

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Information-Centric Networks

Section # 6.3: Evolved Naming & Resolution

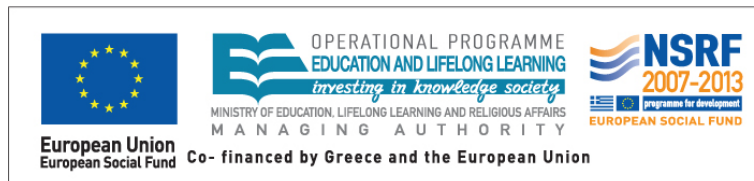
Instructor: George Xylomenos

Department: Informatics



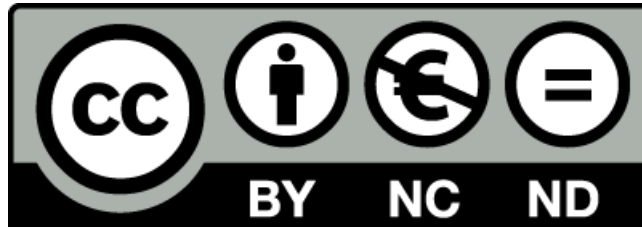
Funding

- These educational materials have been developed as part of the instructors educational tasks.
- The **“Athens University of Economics and Business Open Courses”** project only funded the reformatting of these educational materials.
- The project is being implemented as part of the Operational Program “Instruction and Lifelong Learning” and is co-financed by the European Union (European Social Fund) and national funds.



Licensing

- These educational materials are subject to a Creative Commons License.



Week 6 / Paper 3

- Middleboxes No Longer Considered Harmful
 - Michael Walfish, Jeremy Stribling, Maxwell Krohn, Hari Balakrishnan, Robert Morris, Scott Shenker
 - Operating Systems Design & Implementation (OSDI), 2004
- Main point
 - Middleboxes are everywhere
 - Internet purists scorn middleboxes (with reason)
 - But middleboxes offer valuable functionality
 - How can we retain the functionality without the side-effects?
 - The answer: Delegation Oriented Architecture (DOA)
 - Subset of the “layered naming architecture”

Introduction

- Two Internet tenets are often disobeyed
 - Every Internet entity has a unique network level identifier
 - NAT and host mobility prevent this
 - Network elements should not process other's packets
 - Caches, firewalls, NATs regularly look inside passing packets
- Layer violations make lead to real problems
 - SIP and P2P systems are hindered by IP address translation
 - Hard to deploy new applications
- But middleboxes offer useful functions
 - It would be even better if they could be located off-path
- The Delegation Oriented Architecture (DOA)
 - Globally unique identifiers in a flat namespace
 - Senders and receivers can indicate multiple such identifiers

NATs, NATs and Firewalls

- NAT and NAT
 - Hide networks with private addresses behind a public address
 - NAT looks at address and port, NAT only at address
 - NAT nearly always means NAT, so we only use this term
 - Convenience and flexibility in internal addressing
 - Security since only outbound connections are allowed
 - Static configuration needed to handle inbound connections
 - No way to use the same port for two applications
- Firewalls
 - Inspect inbound and outbound packets
 - Enforce filtering rules
 - Need to be on the path to the endpoint

Architectural overview

- Desired architectural properties
 - Packets should contain global identifiers
 - As used to be the case with IP
 - Application-independent way to express delegation
 - Delegates should not have to be on the direct path
- EIDs: endpoint identifiers
 - Must be independent of network topology
 - Can carry cryptographic meaning
 - 160 bit flat EIDs were chosen
 - Carried in a header between TCP and IP
- EIDs can be resolved to two things
 - An IP address (could be the delegate's)
 - One or more EIDs, as in a loose source route

Architectural overview

- DOA and the two Internet tenets
 - EIDs are globally unique identifiers
 - Packets sent to EIDs actually reach the hosts with these EIDs
 - Network elements only process packets with their own IP
 - A delegate can see that the EID does not match its own
 - It then resorts to local state to further forward the packet
 - No need for complex configuration at NATs
 - Just send the packet to the host with the right EID
- DOA and Internet evolvability
 - DOA allows managed service provision
 - You select your firewall provider and delegate packets to it

Detailed DOA design

- Header format
 - DOA header inserted between TCP and IP headers
 - TCP uses EIDs for checksum calculations
 - Carries at least one source and one destination EID
 - Can be extended
- Resolution and invoking intermediaries
 - The EID is resolved to an erecord containing:
 - EID being resolved
 - Target: IP or one or more EIDs
 - Hint (optional)
 - TTL: caching time
 - Transport connections are bound to the last EID
 - The others need to be traversed on the way to the destination

Detailed DOA design

- Security and Integrity
 - Anyone fetching an erecord must be able to verify its EID
 - Only the owner of an EID should update its erecord
 - A sender must not be able to forge an erecord
- EIDs are the hash of a public key
 - The erecord is signed with the corresponding private key
 - Does not prevent source EID spoofing
 - The receiver resolves the EID again to return responses
- Host software
 - Modified socket calls using a sockaddr_ein struct
 - Connect() and sendto() may require EID lookups
 - Accept() will return an EID
 - Hosts need to be bootstrapped with an EID resolver

Network extension boxes

- Network Extension Box (NEB): akin to a NAT
 - Offers some kind of delegated functionality
 - Preserve headers, using the EID to demultiplex packets
 - Simply insert the right IP address for the EID in the packet
 - End-to-end communication possible
 - Ports are not overloaded
 - VPNs can work around NEBs
 - NEBs can be configured automatically
- Configuration of cascaded NEBs
 - The endpoint must know what to put in its record
 - State must be established at NEBs or in the resolvers
 - This state must not be modified by attackers
 - Assume that each NEB only trusts the upstream NEB

Network extension boxes

- EID maps to EID
 - Each NEB adds an erecord from its EID to its parent's EID
 - Each NEB holds a mapping from its children's EIDs to their IPs
 - Incoming packets are resolved to a sequence of EIDs
 - As they pass NEBs they are sent to the next IP address
- EID maps to EID and a hint
 - As above, but the erecord also holds the IP address in the Hint
 - The IP addresses are included in the header
 - Each NEB can find the next IP address without internal state
- EID maps to IP address
 - Three round protocol to establish state at all NEBs
 - More complex, but the one actually implemented
 - Only requires a single EID to IP lookup by the sender

Network filtering boxes

- Network Filtering Box (NFB): akin to a firewall
 - Essentially remote packet filters
 - No need to be on the path to the endpoint
 - The NFB can work in statefull or stateless mode (as with NEBs)
 - The NFB receives packets and checks its rule base
 - Packets that pass the rules are attested
 - The NFB hashes the passed packet and signs the hash
 - A secret key shared between NFB and endpoint is used
 - Carried in an extension header
 - The endpoint only accepts packets attested by the NFB
 - Of course NEB and NFB can be combined in a chain
 - Can also be combined with an on-path middlebox
 - That middlebox can then check that packets are attested

Implementation

- User level software and Click modules in Linux
 - Click is a modular router building toolkit
 - Runs at both user and kernel levels
 - Allows mature implementations to migrate to the kernel
 - User level daemon (doad) resolves EIDs to erecords
 - Queries the DHT infrastructure
 - Inserts the EID to IP mapping in Click with a private IP
 - Returns the private IP to the client application
 - The client sends the packet with the private IP
 - Click rewrites the packet with the real IP and EID
- NEB prototype: user level implementation
- NFB pototype: user level and Click modules
 - Click module for clients to verify attested packets

Evaluation

- Round-trip times
 - A DNS and a DHT lookup are needed for resolution
 - DNS lookups take from 70 to 190 ms depending on caching
 - Median DHT lookups require 138 ms: needs improvement
 - Proactive caching as in Beehive
 - DNS names could also return erecords
 - Hosts could include their erecords in messages
- Packet size overhead
 - 68 byte header (44 fixed and 24 for security extension)
 - For large packets small overhead, for small packets quite high
- Processing time
 - DOA to IP translation does not take a lot
 - Filtering and verification take far more time

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

End of Section # 6.3

Course: Information-Centric Networks, **Section # 6.3: Evolved Naming & Resolution**

Instructor: George Xylomenos, **Department:** Informatics

