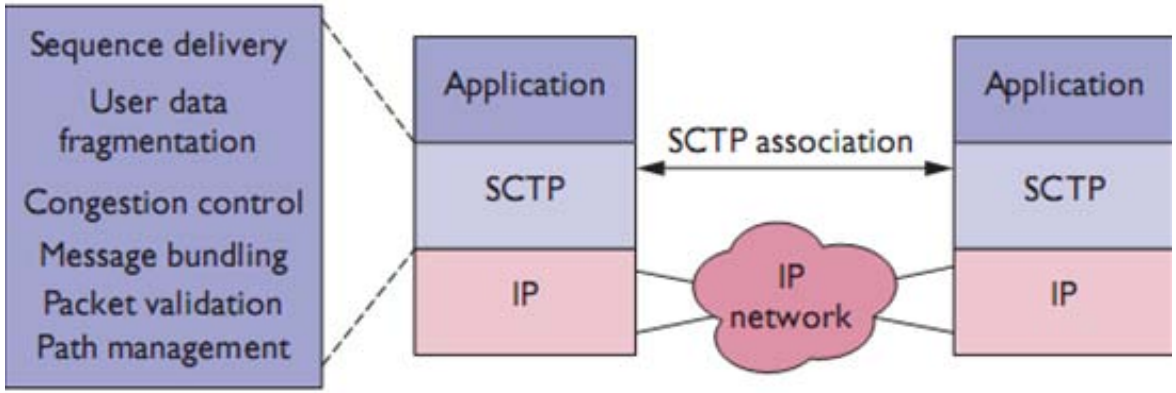
Session Identifier – Based on Network Layer

- Maintain mapping in the network layer
  - ID: only local scope
- Shim6 protocol (RFC5533)
  - ID: A pair of locators used by both sides at the beginning of communication
  - Both sides exchange their locator set
  - Shim6 layer is inserted into the network layer to maintain the mapping



Session Identifier – Based on Transport Layer

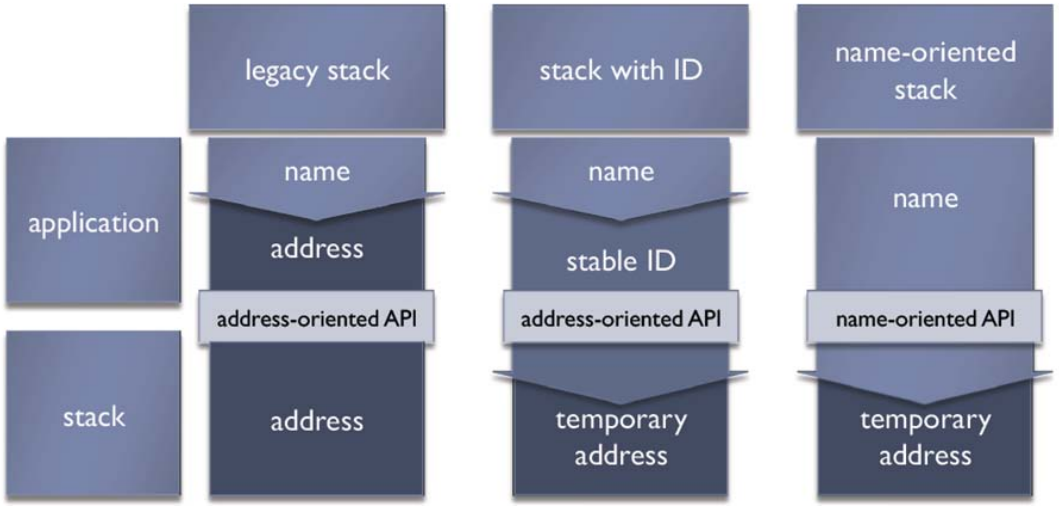
- Modify the transport layer protocol
  - Replace TCP
  - Also, IDs are local
- SCTP (RFC4960)，MultiPath-TCP
  - ID is negotiated by both sides to identify the session
  - Locator sets are exchanged
  - MPTCP can achieve multi-path transport



Service Identifier

- Name the services in the Internet
  - Such as Domain Names

- Name-oriented Stack
  - Use Domain Names as ID
  - Without host ID, map service names directly to IP addresses



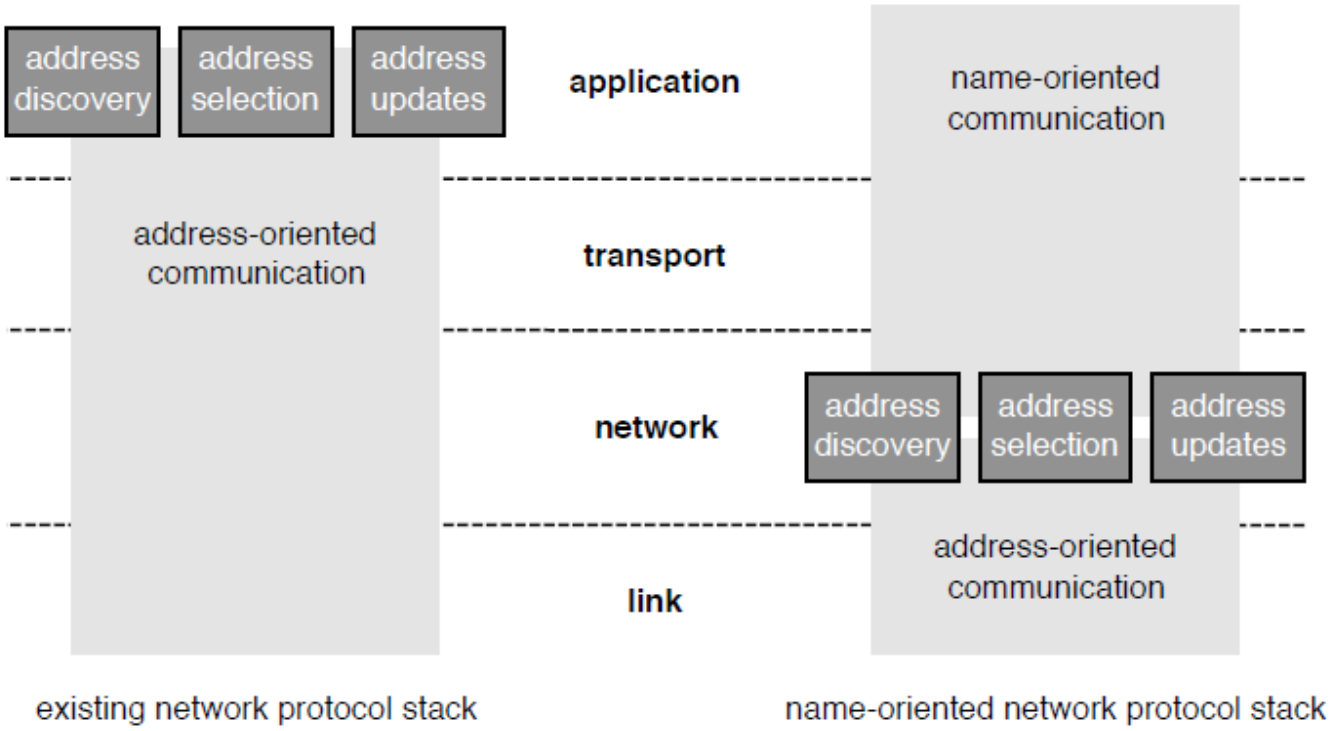
Name-oriented Stack

- In current Internet, applications are responsible of managing the IP addresses in their communication sessions
  - IP address management has become increasingly complex due to multi-homing, mobility and so on

- Name-oriented Stack enables applications to initiate and receive communication sessions by use of DNS names, with
  - Backwards compatibility
  - Name security
  - Deployment incentives

Name-oriented Stack

- Applications and transport protocols operate based on DNS names, all IP address specific functionality is performed at the IP layer
- New API for applications
  - Listen, Open, Accept, Write, Read, Close, …



Data Identifier

- Name the data or content
  - Users care about data and content, not locations or devices
- DONA (Data-oriented network architecture) ,TRIAD
  - Based on IP, map ID to IP before data transmission
  - DONA: self-certifying names (like HIP)
  - TRIAD: URL as names
- i3 project, NDN
  - IP-independent, without mapping
  - i3: structureless names
  - NDN: structured names

Named Data Networking (NDN) - What's the Problem

- Morgan Stanley View:
  - Video streaming is the main contributor for global IP traffic growth in the following years, accounMng for 50% of the total traffic.
- A hot YouTube video downloaded 1,000,000 times from the same servers
  - YouTube request is looking for the data,
  - but the network only knows how to find that data from specific server
- Need new protocols for moving data around for every new application□
  - today's data dissemination is application specific
- Tolerating delay/disruption in data delivery
  - Especially for mobile networking

NDN - How Naming Data Solves the Problem

- Just thinking about routing on data names
- YouTube request reaches a nearby copy of the Data
  - Rather than going to the specific server
- If all requests directed to data nearby:
  - Network natually provides scalable data dissemination
  - Disk technology made massive caching feasible

NDN - The Issues of Today's Internet

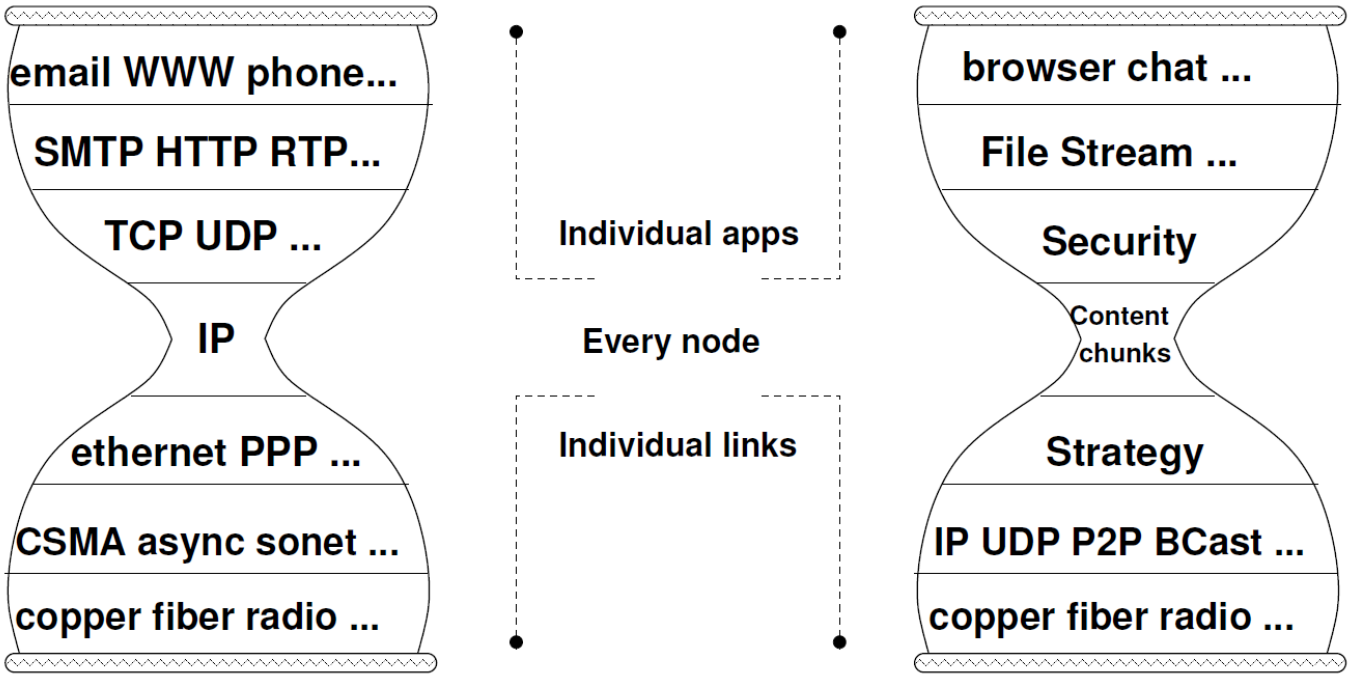
- **Availability**: Fast, reliable content access requires awkward, pre-planned, application-specific mechanisms like CDNs and P2P networks, and/or imposes excessive bandwidth costs.

- **Security**: Trust in content is easily misplaced, relying on untrustworthy location and connection information.

- **Location-dependence**: Mapping content to host locations complicates configuration as well as implementation of network services.

NDN - Solution to The Issues

- To replace *where* with *what*.
- To focus *named data* rather than *named hosts*.



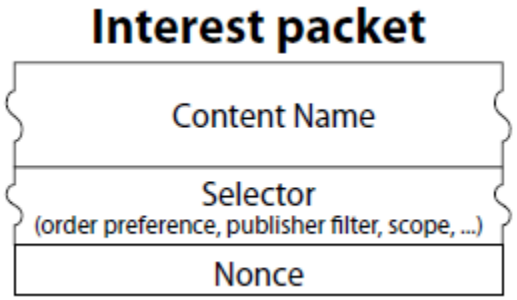
NDN moves the universal component of the network stack from IP to chunks of named content

NDN - Advantages

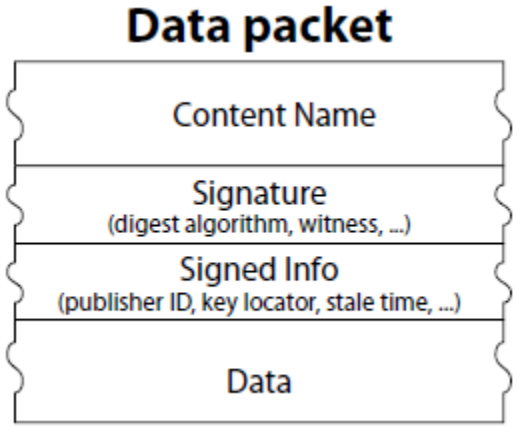
- Strategy
  - NDN can take maximum advantage of multiple simultaneous connectivities.
  - The strategy makes the fine-grained, dynamic optimization choices needed to best exploit multiple connectivities under changing conditions.
- Security
  - NDN secures content itself (Section 5), rather than the connections over which it travels, thereby avoiding many of the host-based vulnerabilities that plague IP networking.

NDN - Packet Types

- Interest



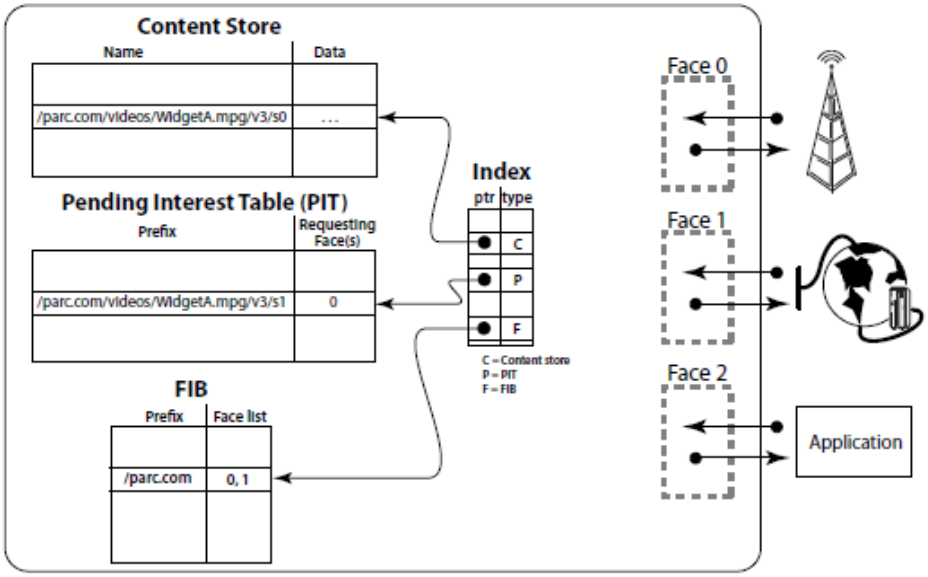
- Data



1. A consumer asks for content by broadcasting its interest over all available connectivity.
2. Any node hearing the interest and having data that satisfies it can respond with a Data packet.
3. Data is transmitted only in response to an Interest and consumes that Interest.

NDN - Forwarding Engine Model

A packet arrives on a face, a longest-match look-up is done on its name, and then an action is performed based on the result of that lookup



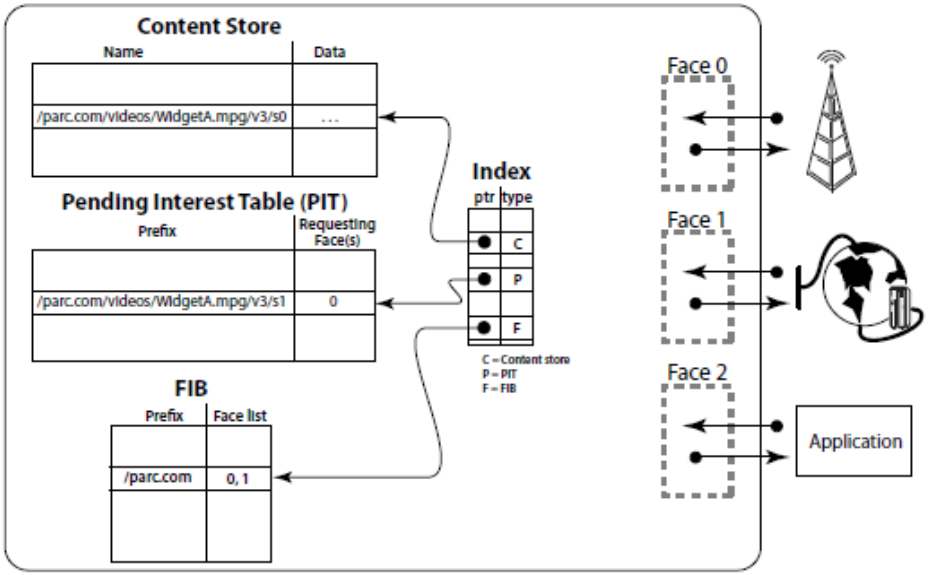
NDN forwarding engine model

FIB

- Be similar with IP FIB
- Forward Interest packets toward potential source(s) of matching Data
- A list of outgoing faces

NDN route Lookup is more efficient than IP, because NDN names used for lookup is structural.

NDN - Forwarding Engine Model



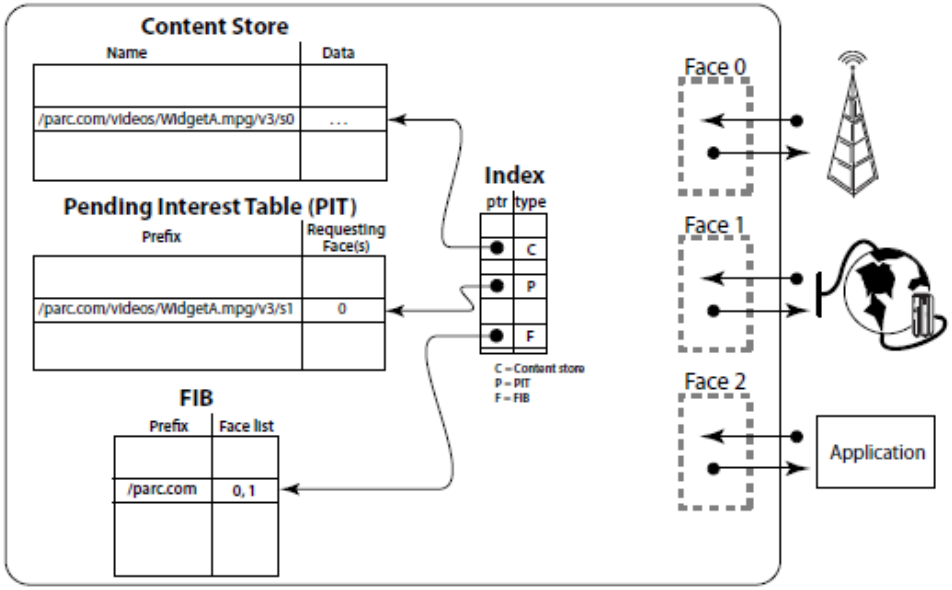
NDN forwarding engine model

Content Store

- Like the buffer memory of an IP router
- Remember arriving Data packets as long as possible

Because NDN is Content-based,
it can deal with caching contents better

NDN - Forwarding Engine Model



NDN forwarding engine model

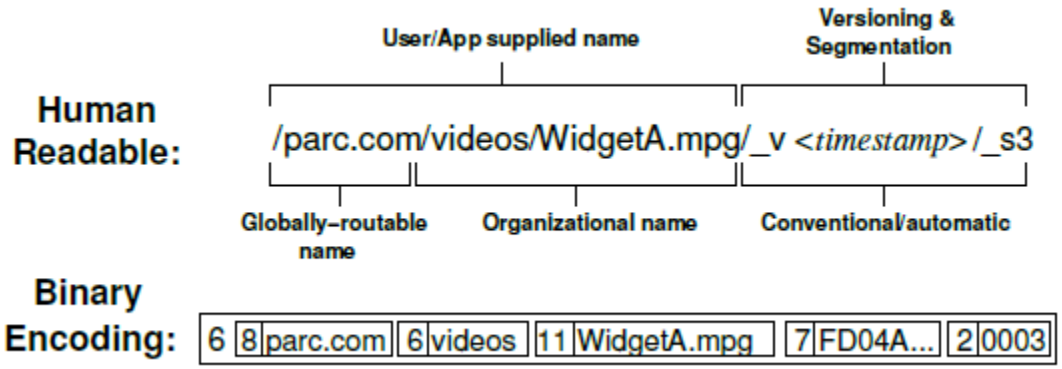
PIT

•keeps track of Interests forwarded upstream toward content source(s) so that returned Data can be sent downstream to its requester(s)

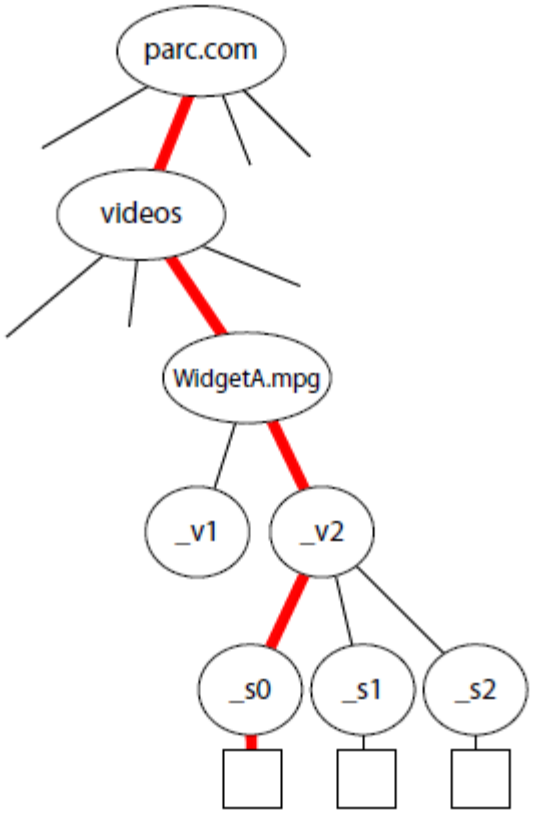
NDN - When a Interest Arrives

- There is already a Data packet in the ContentStore
  - it will be sent out the face the Interest arrived on
  - the Interest will be discarded
- Otherwise, if there is an exact-match PIT entry
  - the Interest's arrival face will be added to the PIT entry's RequestingFaces list
  - the Interest will be discarded
- Otherwise, if there is a matching FIB entry
  - the Interest needs to be sent upstream towards the data.
  - The arrival face is removed from the face list of the FIB entry
  - if the resulting list is not empty, the Interest is sent out all the faces that remain
  - a new PIT entry is created from the Interest and its arrival face
- If there is no match for the Interest
  - it is discarded

NDN - Sequencing



Example Data name



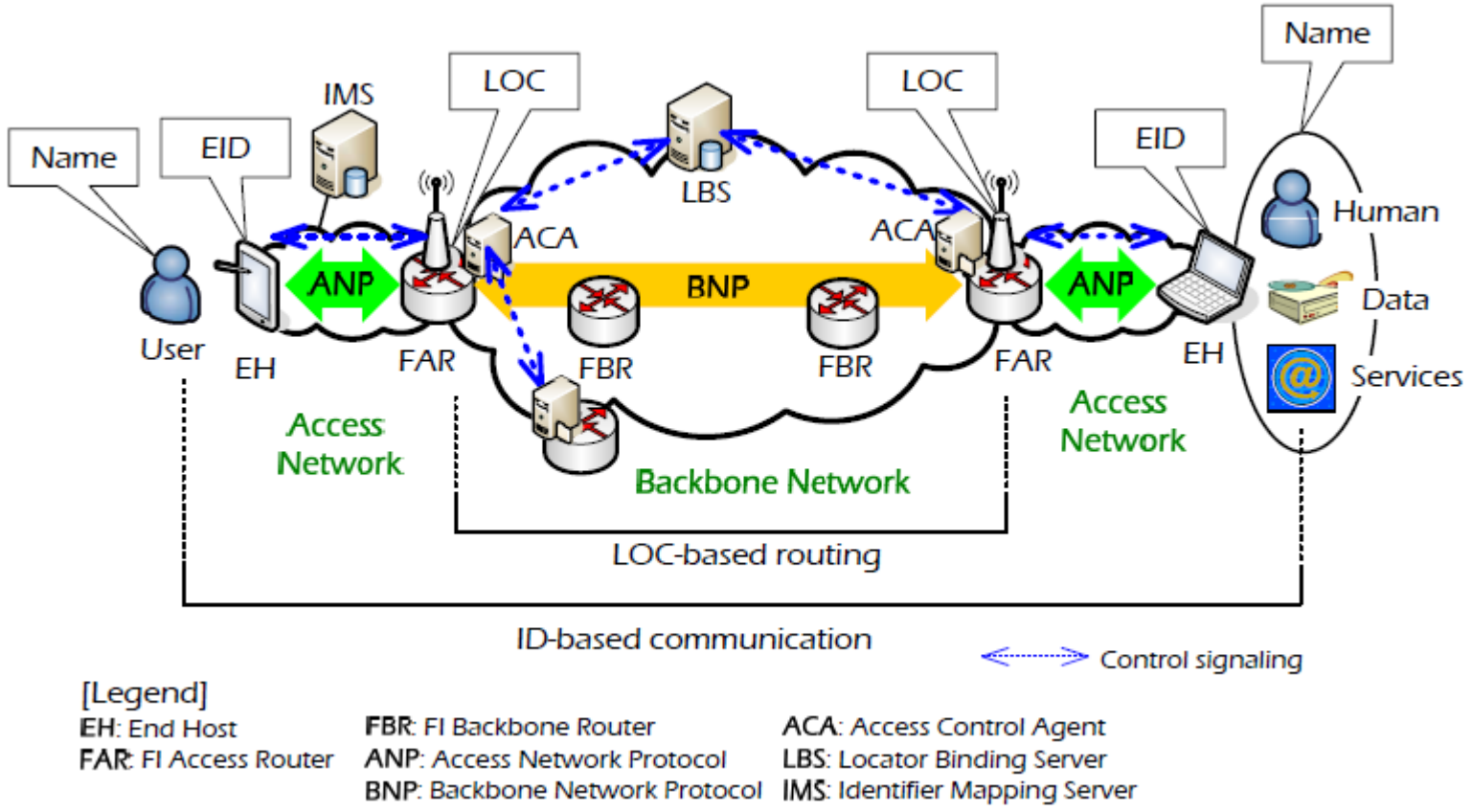
Name tree traversal

ID designs in Mobile Internet

- Mobile-Oriented Future Internet (MOFI)
  - A new data delivery architecture for mobile environment
- MobilityFirst
  - A robust and trustworthy mobility-centric architecture for the future Internet
  - NSF FIA Project

MOFI – Overall Structure

- Three step ID structure
  - Name, EID and Locator

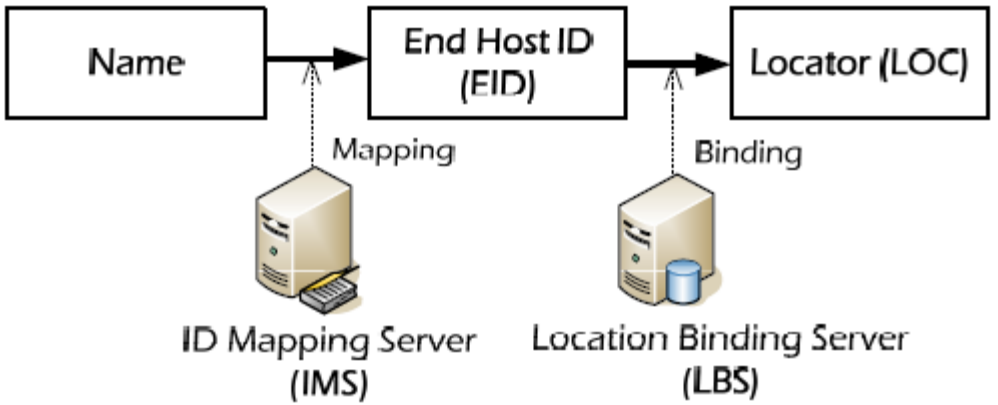


MOFI – IDs in MOFI

- Name
  - Used by human to uniquely identify a corresponding (communicating) object in the network
  - An object may be human, device, data, service, etc.
  - For human's understanding, Name should be human-readable, i. e., alphanumeric.
  - E.g., URI/L, email address, NAI
- End Host ID (EID)
  - A human, service or data identified by name will be served on an end host
  - For delivery of data, the end host should be identified by EID in a static and secure manner, because each ID is used for communication and may be revealed to an unknown user
  - E.g., HID

MOFI – IDs in MOFI

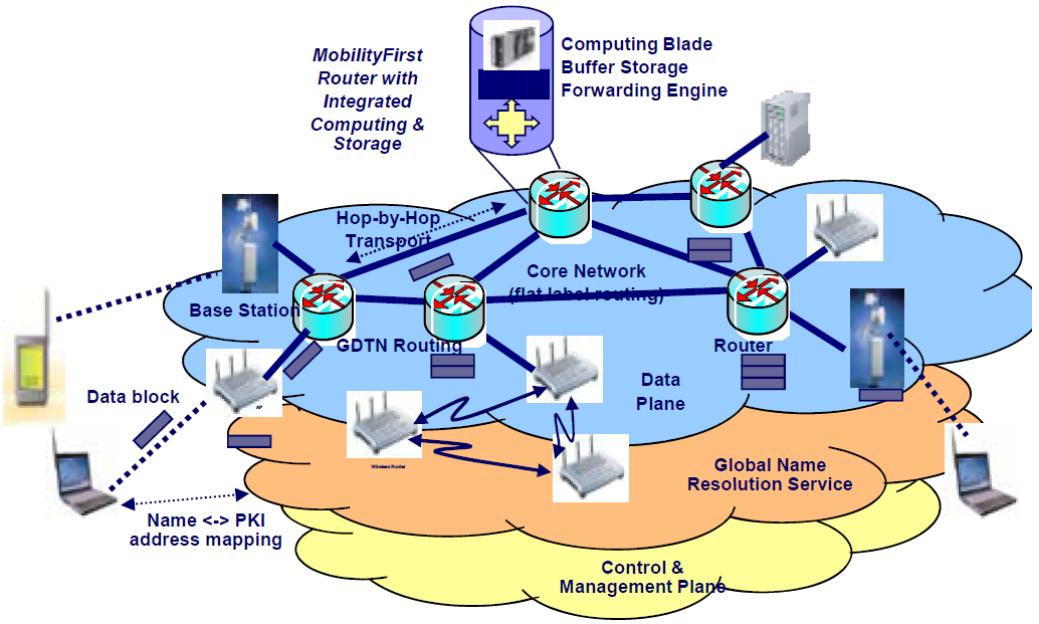
- Locator (LOC)
  - Used to represent the location of an object in the network
  - An LOC may contain the information about topological or geographical location of the user in the network
  - LOC is also used for delivery of data packets between objects in the network
  - E.g., IP address
- Two separations using IMS and LBS
  - Name from EID
  - EID from LOC



- Assume that NAI type Name, Fixed length cryptographic EID, and re-use of IP address as LOC (tentatively)

MobilityFirst - Network Overview

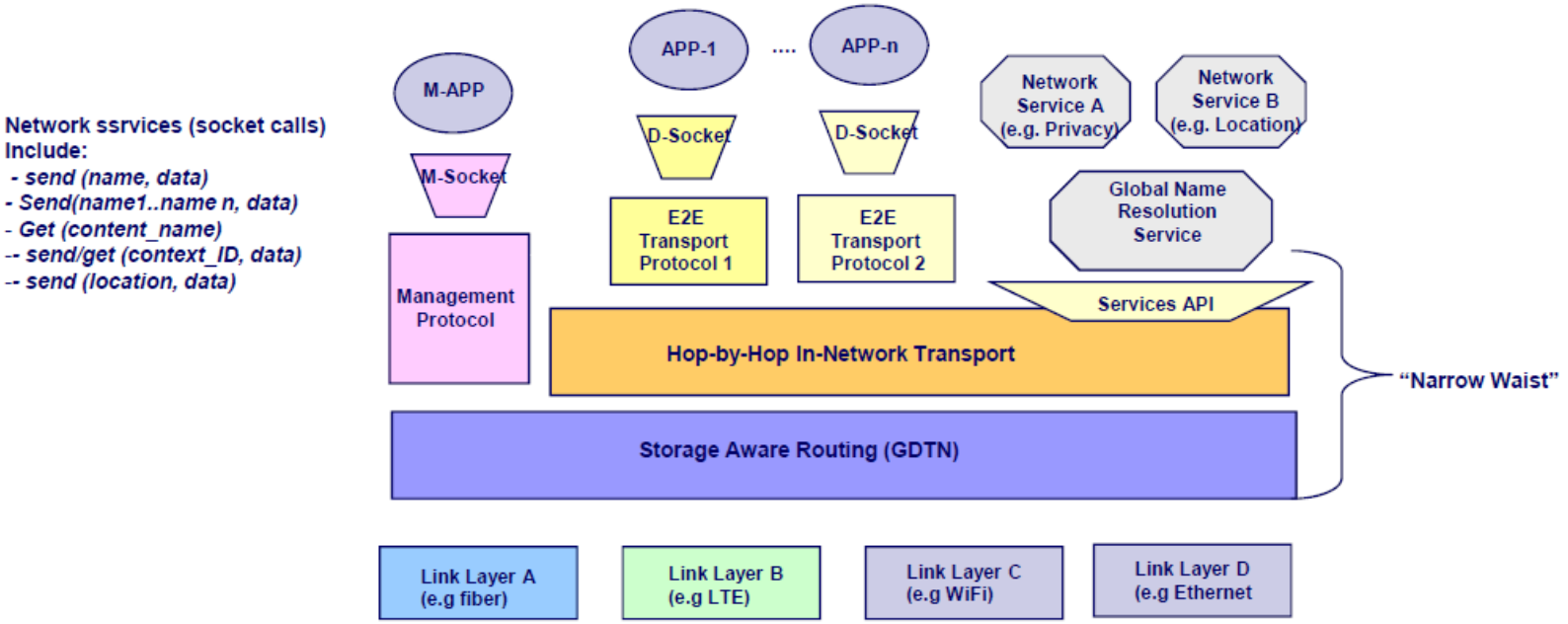
- MobilityFirst key features
  - Fast global naming service
  - Self-certifying public key
  - Flat label addressing in core
  - Storage-aware (generalized DTN) routing in access
  - Hop-by-hop (segmented) transport
  - Programmable computing layer
  - Support for content/context/location
  - Separate network mgmt plane



- New components, very distinct from IP, intended to achieve key mobility and trust goals

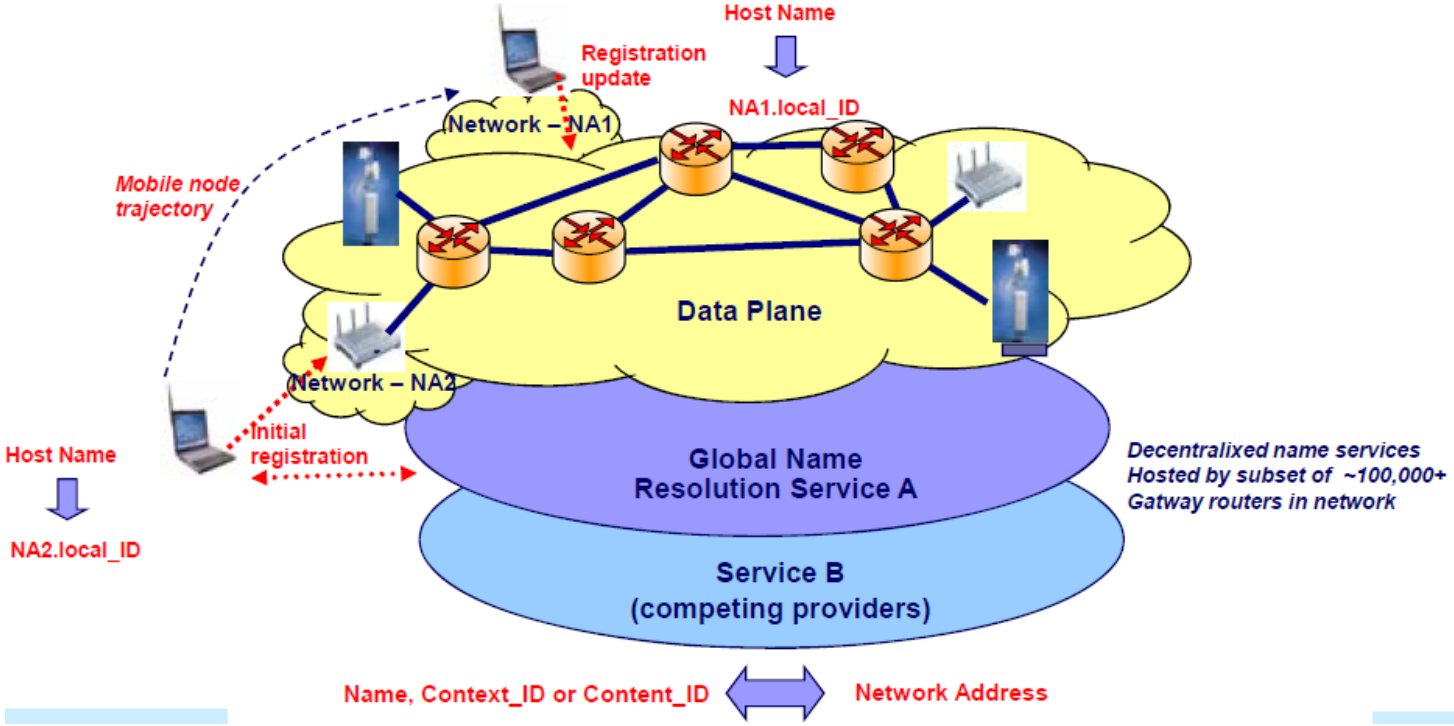
MobilityFirst - Protocol Stack

- Core elements of protocol stack (“narrow waist”)
  - Global Name Resolution Service
  - Storage-aware routing (GDTN)
  - Hop-by-hop (segmented) transport
  - Services and management API’s
- Multiple TP and link layer options + programmable services



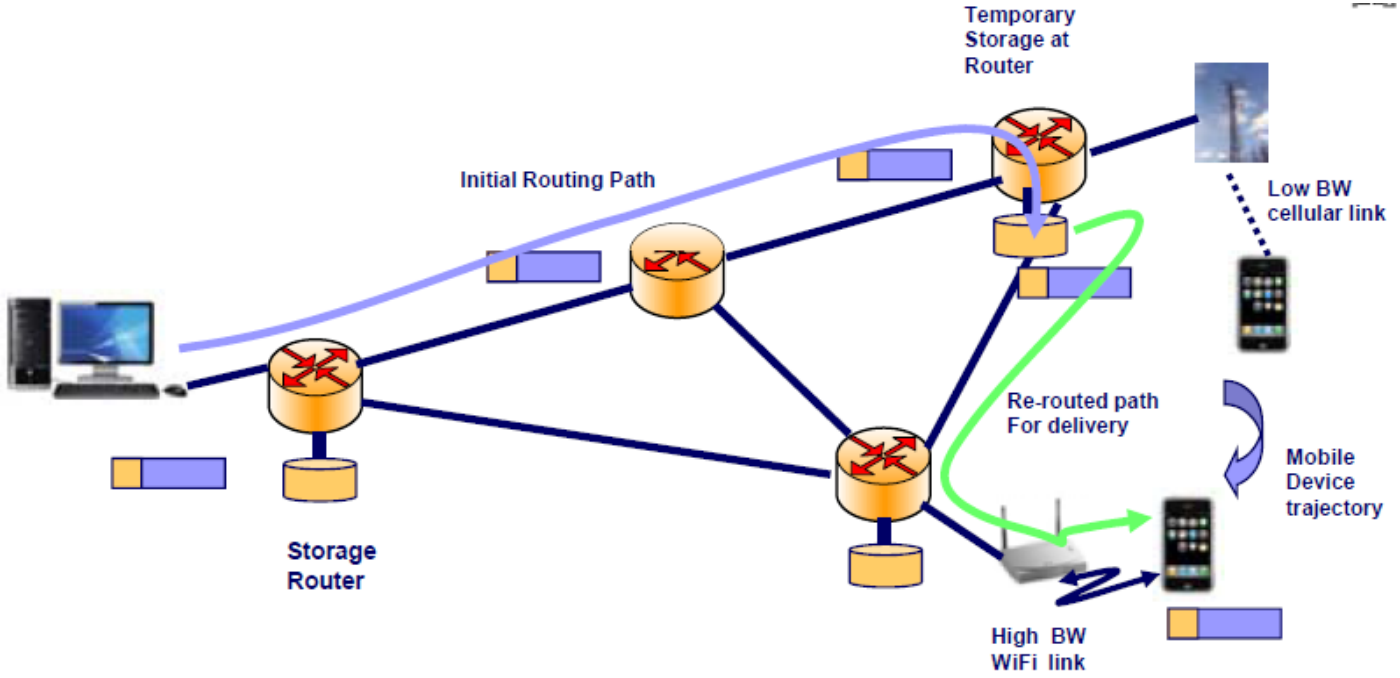
MobilityFirst – Global Name Resolution Service

- Fast Global Name Resolution a central feature of architecture
- Distributed service hosted by routers
  - Multiple competing providers to avoid single root of trust
  - Fast updates ~50-100 ms to support dynamic mobility (..home agent optional)
  - Service will scale to ~10B names via P2P/DHT techniques, Moore's law



MobilityFirst – Storage Aware Routing

- Storage aware (generalized DTN) routing exploits in-network storage to deal with varying link quality and disconnection
- Routing algorithm adapts seamlessly adapts from switching (good path) to store-and-forward (poor link BW/disconnected)
- Storage has benefits for wired networks as well..



Some proposed Solutions in China

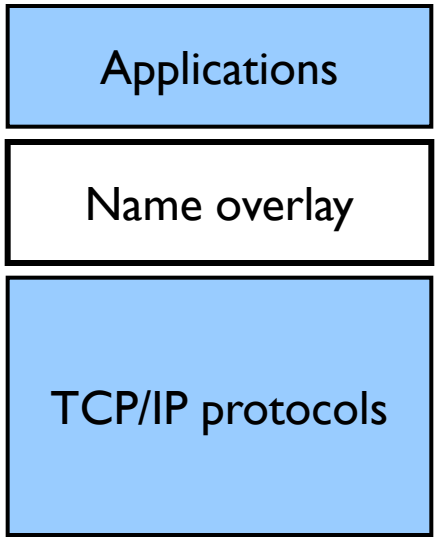
Some proposed Solutions in China

- Tsinghua University
  - Network Research Center
  - Computer Science Dept.
- Beijing JiaoTong University

NOL (Name overlay) Features

- Incremental design
- Host ID：overlay layer on top of TCP/IP stack (no change to TCP/IP socket stack)
  - Borrow Domain Name as ID
- Core/edge split by eliminate the PI prefix announcement to the core network by NTR (Name transfer Relay)
  - Like a NAT extension

Architecture



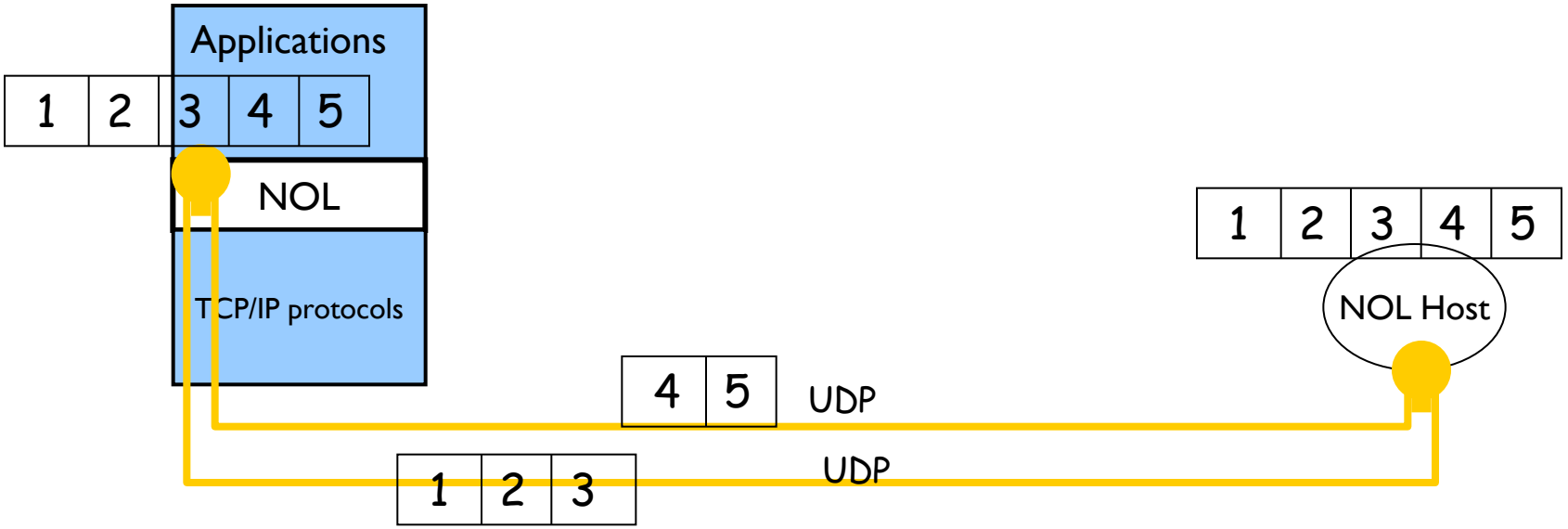
- An Name-Overlay Layer on top of TCP/IP stack (may call it a “session layer)”
- Host name as ID (topology-independent)
- IP address as locator
- Overlay layer functions:
  - host name configuration, register and authentication (security mechanisms…)
  - help to establish transport connection channels by names (IDs)
  - create application sessions on top of NOL

Name Transfer Relay (NTR)

- A device to cooperate with NOL layer of the host
  - To create address mapping based on name
  - To help host behind NTR to receive incoming connections from outside
- Deployed at access point of edge networks
- Can also split the core/edge by eliminating the PI prefix announcement

Multihoming, Multi-path, Mobility…

- An end host can be assigned multi-homed addresses; multipath by using multiple NTRs
- A host name is used as ID in a application session by NOL
  - keep application continuity for mobility



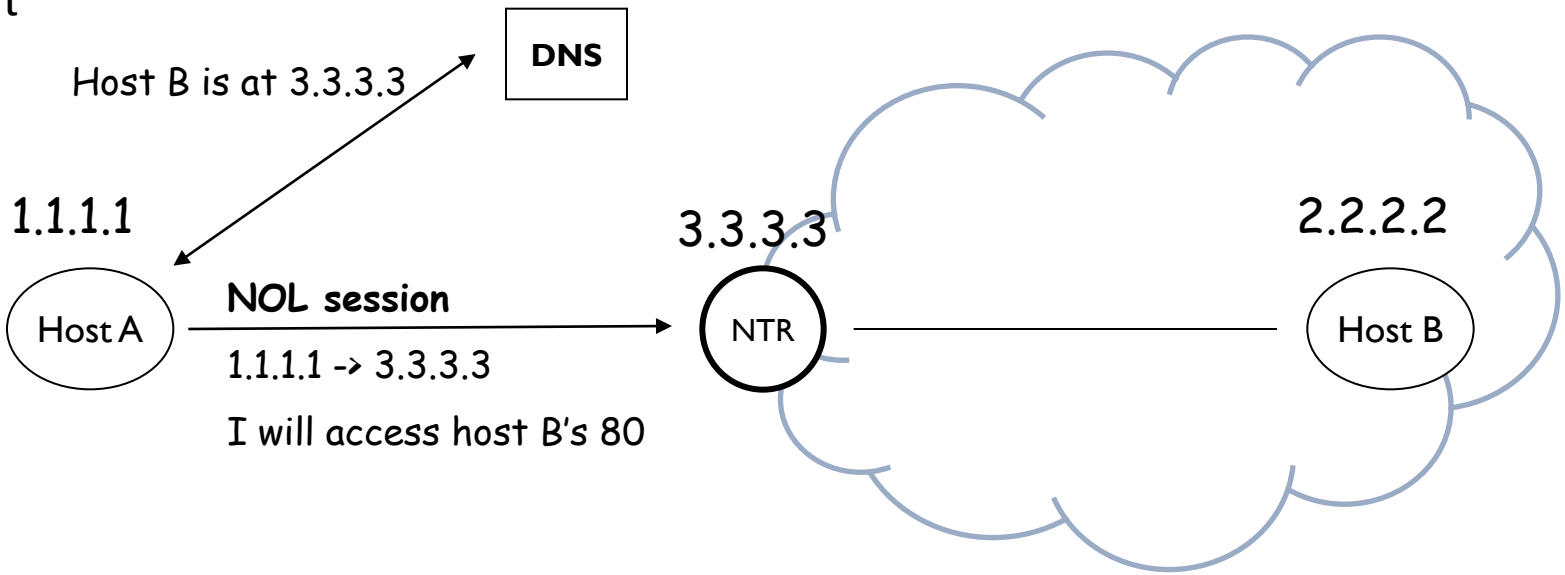
Initiate access to NTR hosts (by name)

First, Host B obtained a name from NTR (e.g., B@network.com) which is also known by NTR

Name of NTR hosts should be limited in a domain hierarchy, fox example, email address "**host@domain**" We just query DNS for the "domain" that will not increase the load of DNS.

Second, host A query DNS for host B, and the returned IP is NTR 3.3.3.3, (configured in DNS)

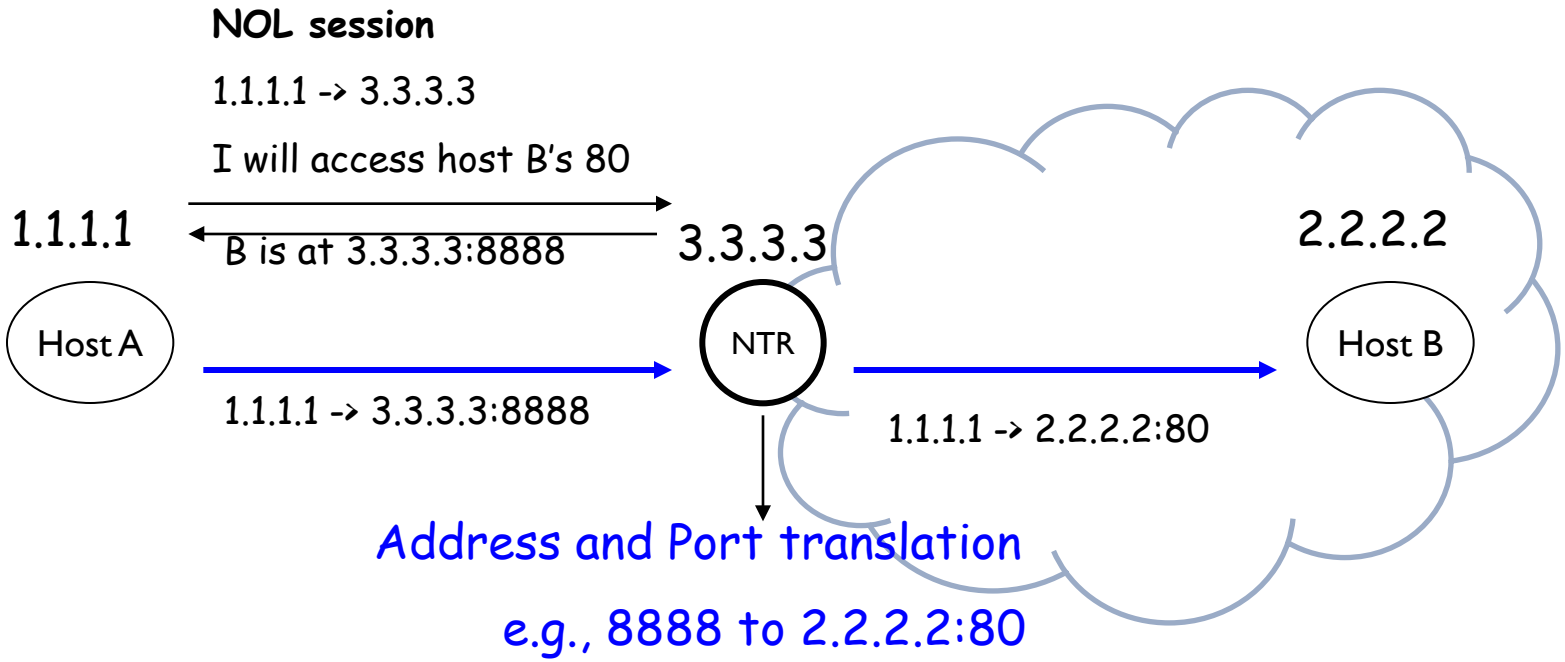
And then, host A's NOL will initiate NOL session to NTR, with B's name and port



Then, NTR look up B's name, and knows that B's IP is 2.2.2.2

We assume NTR has only one IP address, (here, 3.3.3.3), it will create a port to 2.2.2.2 mapping entry (e.g., port 8888 -> 2.2.2.2:80), and return the port (e.g., 8888) to A, NOL session is successful. (If NTR has many addresses, create address-to-address translating entry, or other type of translation)

Finally, host A initiate packet to 3.3.3.3:8888, and NTR translate it to 2.2.2.2:80. Initiating access from A to B succeed :)



Address multiplexing by name and Port translation

Summary

- Pros
  - No need to change TCP/IP stack (sockets or OS kernel) of NOL host, No need to change DNS core protocol (just change the content)
  - NOL applications can communicate with legacy applications
  - Core/edge separation to reduce PI prefix announcement
  - Multihoming, Multi-path, mobility…
- Cons
  - Translating cost at NTR

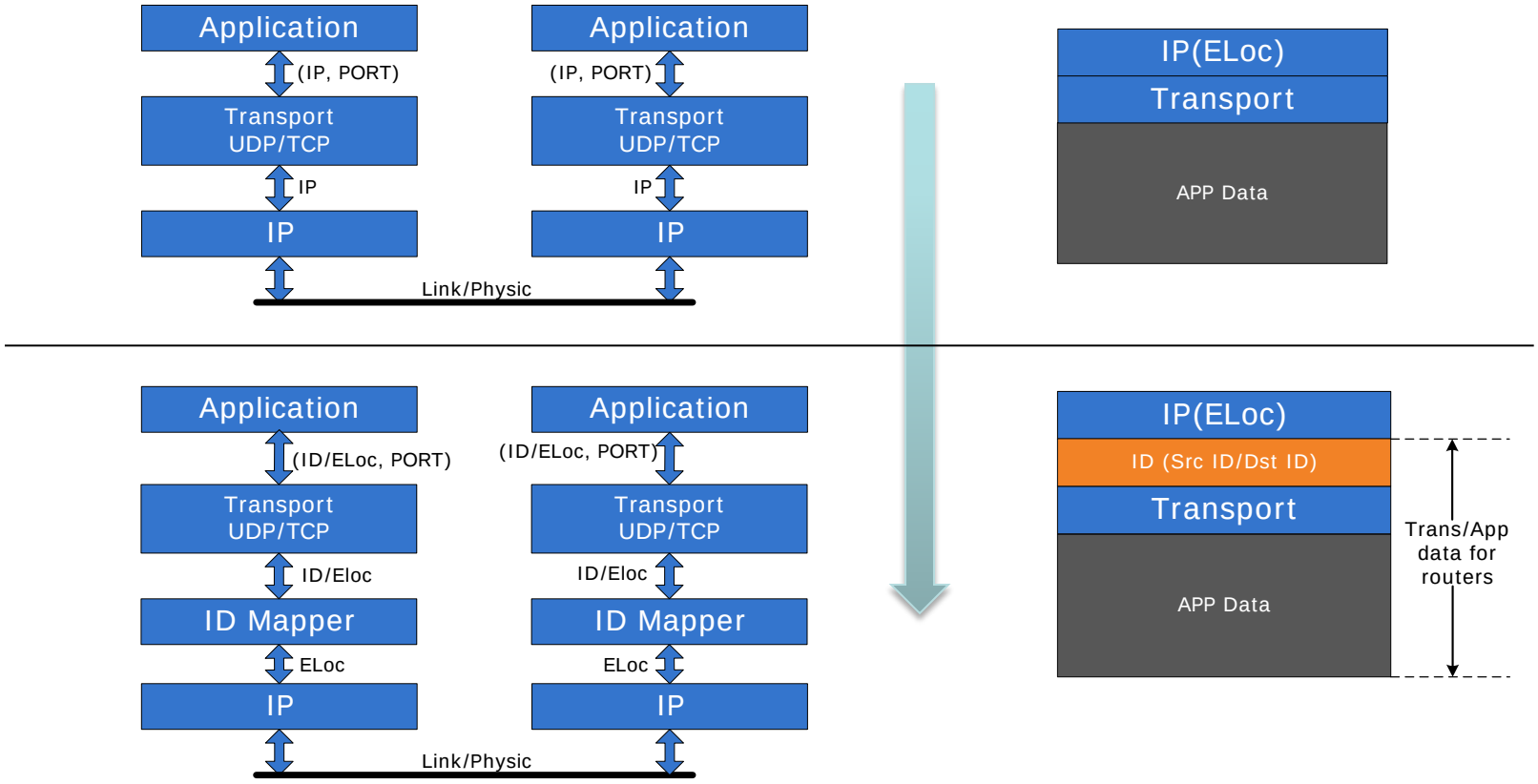
Some proposed Solutions in China

- Tsinghua University
  - Network Research Center
  - Computer Science Dept.
- Beijing Jiao Tong University

Features

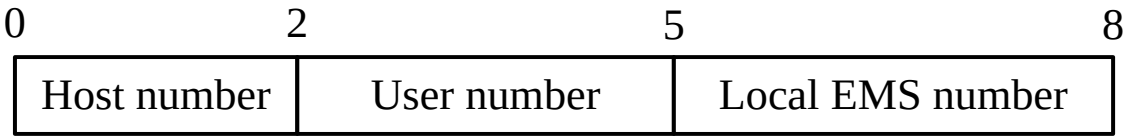
- Host+user ID：new layer in the stack
  - Structured ID
- Core/edge split
  - ELoc and Rloc
- Terms
  - ID, ELoc, RLoc
  - EMS: to map ID/Eloc (locator in edge)
  - RMS: to map Eloc/Rloc (locator in core)

ID Layer and New Socket APIs

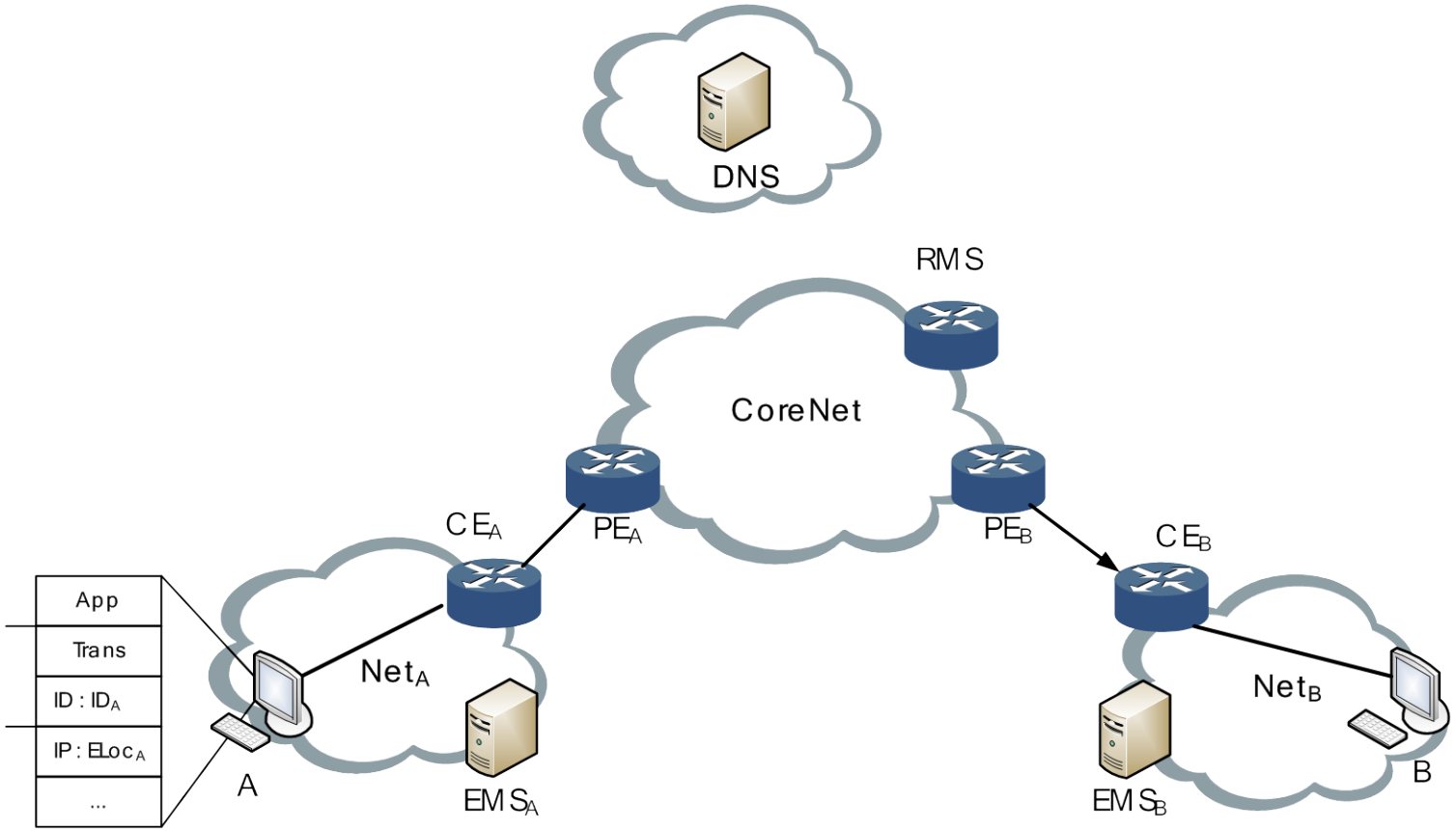


Identifier Design

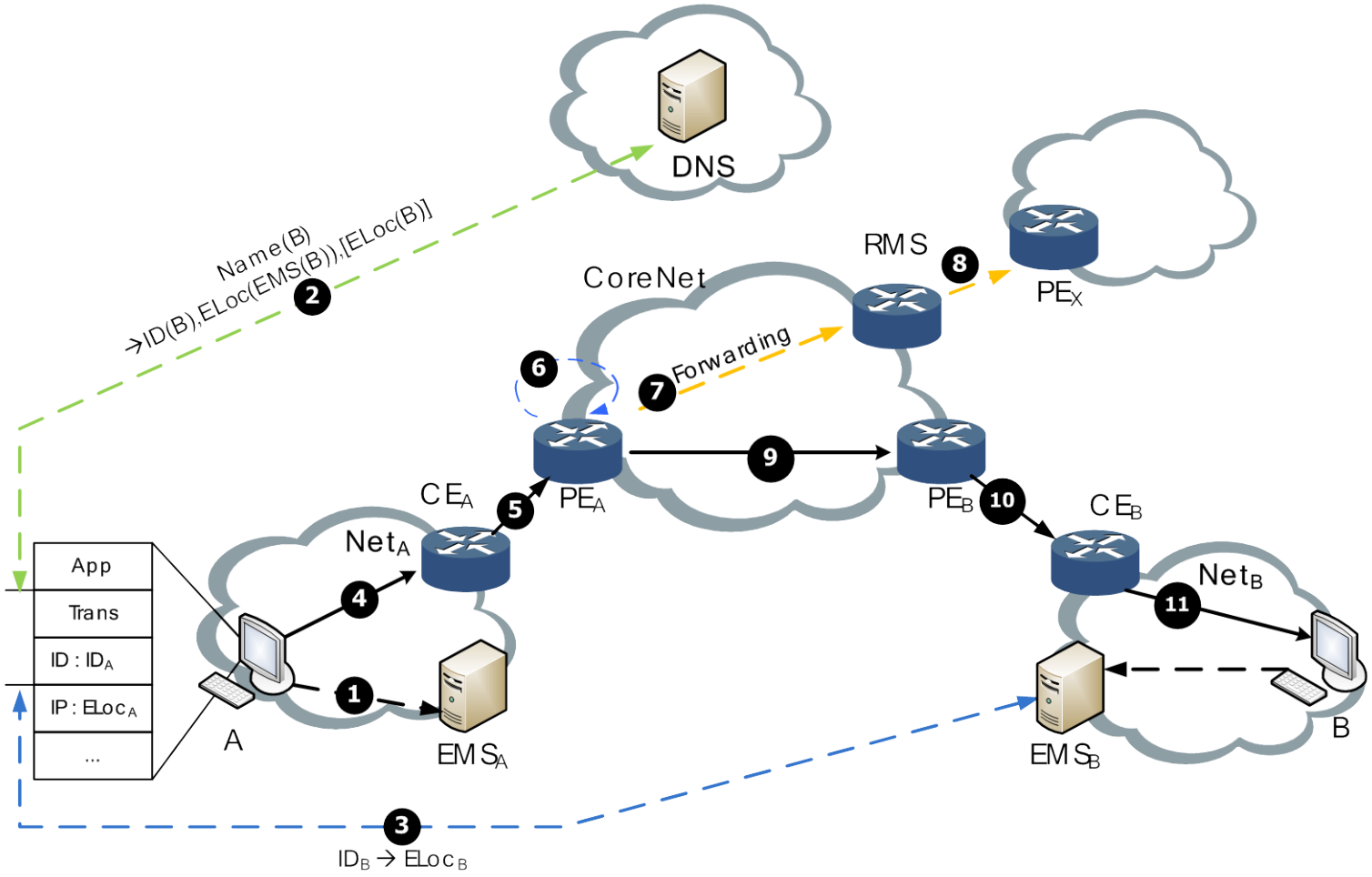
- User ID
  - Identify a user, also for authentication
  - name@organization
  - xiaoming.zhang@tsinghua.edu.cn
- Host ID
  - Identify a user's host, used in transport
  - A user can have many hosts, including a default one
  - number#name@organization
  - 3#xiaoming.zhang@tsinghua.edu.cn
  - Can be encoded into 8 octets



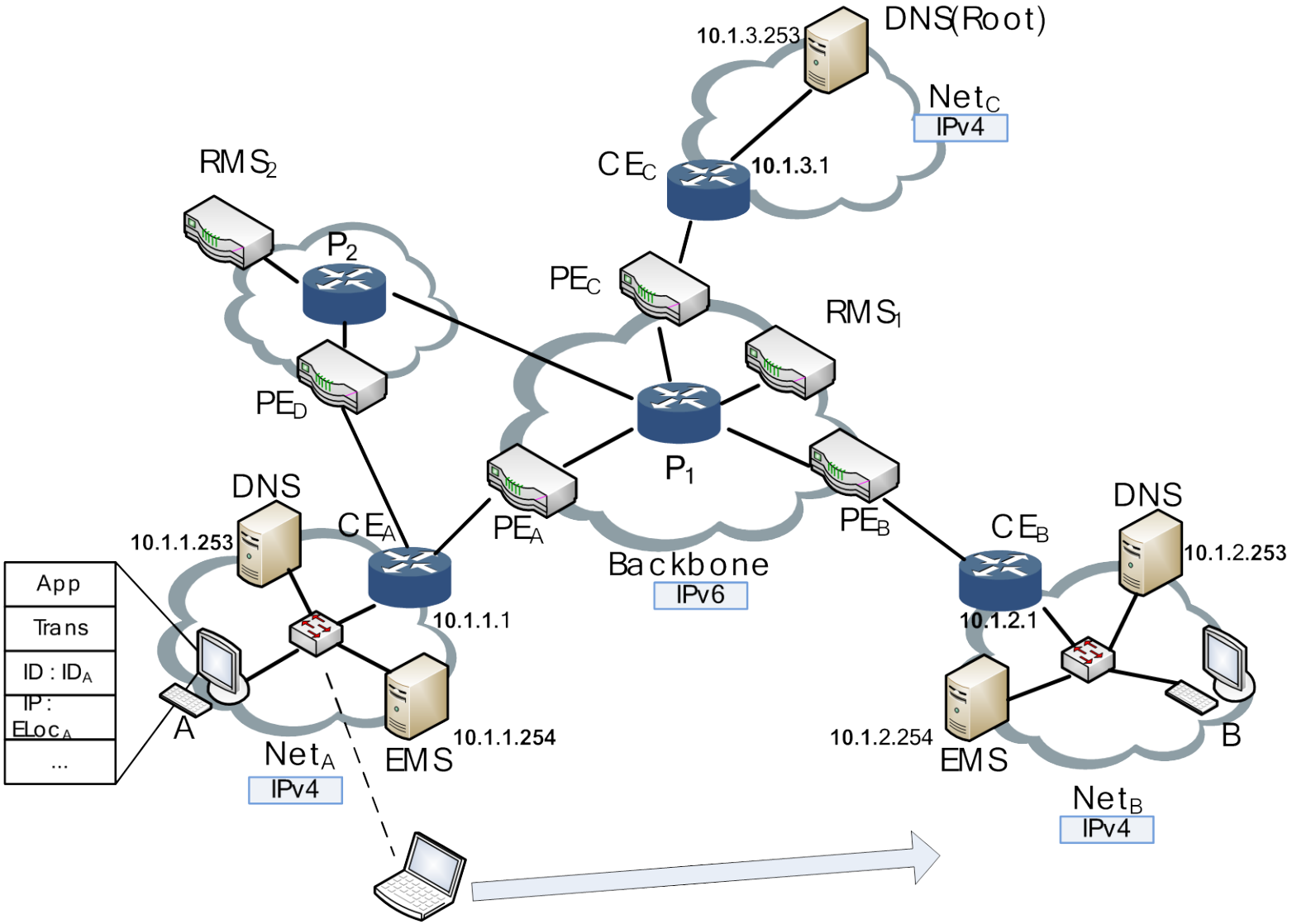
Components


DNS
RMS
CoreNet
CE_A
PE_A
PE_B
CE_B
App
Trans
ID : ID_A
IP : ELoc_A
...
A
Net_A
EMS_A
Net_B
EMS_B
B

Procedure


DNS
Name(B)
→ID(B),ELoc(EMS(B)),[ELoc(B)]
2
CoreNet
RMS
8
PEX
6
7
Forwarding
9
CEA
5
PEA
PEB
10
CEB
NetA
4
1
App
Trans
ID : IDA
IP : ELocA
...
A
EMSA
NetB
11
EMSB
B
3
IDB → ELocB

A Prototype


DNS(Root)
10.1.3.253
Net_C
IPv4
CE_C
10.1.3.1
RMS_2
P_2
PE_C
RMS_1
PE_D
P_1
DNS
10.1.1.253
CE_A
PE_A
PE_B
CE_B
DNS
10.1.2.253
Backbone
IPv6
10.1.1.1
10.1.2.1
App
Trans
ID : ID_A
IP : ELoc_A
...
A
Net_A
IPv4
EMS
10.1.1.254
10.1.2.254
EMS
B
Net_B
IPv4

ID/Locator Separation

- Characters
  - Separation of backbones' addressing space and edges' addressing space
  - Reduce the number of globally announced prefixes
  - Extended sockets are introduced to execute the new function for the new Identifier

Some proposed Solutions in China

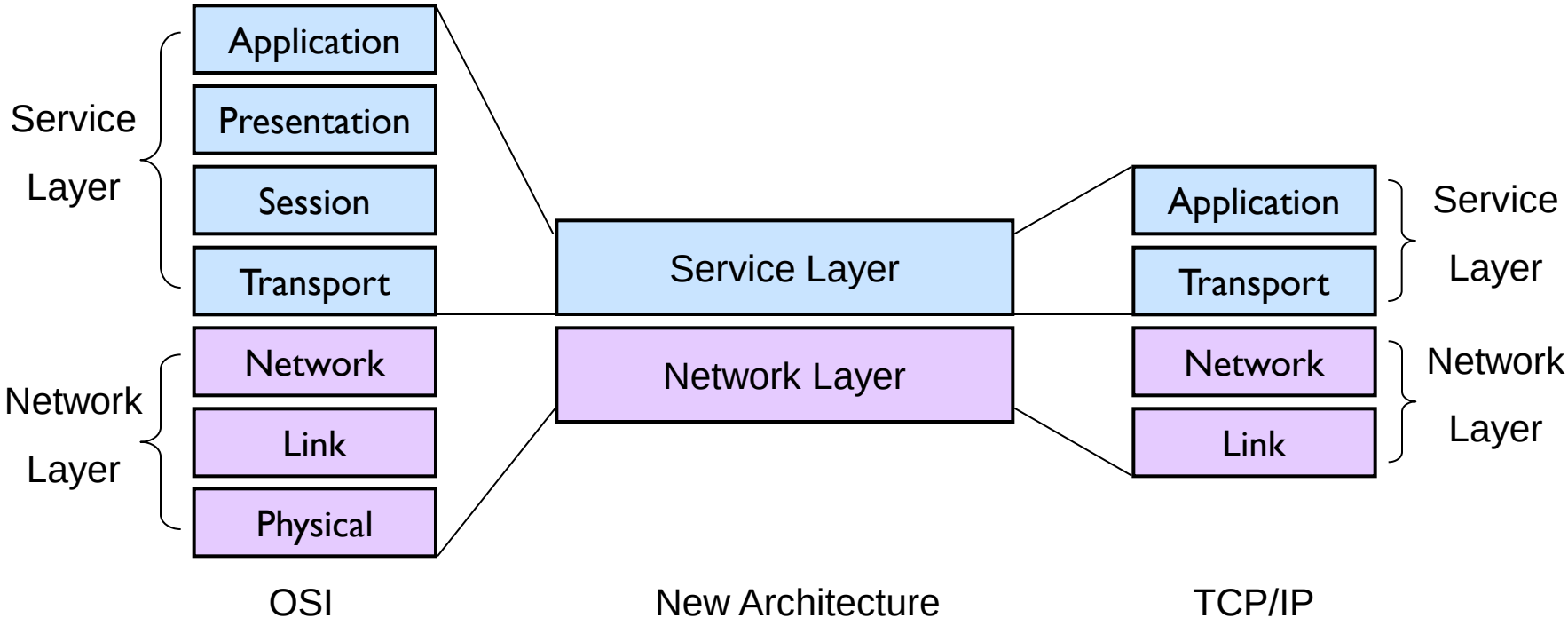
- Tsinghua University
  - Network Research Center
  - Computer Science Dept.
- Beijing JiaoTong University

Features

- Clean-slate design
- ID：hierarchical design
  - Service ID
  - Connection ID
  - Access ID
  - Routing ID
- Core/edge split
  - Access ID and Routing ID

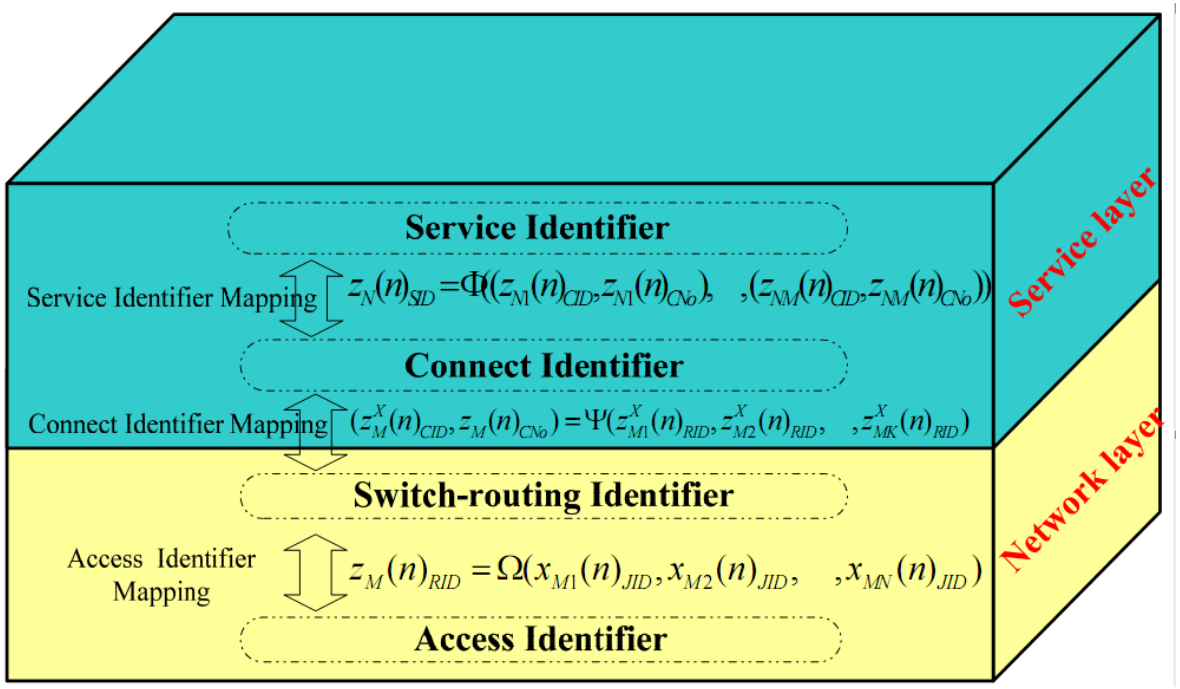
Architecture

- Two Basic Layers
  - Service Layer
  - Network Layer

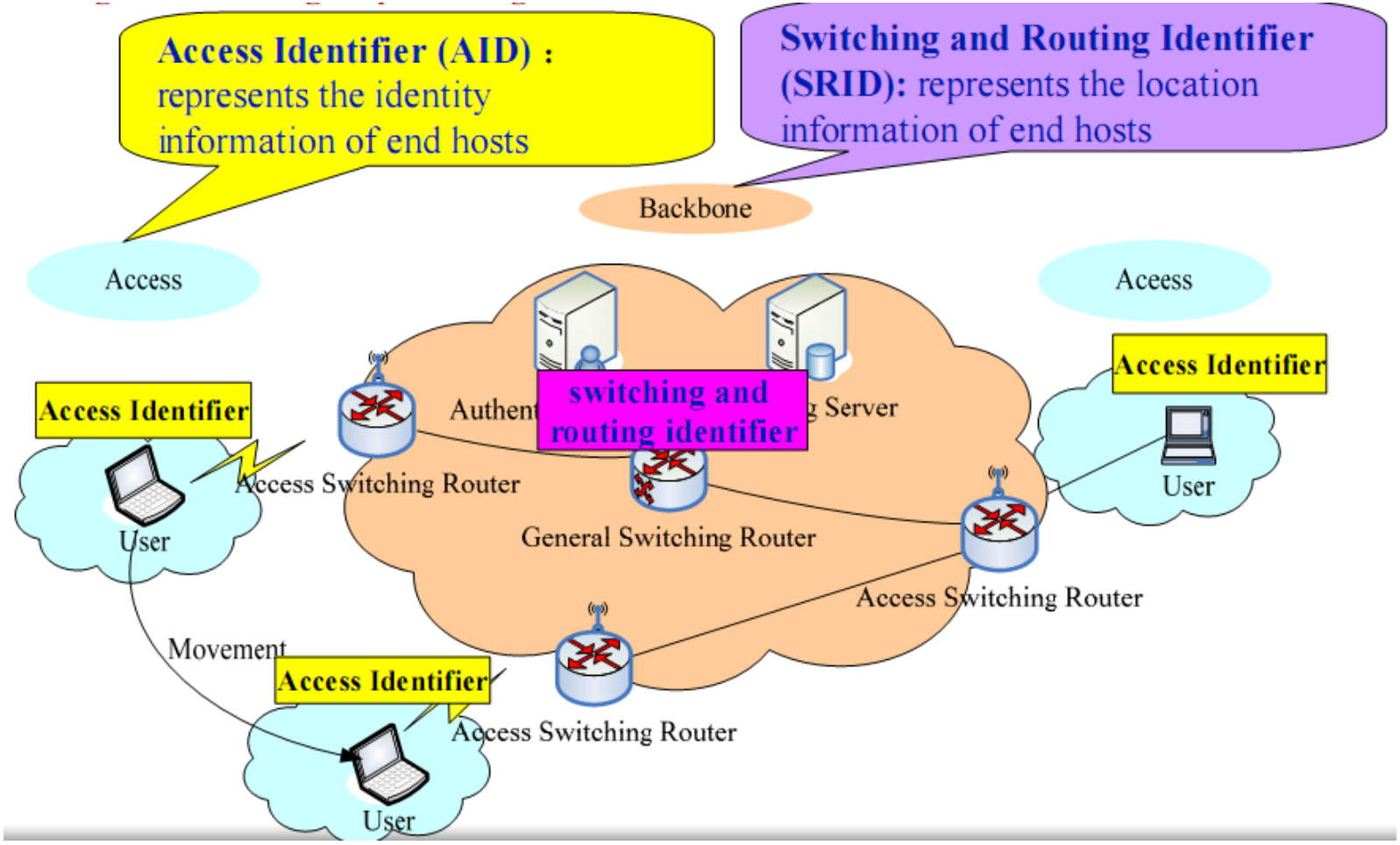


Architecture (cont.)

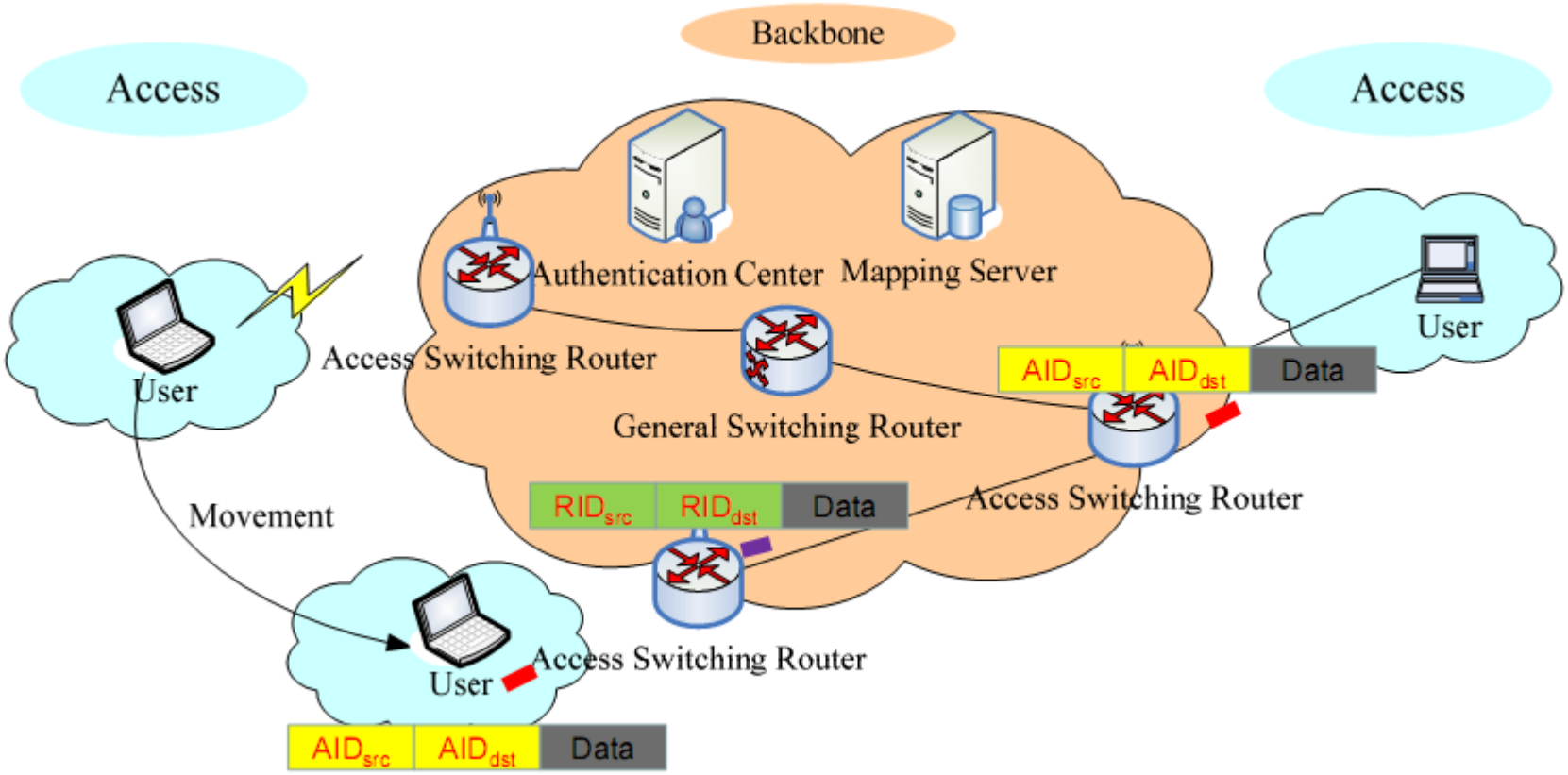
- Service Layer
  - Service ID
  - Connection ID
- Network Layer
  - Routing ID
  - Access ID



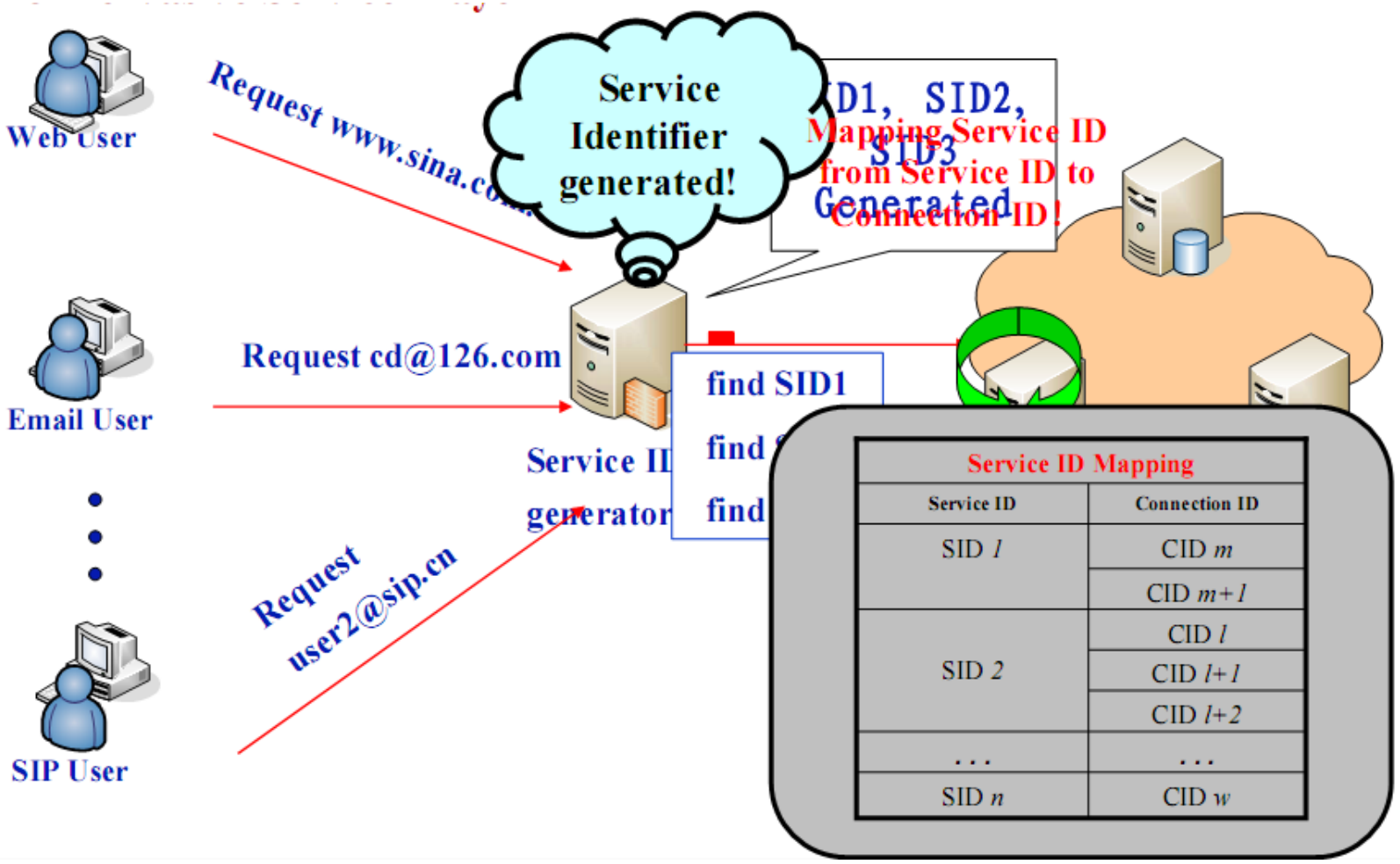
Network Layer



Network Layer (cont.)

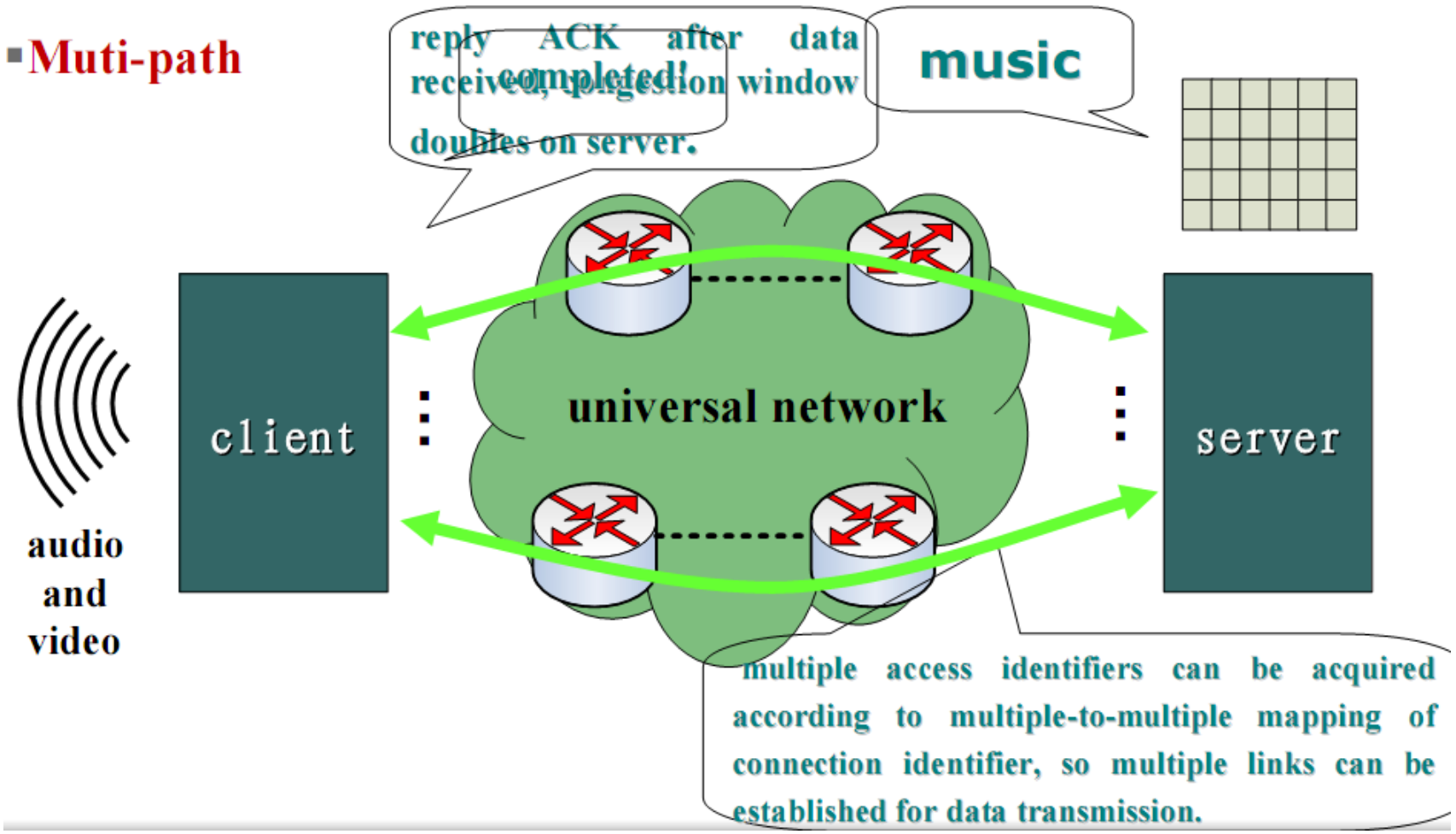


Service Layer



Multi-path and Multiplexing

- Muti-path



Summary

- Pros
  - Integrated design for future information network (new IDs from network layer to application)
  - Multi-path transmission
  - Host ID for mobility and authentication
- Cons
  - Long-term solution and it’s hard to be incrementally deployed from the current Internet (Clean Slate)

ID Design suggestions

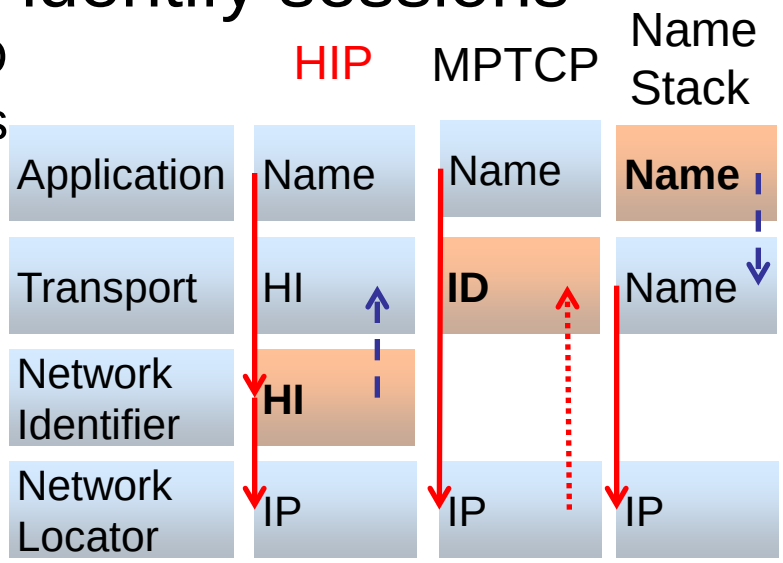
- Layering of ID
- Application solution
- Incremental deployment

Design Suggestions – layering of ID

- Layering in the name space was not enforced from the beginning of the Internet
  - Network layer and transport layer are tightly coupled
  - It has brought problems when new requirements such as mobility and multi-homing appear
- Adopt modularity in designing new name spaces
  - Names in adjacent layers should be completely independent

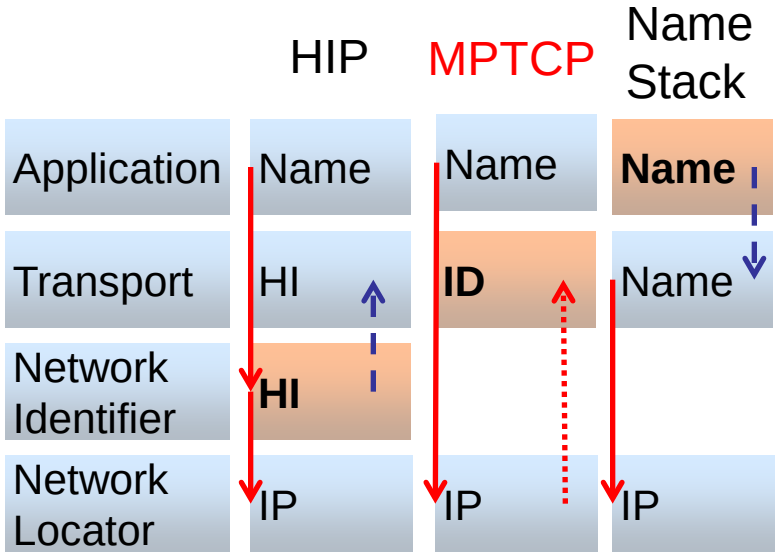
Design Suggestions – layering of ID

- ID in the network layer (host or node ID)
  - HIP, RANGI, LISP, lvip, RANGER, ROFL, …
  - Used by upper layers when setting up connections
  - Benefits the E2E authentication mechanism
- Need global mapping service
  - Modify DNS or
  - Add new infrastructures
- Transport layer may use it to identify sessions
  - Tight coupling in session and host ID
  - It is difficult to switch communicaions between hosts
  - Multi-device can not be supported



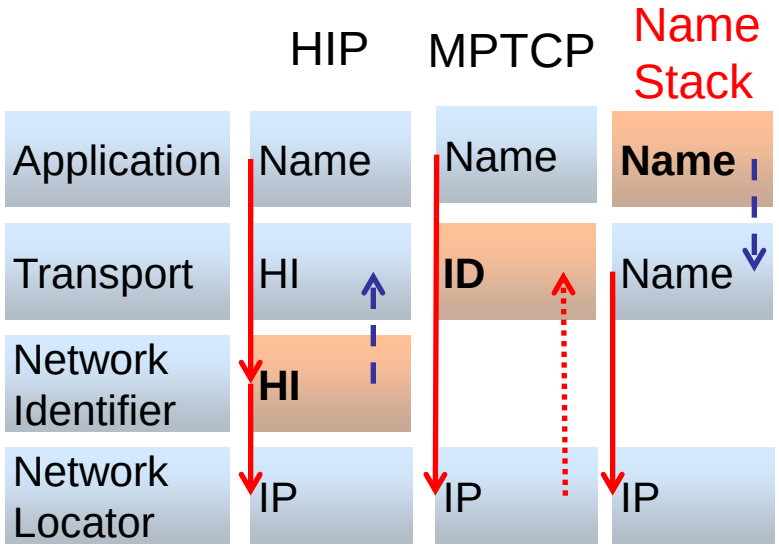
Design Suggestions – layering of ID

- ID in the transport layer (session ID)
  - MPTCP, SCTP, Shim6, …
  - Identify the end-point in E2E communications, local scope
  - Mappings are maintained by both sides, no network-change
- Have problems in mobility
  - Locator-update in a E2E mechanism way
  - Session may break when both sides are on the move



Design Suggestions – layering of ID

- ID in the application layer
  - Name Stack, …
  - Users care about services and contents, not devices or locators
  - Bind sessions directly to service or data ID, not host or node ID
  - Communication sessions can switch between devices
- Mapping mechanism may be a challenge
  - Huge amounts of app-layer IDs
  - Updates will be frequent



Design Suggestions – application solution

- Problems can be handled by app-level solutions
  - Maintain the continuity of data transmission when session breaks
- But it will bring lots of work to apps
  - Repeat the code in every application
  - And no app-solution may unify the market, inter-working between them will be difficult
- Also, applications don’t know much about the lower layers
  - Can’t know which interface is appropriate
  - Can’t tell if things have changed, even if the connection stays alive
  - So they can’t make a good decision

Design Suggestions – incremental deployment

- Some approaches from industry come into use quickly in the Internet
  - MobileMe by Apple, C2DM by Google
  - Not perfect, but they only use existing protocols
- Most solutions in this talk are not widely used
  - They have huge deployment cost (even clean-slate)
- Incremental deployment is important
  - Backwards compatible
  - Easy transition to new protocol

References

1. http://www.ietf.org/mail-archive/web/rrg/current/maillist.html#05058
2. ttp://www.dagstuhl.de/en/program/calendar/semhp/?semnr=09102
3. http://www.asiafi.net/meeting/2010/WIFI/main.htm
4. http://www.itu.int/itu-t/recommendations/index.aspx?ser=Y
5. J. Abley, B. Black, and V. Gill, Goals for IPv6 Site-Multihoming Architectures, IETF RFC 3582, 8/2003
6. R. Stewart, Ed. Stream Control Transmission Protocol, RFC 4960, 2007
7. L. Ong, J. Yoakum, An Introduction to the Stream Control Transmission Protocol (SCTP), RFC 3286, 2002
8. G. Huston,Architectural Commentary on Site Multi-homing using a Level 3 Shim IETF Internet Draft draft-ietf-shim6-arch-00.txt, 7/2005
9. P. Nikander, R. Moskowitz, Host Identity Protocol IETF Internet Draft draft-moskowitz-hip-07.txt, 6/2003
10. http://christianvogt.mailup.net/pub/2009/vogt-2009-name-oriented-sockets.pdf
11. Dino Farinacci, V. Fuller, D. Oran, D. Meyer, Locator/ID Separation Protocol (LISP), Internet draft, draft-farinacci-lisp-03, 2007.
12. R. Atkinson, ILNP Concept of Operations, draft-rja-ilnp-intro-03, 2009
13. TRIAD: http://www-dsg.stanford.edu/triad/
14. P. Francis, Pip Near-term Architecture, RFC1621, 1994

References

15. B. Ahlgren, J. Arkko, L. Eggert, and J. Rajahalme. A node identity internetworking architecture. Infocom 2006.
16. K. Sollins, Architectural Principles of Uniform Resource Name Resolution, RFC2276, 1998
17. A. Walsh, A Uniform Resource Name (URN) Namespace for the Web3D Consortium (Web3D), RFC3541, 2007
18. MobileMe: http://tools.ietf.org/html/draft-zhu-mobileme-doc-01
19. Host Identity Protocol (HIP) Architecture: http://tools.ietf.org/html/rfc4423
20. Matthew Caesar, Tyson Condie, Jayanthkumar Kannan, Karthik Lakshminarayanan, Ion Stoica: ROFL: routing on flat labels. SIGCOMM 2006:363-374
21. Multipath TCP (MPTCP): https://datatracker.ietf.org/doc/draft-ietf-mptcp-architecture/
22. i3 project: Internet Indirection Infrastructure: http://i3.cs.berkeley.edu/
23. DONA: Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, Ion Stoica: A data-oriented (and beyond) network architecture. SIGCOMM 2007:181-192