Aalto University

Customer Edge Switching

Brief overview of ongoing work

Prof. Raimo Kantola

Nicklas Beijar, Jesus Llorente, Petri Leppäaho, Maryam Pahlevan

Agenda

- Background
- Introduction to Customer Edge Swiching (CES)
- Customer Edge Traversal Protocol (CETP)
- Interworking with legacy IP
- Application compatibility
- Application Layer Gateways (ALGs)
- Prototype

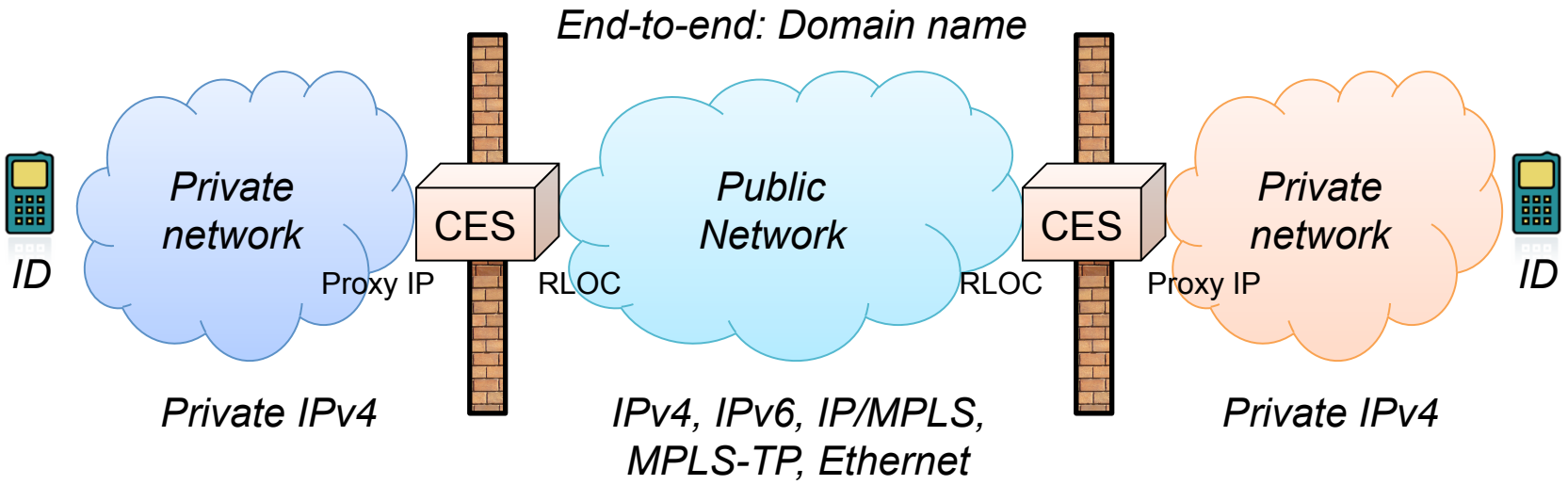
Goals

- Develop the NAT/firewall into something that is part of the architecture, not an exception
  - Enable better cooperation between benevolent users and pruning out the bad guys (hackers, spammers, fraudsters…)
- Avoid NAT traversal mechanisms (STUN, TURN, ICE)
- Enable inbound traffic in a controlled way (based on policies)
- Reduce unwanted traffic
- Separate customer network from public network
- Improve scalability, multihoming
- No changes to hosts, applications, IP stack

CES and SDN

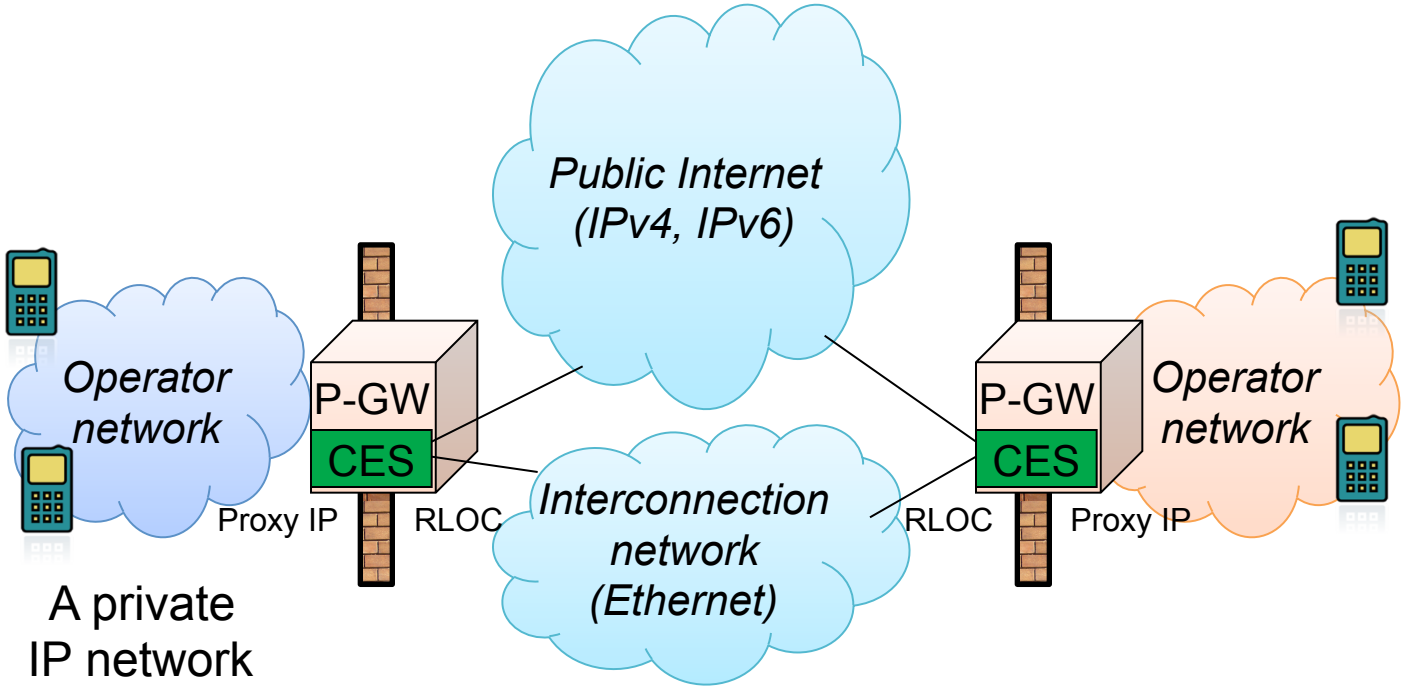
- Firewall as a cloud service?
  - To me as a customer this would seem to make sense
- As a use case
  - Study the limits of scalability of OF++ : break CES CP and DP into different elements and test how that works

Customer Edge Switching



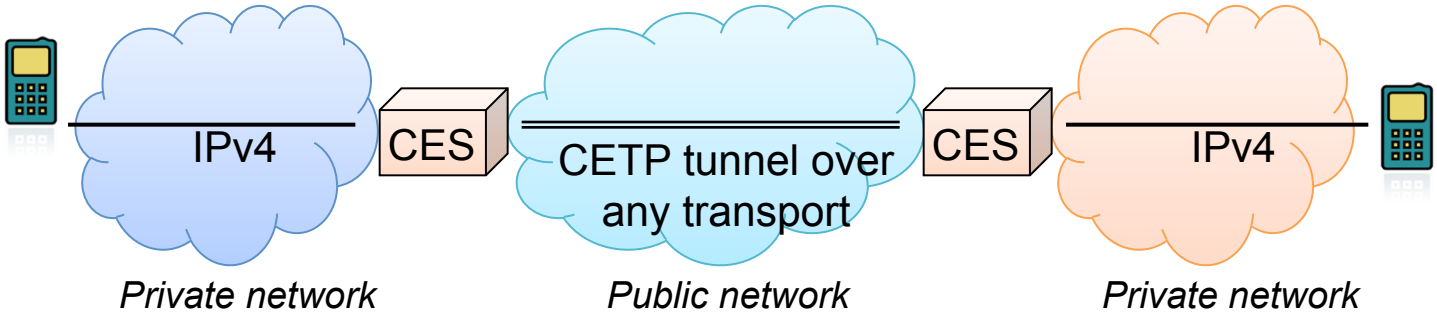
- Separates the customer network from the public network
- Separates the name from the routing address
- Each network can use different routing and transport
- Collaboration between CES devices → trust

Deployment in the System Architecture Evolution (SAE)

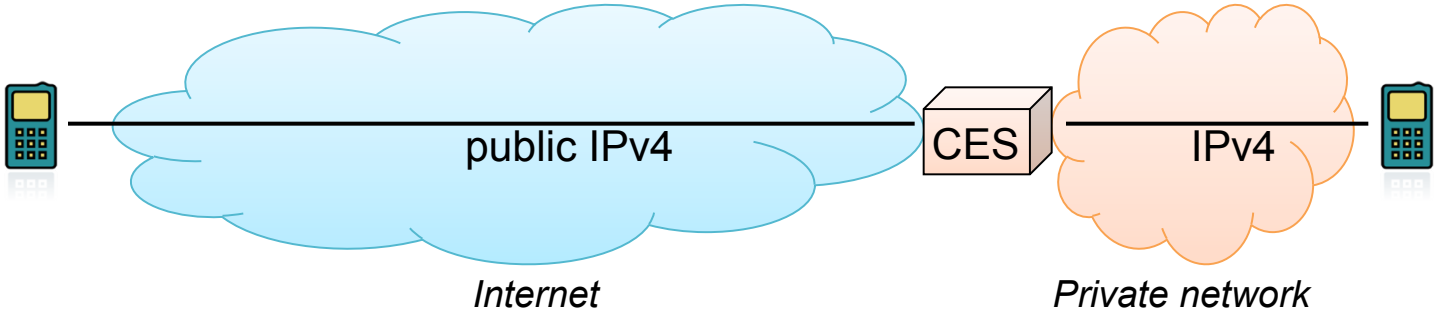


Scenarios

- CES-to-CES scenario: Both endpoints behind a CES



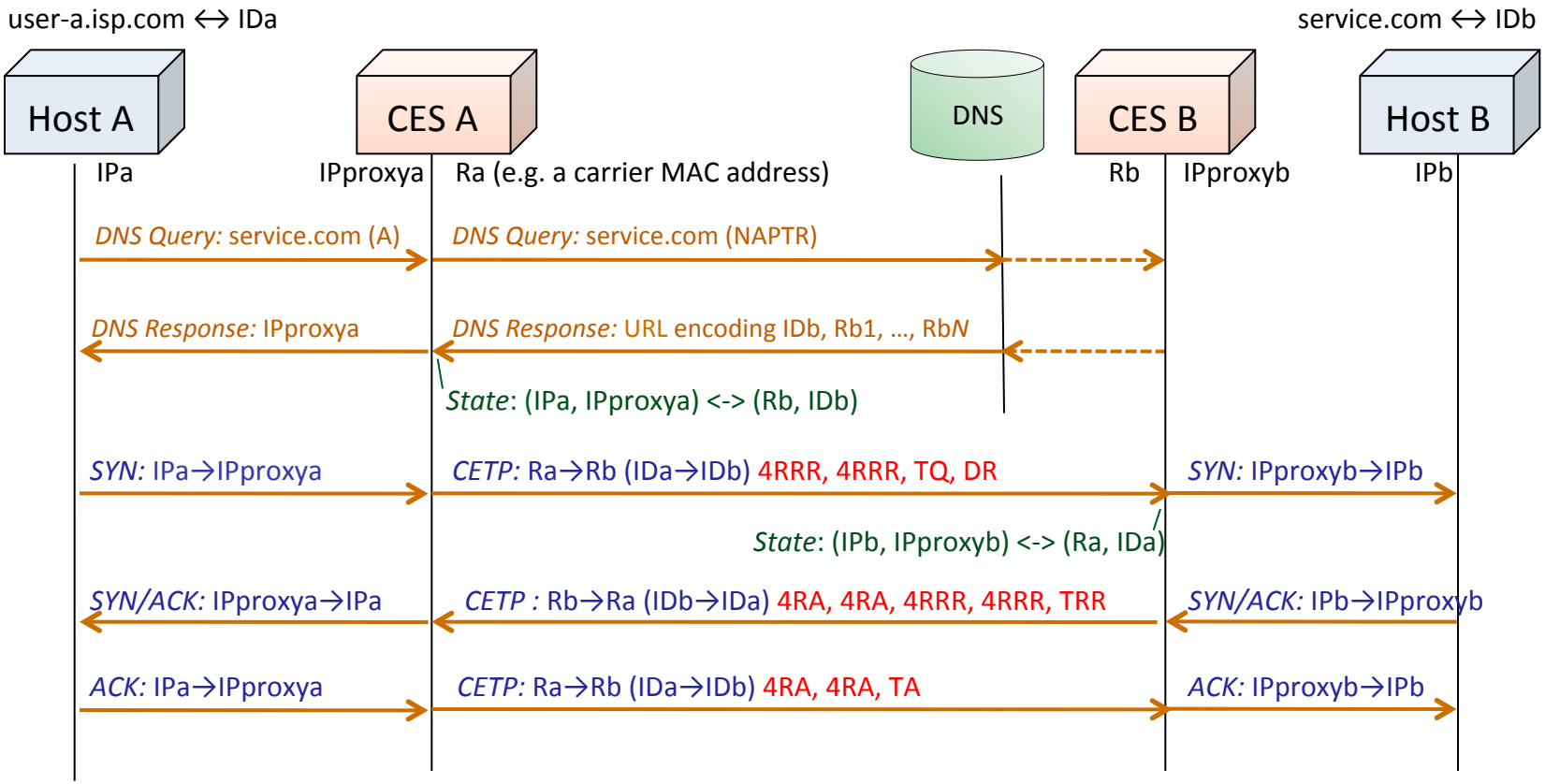
- Compatibility scenario: One endpoints behind a CES



Customer Edge Traversal Protocol (CEPT)

- Control signaling between CES devices
  - Many ID types from anonymous to certificates
  - Policy control for admitting traffic and for every CETP feature
  - Return routability checks (avoiding spoofed addresses)
  - Postponed connection state creation (prevent DoS attacks)
  - Negotiation of ID types
  - ID validity checks
  - Signatures
- Tunneling with header compressions
  - Transports the source and destination IDs
- TLV encoding → Extensible

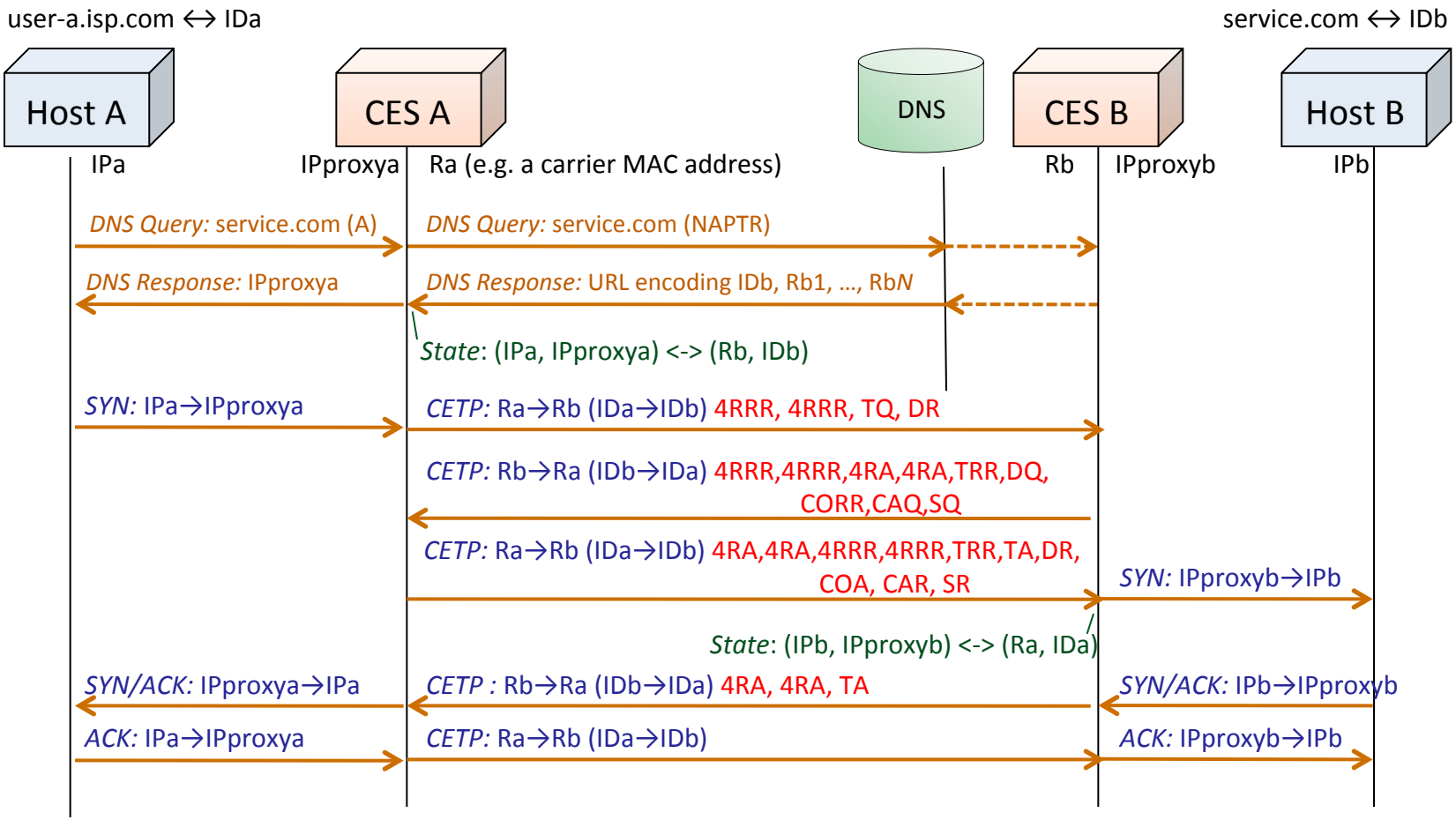
CETP Signaling Example (lax policy)



4RRR=RLOC Reliable reply, 4RA=RLOC Ack, TQ=Timeout query, TRR=Timeout reliable reply, TA=Timeout ack, DR=Domain reply

Aalto University

CETP Signaling example (strict policy)



4RRR=RLOC Reliable reply, 4RA=RLOC Ack, TQ=Timeout query, TRR=Timeout reliable reply, TA=Timeout ack, DR=Domain reply, CORR=Cookie reliable reply, COA=Cookie Ack, SQ=Signature query, SR=Signature reply

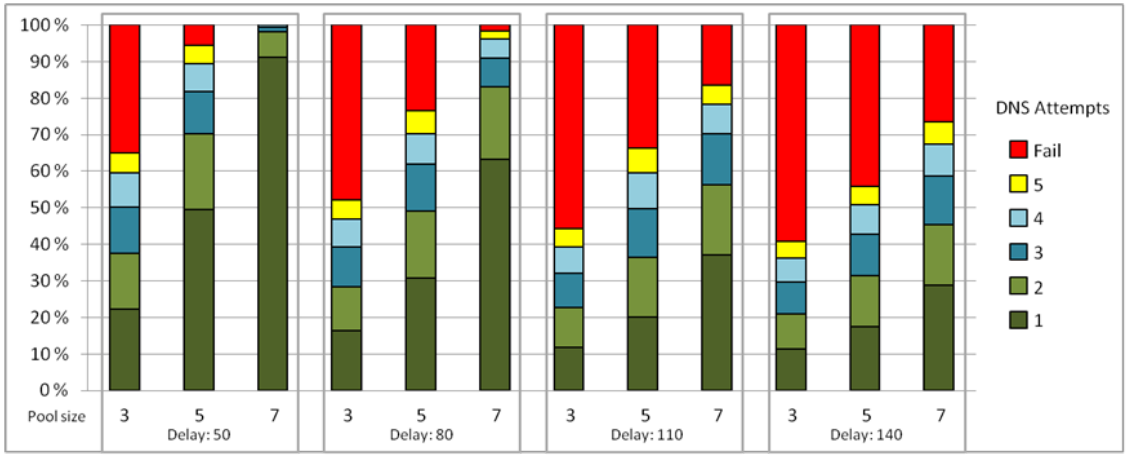
Policies

- The policy determines what is required before accepting a connection
  - Return routability check
  - Domain name checking
  - Certificates
  - Given type of ID
  - CETP Policy = a few vectors + a few scalars
- Input from other systems
  - Reputation level
  - DPI
- Can be either static or dynamic

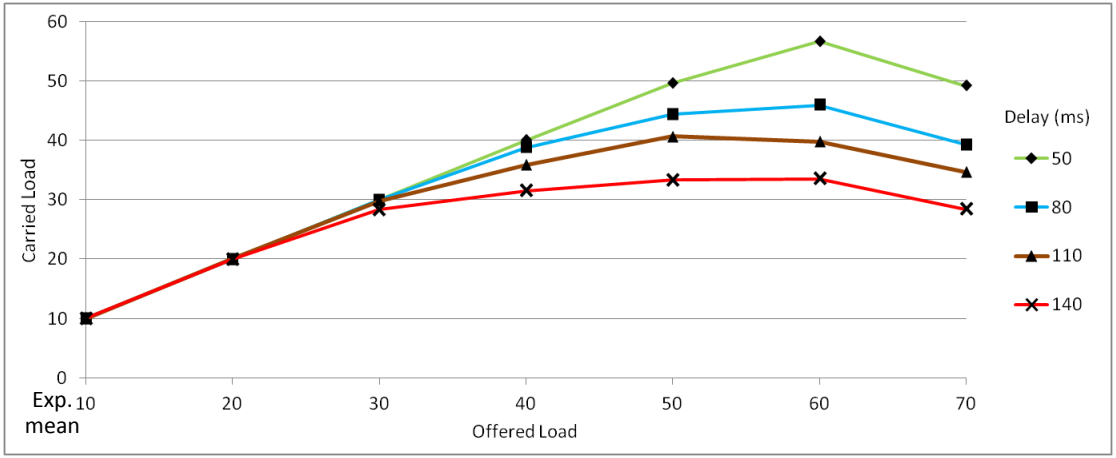
Interworking with Legacy IP

- Necessary to provide connectivity with the Internet
- Techniques for enabling inbound connections via CES
  - Do not require changes in either network or hosts.
- Operate with a pool or a single public IP address
  - A single public address can be reused for multiple sources and destinations
- Circular address pool
  - Uses incoming DNS queries to create state and forward subsequent data packets to the private network
  - Efficiency and performance are determined by network delay and size of the public pool
  - Service time of a flow = network delay → Erlang-B model applies

Interworking with legacy IP



Impact of delay and pool size with a fixed load of 60 new connections per second



Measurement of carried load vs. offered load for a circular pool of 5 addresses and different network delays

Application compatibility

- Like other NAT-like devices, CES must separately process protocols that transport IP addresses within messages (private IP → public IP)
  - Additional challenge: hosts have no global IP addresses
- Application testing study
  - to identify protocols that need Application Layer Gateways
  - to detect any protocols/applications that are not compatible with the CES concept
- Application Layer Gateways (ALGs) implemented for SIP, FTP, ICMP and a proxy for HTTP(S).
- Other things that work: Skype, SSH, Telnet…

CES Friendly Application Protocols = NAT friendly app. protocols

- Learn remote IP addresses in DNS (or DHCP)
- Use well-defined ports
- Use FQDNs for Identification
- Do not
  - Carry IP addresses in the content of their own control messages
  - Do not carry ports in the content of their own control messages
  - Use IP addresses as identifiers

Applications that violate the above, either need an ALG or work only as well as with NATs

Application Layer Gateways for SIP

- SIP transports IP addresses in the SIP header and in SDP, mappings needed for signaling and media flow
- The ALG adapts the IP addresses and the ports to achieve connectivity
- No global IP address → FQDN is a better alternative
  - FQDNs are allowed by SDP [RFC 4566] but usually applications use IP addresses
- Using FQDN edge-to-edge is more straight forward approach than IP
  - No need to store temporary information
  - Algorithms/code easier to understand

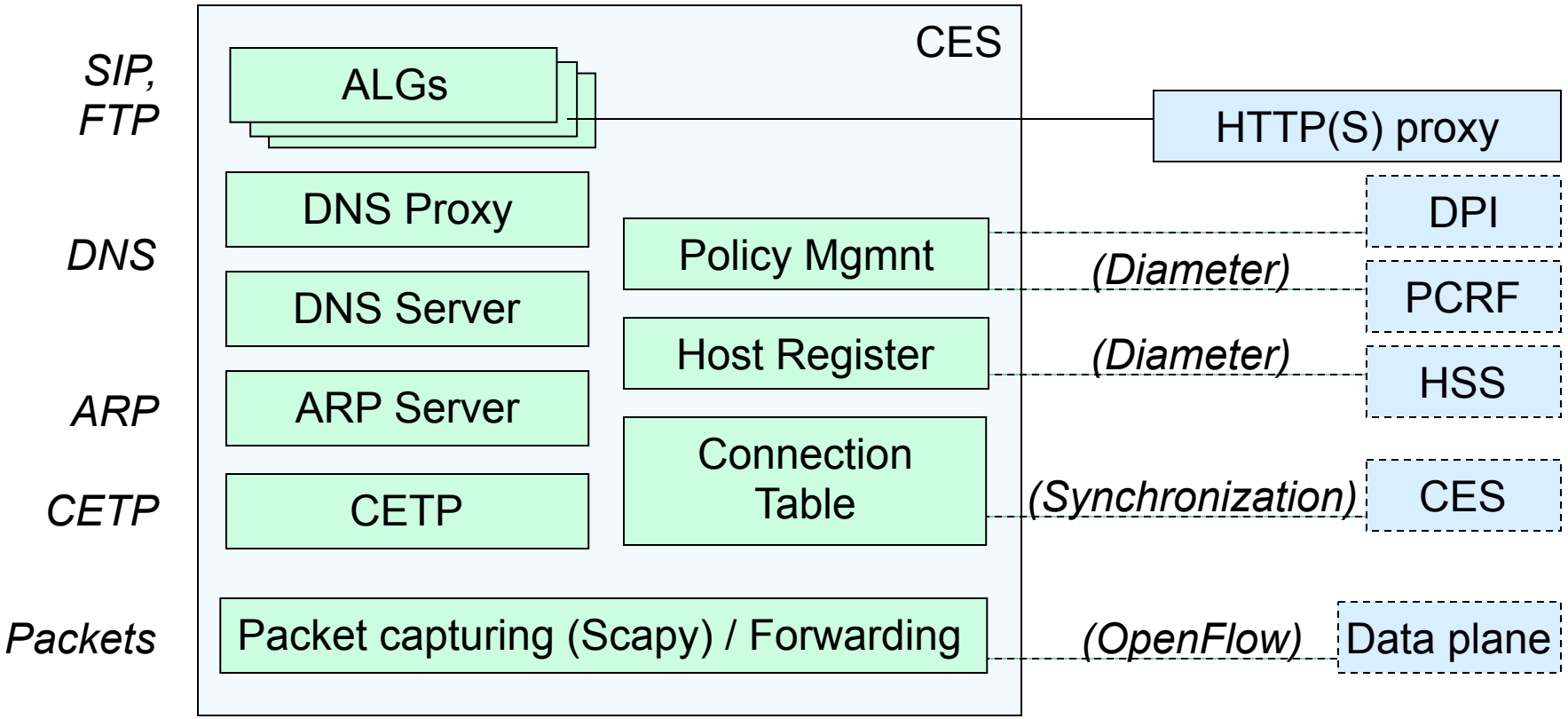
Application Layer Gateways for SIP

- Different scenarios require different algorithms
  - Information from the previous packets is used to define the type of the connection (similar to a stateful firewall)

Number	Algorithm name	Functionality
1	Local-FQDN	Packets inside private network
2	Local-IP	Packets inside private network
3	CES-CES-out-FQDN	Packets to another CES device
4	CES-CES-in-FQDN	Packets from another CES device
5	CES-CES-in-IP	Packets to another CES device
6	CES-CES-out-IP	Packets from another CES device
7	Public-in-IP	Packets from public network
8	Public-out-IP	Packets to public network
9	Public-out-FQDN	Packets to public network

- Prototype tested in 24 different scenarios

Prototype implementation running in real and virtual machines



CES Summary

- CES = Collaborative Firewall
- Is incrementally deployable one customer network at a time. We propose to start from Mobile networks and IoT
- Helps users to cooperate in order to root out selfish/unti-social strategies (hacking, trojans, botnets, spamming, fraud, stealing other people's information etc.)
- Allows hosts in private address space to communicate globally
- Introduces IDs to hosts/users/services
- Isolates technology choices in the core and in customer networks.

Thank you for your attention!

Questions?