Middleware vs. Pointerware

3rd Swedish Networking Workshop
Marholmen, September 11-14, 2002

Christian Tschudin / UU
On the Architecture Papers and the Presentation Style

Two forthcoming (Oct 2002) HotNets papers, as submitted:

Braden/Faber/Handley (NewArch):
   From Protocol Stack to Protocol Heap – Role Based Architecture

Tschudin/Gold:
   Network Pointers

- Not a critique - take papers at face value
- Identify trends behind these two papers
- Paraphrasing instead of quoting
The Trend in a Picture: Reforming the Waist

Network layer will grow, split
Pressure to customize/evolve the network. Two trains:

1. Raising the level of the API. Consequences:
   - fixed IP semantics $\leftrightarrow$ rich NW services
   - protocol design $\leftrightarrow$ software engineering

2. Isolating a “proto”-IP layer:
   - more primitive than IP
   - untangle IP name/address/routing concepts
   → generalized network pointers (not names!)

Deconstructing IP — reassemble IP as one possible service
To provide a clean architecture for middle boxes, because today:

- x.5 layers (MPLS @ 2.5, IPsec @ 3.5, TLS @ 4.5) show that layering concept failed, not flexible enough
- feature interaction is not controllable
- implicit or no signaling at all: e.g. how to discover, configure a middle box?
The Trend in a Picture: Reforming the Waist

Network layer will grow, split
Protocol Heap Paper: The Proposal

- Relax strict nesting/layering for subfunctions
- Combine basic NW layer (sub-) functions at run time
- Explicit signaling, down to packet level
- Controlled composition
  - sanitized w.r.t. feature interaction

(Old) Background, Relations:

a) stack implementation discourse in the early 90ies,
  x-kernel, protocol stack graphs, flexible protocol stacks, ILP etc
b) Intelligent networks
Protocol Heap Paper: Three Building Blocks

1. “Roles” – functional modules for the heap, a hundred
   Ex: fragment, en/decrypt, de/compress, flow ctrl etc

2. “Actors” – realize (possibly jointly) a role, HW + SW

3. Set of packet headers instead of header sequence:
   – per role headers (“role–specific–headers” RSH)
   – RSH as parameters, signaling the actors
   – partial execution order rules
   – header set changes on the route

Heap structure is mirrored by unordered set of packet headers.
Roles have a unique ID (name)

Standardization of “well known” roles

Careful verification before standardization.

Inaccuracy in my presentation re Protocol Heap Paper:

- Braden et al. basically argue for a full heap (link layer up to application), but pick a partial heap on top of IP.
  - conceptually and actually it’s middleware.
Protocol Heap Paper: Forces at Work

- Telcos probably will like it:
  paper talks their language – signaling, feature interaction, verification, controlled evolution, standardization etc

- ISP might like it:
  support for middle boxes, probably official blessing to do economic engineering at the network edge (i.e. customers)

- IETF people could give in, too:
  more flexibility, retain control, still IP is king

OK, we did not look into soundness of header processing, complexity of social constructs (standardization) etc.
The Trend in a Picture: Reforming the Waist

Network layer will grow, split
NW Pointer Paper: Motivation / Diagnosis

Want routing steering, overlay forming support in the NW layer, because

- everybody needs overlays, so NW should provide basic mechanisms
- global end addresses put too much stress on NW arch, need a level of *indirection* e.g., explicit name to “route handler” translation
- want “directable indirection” for end nodes, permits to restore e2e semantics
NW Pointer Paper: Three Building Blocks

1. “Underlay”: work below IP
   – enable many Internets (VPN, Overlays)
   – paths as a basic abstraction, IP an interface

2. “Selectors”: a single packet header
   (pointer value), basically a demux layer

3. “Explicit Resolution”: map (anything) to selectors:
   – IP addresses (name!); selector = pointer to ARP cache
   – URL; selector = pointer to closest content cache
   – route through functional modules (… roles)
Middleware vs Pointerware: Discussion wrt Winternet

- Killing the “holy IP cow” for the sake of progress – agree?

- Heap and Pointer: harmony or dissonance?
  - heap: CONTROL!
  - pointer: ANARCHY!
  but might need each other.

- Assume that the proposed trend is real:
  does Winternet cover the new areas?