FARA: Reorganizing the Addressing Architecture

David Clark

MIT Laboratory for Computer Science

Bob Braden, Aaron Falk, Venkata Pingali

USC Information Sciences Institute

FDNA Workshop, SIGCOMM 2003
Karlsruhe, Germany
August 27, 2003

Introduction

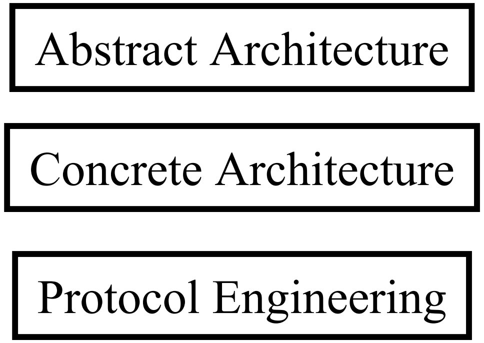
- NewArch project objective: create a better technical design for the Internet to meet today’s and tomorrow’s requirements.
- Can top-down, abstract “architectural” reasoning help in this effort?
- This paper describes an exercise to explore this question.
  - And hopefully explore some new design territory as well as much well-trod ground.

This is not a rerun…

- The original Internet design effort was largely bottom-up.
  - Found one approach that met the apparent requirements
  - Guided by some abstract thinking about protocol modularity, robustness, etc., but design effort was generally pragmatic.
  - A top-down discussion of the “Internet Architecture” and its relation to the requirements only came 10 years later (Clark, SIGCOMM 88).
- But: we understand more *(?)* about network architecture today than we did in 1978.

What is [Network] Architecture?

- And, "How do we know when we are done?"
  - The E2E Principle is part of the architecture
  - The hour-glass protocol stack is part of the architecture
  - Is the IP protocol spec part of the architecture?
- High-level abstract design principles are important and useful, but they are not a complete architecture.



Abstract Architecture

Concrete Architecture

Protocol Engineering

A 3-Step Approach

1. Define an abstract architectural model (FARA).
   - Encompass an interesting part of the design space, while leaving many details unconstrained.
   - Chose to maximize generality, to make the problem harder.
2. Define an architecture that instantiates FARA (the M-FARA architecture.)
   - Bind free parameters in FARA and define mechanisms.
3. Build a prototype of M-FARA.
   - Define real protocols, packet formats, facilities, scenarios
   - Build and demonstrate a toy implementation

The Perpetrators

- FARA design: Dave Clark
- M-FARA design: Aaron Falk, Venkata Pingali
- M-FARA prototype: Venkata Pingali, Ted Faber

With *lots* of advice from other members of the NewArch project:

- Bob Braden, Noel Chiappa, Mark Handley, Karen Sollins, and John Wroclawski

FARA Objectives

- Cleanly decouple end-system identity from network-layer forwarding functions
  - The familiar *location/identity* split
  - Support general mobility (includes multihoming)
- Avoid the need for a new global namespace for identity
- Provide E2E security with a range of assurance levels
- Generalize the architecture along several dimensions
- Support diverse naming & forwarding mechanisms

FARA* Model

* “Forwarding directive, Association, and Rendezvous Architecture”

(Also called "FARADS architecture")

- Re-modularization of function
  - Entities
  - Associations
  - The communication substrate
  - Forwarding Directives
  - Rendezvous

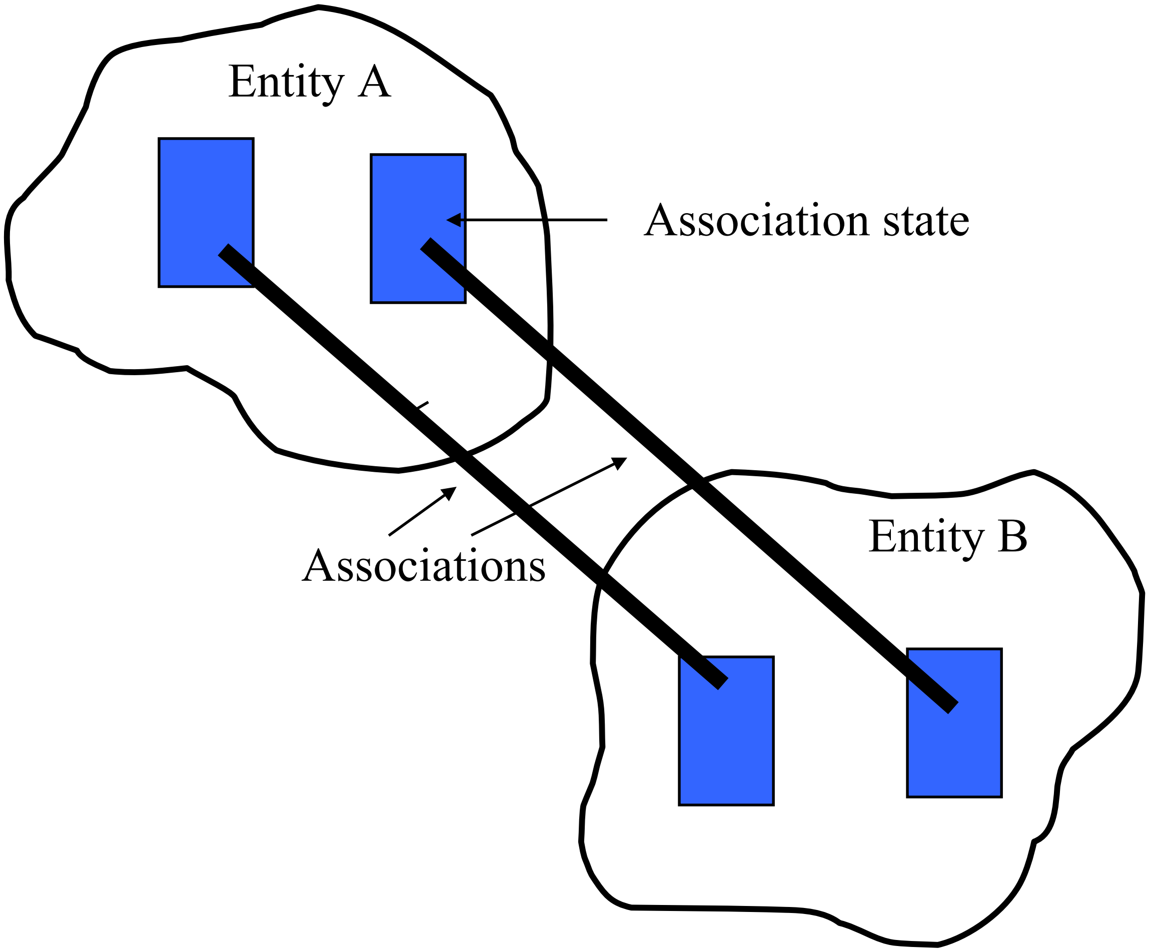
Entity

- The "thing" that has state and communicates.
  - A generalization of an end-system application.
- The smallest unit that can be mobile.
- An abstraction: might be a thread, process, set of processes, host, cluster of machines ...
  - (FARA allows all, but a derived architecture may only provide mechanisms to support a subset of these alternatives.)

Association

- A logical commun. link between two entities.
  - End-to-end communication abstraction
  - Has evolving shared communication state
    - E.g., for reliable delivery, E2E security, …
- Data packet carries Association ID (AId) for destination entity.
    - Entity uses AId to demux packet to association state
    - AId is local to entity, and its format is unspecified by FARA.
    - Packets may also carry source AId's

FARA end-to-end abstraction



Communication Substrate

- Packet delivery mechanism (~ network layer)
  - FARA assumes connectionless delivery, but makes no assumption about the delivery mechanism.
    - One possibility: h/h forwarding with globally-unique topological addresses, as in the current IP.
    - A derived architecture may allow multiple mechanisms.
  - Delivery: all the way to the entity
- Comm. substrate may also provide:
  - Delivery failure notification *(ICMP)*
  - Resource control (congestion notification, QoS)
  - Network-layer security (VPN tunnels, etc.)

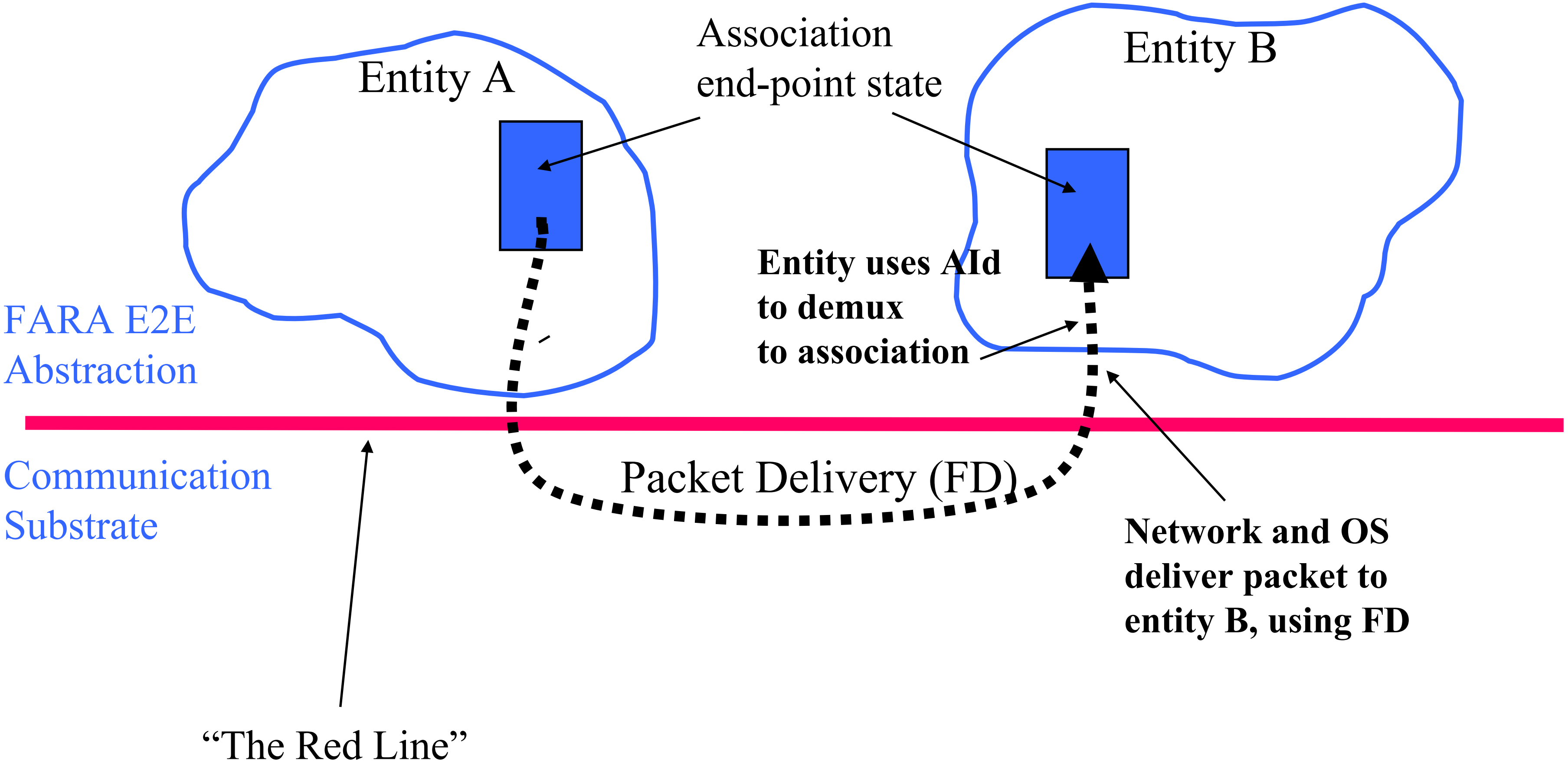
Forwarding Directive (FD)

- Each FARA packet carries a destination FD

  (and probably a source FD)

- Comm. substrate uses FD to deliver the packet.
- FARA does not specify the format or contents of FD.
  - Derived architecture must define.
  - Depends upon supported forwarding mechanism(s)
  - Could be simple global address, source route, or something more complicated
- When an entity that anchors an association moves: the FD changes, the AId doesn't.

Packet Delivery



Entity A

Entity B

Association end-point state

Entity uses AId to demux to association

FARA E2E Abstraction

Communication Substrate

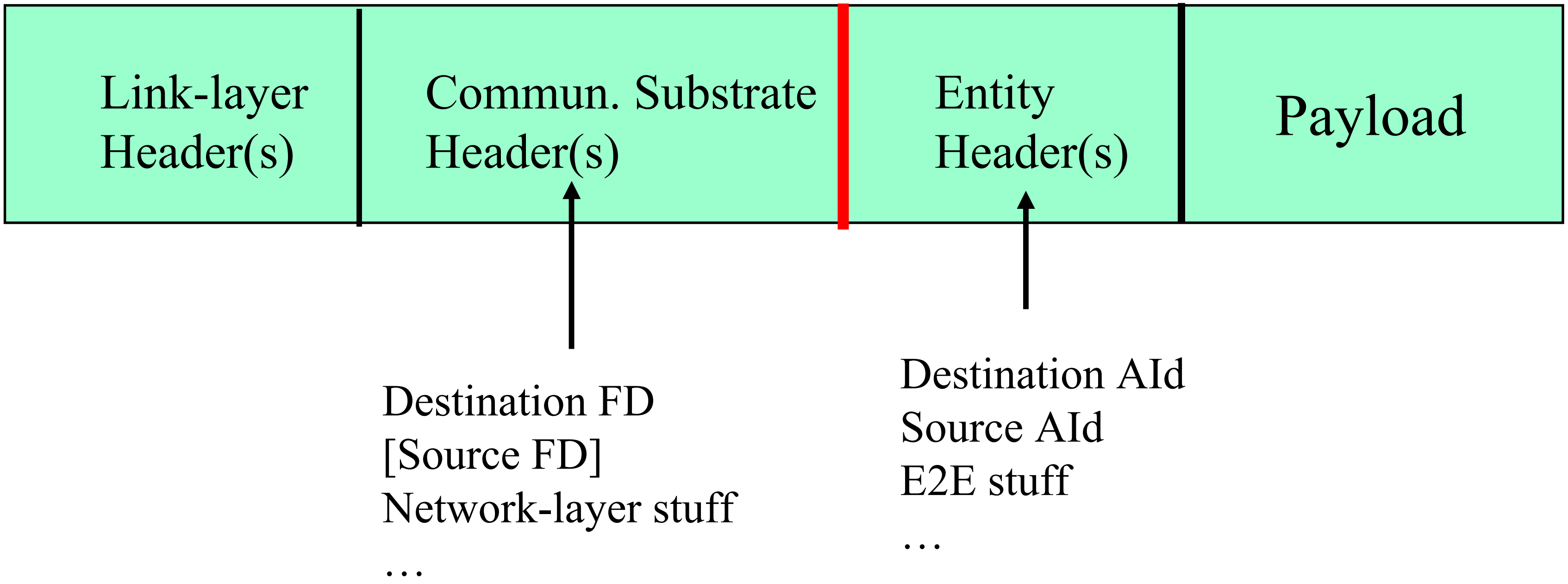
Packet Delivery (FD)

Network and OS deliver packet to entity B, using FD

“The Red Line”

FARA Packet Contents

Exact packet contents and format depend upon derived architecture and detailed engineering, but FARA implies the following:



Link-layer Header(s)

Commun. Substrate Header(s)

Entity Header(s)

Payload

Destination FD
[Source FD]
Network-layer stuff
…

Destination AId
Source AId
E2E stuff
…

FARA Assumptions

- An Entity is the unit of mobility.
  - For any flavor of logical or physical mobility.
- Associations do not have global names.
  - AId’s local to entity, invariant in a move.
- Entities do not have global names.
  - Defined implicitly by location – an FD.
  - There must be higher-level mechanisms to allow users to locate/construct FDs for target entities.
  - There will be (perhaps many) user-level name spaces -- “service names”
- Globally-unique network addresses are not required (but are permitted)

Creating an Association

- Use Rendezvous mechanism
- Simple Rendezvous case:
  - Discovery phase: Directory Service maps
    Service Name –> FD
  - Initiation phase:
    Send initial packet to target FD.
  - Target entity assigns AId
  - Handshake to create shared state for reliability, security, etc.

More General Rendezvous

- Directory Service => Service Name –> (FDi, RI)

  (RI: Rendezvous Information)

- Range of possible rendezvous mechanisms:
  - FD Generation at sender:
    - Sender generates complete FD from RI and FDi.
  - Third party remapping of FD:
    - Initial packet sent with FD of proxy/agent for target entity.
    - Proxy/agent rewrites (FD, RI) and forwards initial packet.
  - FD remapping at receiver:
    - Send RI in initial packet; target entity remaps locally.

Security

- FDs are not (necessarily) global and are not stable; they may be rewritten, may change due to mobility.
- Therefore, an entity must implement some packet validation mechanism
  - For initial association establishment
  - (Perhaps) for all subsequent packets in association
  - FARA leaves these mechanisms to the entities and to a derived architecture.
  - Intent: support variety of validation mechanisms
    - Trade security level vs. overhead.
    - E.g., lightweight security compable to IP's, or cryptographic security.

Step 2: Instantiating FARA

- FARA sounds nice, but is it self-consistent? Useful?
  - To get assurance, need to try deriving one or more specific architectures, complete with mechanisms.
- We designed and prototyped one derivative architecture, M-FARA.
- Chose an interesting point in FARA space --
  - Explore mobility and addressing aspects of FARA
  - Demonstrate location/identity decoupling
- Not a complete architecture

M-FARA Architecture (1)

- Network Addressing
  - Multiple distinct addressing realms
  - Addresses unique within each realm.
  - Support mobility across realm boundaries

M-FARA Architecture (2)

- Packet delivery mechanism
  - Hop/hop forwarding within realm, source routing across realms.
  - FD contains realm/realm source route.
  - Reverse FD constructed in flight
- To simplify route computations: assume a distinguished "core" realm.
  - Every entity knows FDup to reach core realm.
  - Directory Service contains FDdown to reach target from core realm.
  - Sender generates complete FD as (FDup, FDdown)

M-FARA Architecture (3)

- FD Management
  - When destination entity moves, construct new FD and tell sender.
  - Mobility agents (M-agents): 3rd party rendezvous points to maintain and update active FDs.
- Security
  - Authentication of sender initially & after every move.
  - Not authenticate every packet
  - DCCP-style connection nonce
- See paper for more information.

Step 3: M-FARA Prototype

- Toy implementation
- Entities, “FARA kernels”, and M-agents are Unix processes
- Associations mapped onto Internet overlays
- Reliable association uses TCP subset (1-byte window)
- Two addressing realms: IPv4 and IPv6
- It works …
  - Seamless migration of endpoint of a reliable association to new attachment point in same or different addressing realm.
  - Re-authentication using connection nonce when FD changes.

Other Possible FARA Instantiations

- IPv4-FARA
  - Restrict FARA generality to get functional equivalence of basic IP architecture
  - Entity is a process in a host.
  - Packet delivery: hop/hop forwarding
  - One addressing realm, one global network address space.

- IPv4-FARA would still be significantly different than current IP architecture.
  - FD = (IPaddr, dest port) pair
  - AId = "file descriptor"
  - Replacement for TCP:
    - No ports
    - New security mechanism(s)
    - Logically within user process, but sensible to implement it within OS kernel.

Some Limitations

- The FARA model does not currently handle:
  - Multicast
  - Middleboxes
- M-FARA does not:
  - Define QoS or congestion control mechanisms
  - Explore a range of rendezvous mechanisms
  - Attempt movement of an entity to a different end system (but this is an OS problem more than a network problem)

Prior Work

- The architectural paths we trod were well worn...
- Significant footprints we recognized were left by: John Shoch (1978), Jerry Saltzer (1982), Paul Francis (1994), Victor Antonov (1995), David Cheriton (2001), and Robert Moskowitz (2001), but there were many more...

Conclusions

- We have tried to give a linear explanation of a rather non-linear research effort.
- One conclusion: top-down architectural reasoning can be very useful, but you have to iterate between top-down and bottom-up (in the current stage of our understanding of network architecture, at least.)
- For presentations on FARA, documentation of M-FARA and its prototype, and download of M-FARA code:

http://www.isi.edu/newarch/fara