An Axiomatic Basis for Communication

M. Karsten\textsuperscript{1}, S. Keshav\textsuperscript{1}, and S. Prasad\textsuperscript{2}
\textsuperscript{1}University of Waterloo
\textsuperscript{2}IIT Delhi
Did you know that...

For example:

- NAT = ATM
- DNS: Forwarding overlay
- Source routing is heavily used in the Internet
Outline

• The unreasonable Internet
• The axioms of communication
• Notes on formalization
• Conclusions
The unreasonable Internet

- Original Internet assumptions
  - Static public IP addresses
  - 5-layer stack
  - No layer violations
  - Forwarding based only on IP routing tables
In fact...

- All these assumptions are violated
  - DHCP, NAT, Mobile IP -> dynamic IP
  - Many more layers (VLAN, P2P, MPLS ...)
  - Layering extensively violated (NAT, firewall, DNS redirection)
  - Forwarding based on VLAN ID, MPLS ID, source IP (!)
But...

- It still works
  - mostly
    - for most people
- Why?
Hypotheses

- All the changes to the original architecture still preserve some invariants (wrt forwarding)
  - ‘Axioms’ of communication
- If we can state these axioms and analyze them, we can know the limits of what is feasible
  - eg. deliverability of messages
- We can also come up with an expressive pseudo-language to implement any packet forwarding scheme
Divide and Conquer

- We are only studying connectivity (naming, addressing, routing, forwarding)
- Other areas, such as medium access, reliability, flow control, congestion control, and security are ignored (for now)
A diversion...
Axiom: arch

Coliseum, Rome
Axiom: lintel

Big temple, Thanjavur, India
Internet-style architecture

Hearst Castle, California
Axiomatic engineering

Tenerife Airport, Tenerife, (Calatrava)
Axiomatic engineering

Bilbao Museum (Gehry)
Outline

• The unreasonable Internet
• **The axioms of communication**
• Notes on formalization
• Conclusions
The axioms

• Will state them, and try to explain why we chose them

• Grouped into a few sets
Naming and binding
Naming and Binding

- Saltzer (1978) with some modifications
  - An object is a software or hardware structure
  - Name is a regular expression that refers to a set of objects
- Binding
  - noun: mapping from name to set of objects
  - verb: choosing the object mapped to a name
- Address
  - A lower-level name used to access an object
Naming and Binding...

- **Context**
  - Set of mappings
  - Name is interpreted wrt a context (multiple contexts may resolve the same name differently)

- **Resolution mechanism**
  - Locates the mapping for a name within a context
Communication axioms
Millau Viaduct
Communication axioms

• Certain objects can *directly communicate* with each other
  • shared memory or on a physical medium

• *Network Processing Object* (NPO) is an object that can directly communicate with some other NPO(s)

• Each NPO has a local set of mappings, called its *context state* (e.g. forwarding table)
Communication axioms

- NPOs that can directly communicate with each other are *neighbours*

- Unit of communication is a *message*
  - message = header + payload

- *Any name in a header is an address*
  - Header can have a *stack* of addresses
  - Topmost one is the current *destination address*
Communication axioms

- *Forwarding* is an extension of direct communication where neighbours repeatedly pass on a message to a set of neighbours, so that the message eventually arrives at a set of destination NPOs

- transitive relation of direct communication

- *Resolution* can not only return a ‘lower-level’ name, but also set of neighbours for a name
Operations
Fundamental operations

- Split operations into forwarding (move messages) and control (routing, path setup, remote name lookup)

- We describe some fundamental ways to move and manipulate a message, e.g.
  - receive/send – direct communication
  - push/pop – modify address stack
  - lookup (a name in a context table)
Forwarding

• Define local context state as
  • \{<\text{name} \rightarrow \{<\text{NPO}, \text{name}>}\}\}

• Forwarding code:

  message msg = receive();
  name n = pop(msg);
  \{<\text{NPO}, \text{name}>\} S = lookup(n);
  for each <\text{NPO}, \text{name}> s in S
    outmsg = copy(msg);
    push(outmsg, s.name);
    send(s.NPO, outmsg);
  endfor
Structural axioms
Baha’i Temple, New Delhi
Structure axioms

• The NPO that pushes an addresses and every NPO that resolves (i.e. lookup) or removes that address are peers

• Peers that push and pop an address establish a link

• Sequence of peers forming a link is a path
Structure axioms

• *Iterated forwarding* a message is binding its destination name to a set of destination NPOs

• Set of peer NPOs that forward a message with the same destination address to the same set of NPOs provide a *consistent binding*

• A *scope* of a name is the set of peers that provide a consistent binding for that name

• Scopes may contain special names, such as the *broadcast* name

• Mechanisms to provide consistency in a scope are called *routing*
Outline

- The unreasonable Internet
- The axioms of communication
- Notes on formalization
- Conclusions
Formalization

• We associate operational semantics with each operation, consistent with axioms

• Desirable properties become theorems

  • e.g. we can ask “Is deliverable (A,B) a valid theorem in our system?”
Operational semantics

- Each operation updates the state of an abstract machine

- configuration = 
  <stack of values | context state | operations>

- e.g.
  $<\text{n}_1\text{n}_2\text{n}_3...\text{n}_d|\text{cs}|\text{pop};\text{p'}> \rightarrow <\text{n}_1,(\text{n}_2\text{n}_3...\text{n}_d)|\text{cs}|\text{p'}>$

- Well-known theory to reason about invariants about partial correctness and progress
Outline

• The unreasonable Internet
• The axioms of communication
• Notes on formalization
• Conclusions
Railway Station, Lisbon (Calatrava)
Sample Observations

• NAT $\approx$ MPLS $\approx$ ATM
  outgoing source port $\sim$ label

• Recursive DNS lookup – forwarding based on DNS destination using UDP tunnels

• Stack of $<$port number, IP protocol ID, IP address, Eth protocol ID, MAC address$>$
  $\approx$ record route and source routing
Conclusions

• The Internet is complex, yet it works

• We think it’s because protocol designers implicitly follow some rules (axioms)

• We explicitly state the axioms – clarity

• Allows us (hopefully) to do formal analysis: correctness, deliverability, (performance, errors)

• Also allows us to construct a universal forwarding engine