

# Plutarch: An Argument for Network Pluralism

Andrew Warfield

`andrew.warfield@cl.cam.ac.uk`

Jon Crowcroft, Steven Hand,  
and Andrew Warfield  
*University of Cambridge*

Richard Mortier  
*Microsoft Research Cambridge*

Timothy Roscoe  
*Intel Research Berkeley*



UNIVERSITY OF  
CAMBRIDGE  
Computer Laboratory

# Why a new architecture?

---

- ▶ “Architecture” papers need good motivation...
- ▶ Three major reasons:
  - 1. Things IP is bad at. (Deficiencies)**
    - Mobility, multicast, route convergence, non-endpoint-based addressing, novel link layers, service differentiation, management, accounting, etc.
  - 2. The apocalypse. (Scale issues)**
    - Insufficient Address space, routing collapse, DoS/ Worms
  - 3. Freedom to innovate. (Boredom)**
    - Commodification of IP stifles innovation
    - Either do incremental polishing or do overlay nets.



# IP Philosophy

---

“The top level goal for the DARPA Internet Architecture was to develop an effective technique for multiplexed utilization of existing interconnected networks.”

- D. Clark, “The Design Philosophy of the DARPA Internet Protocols”



UNIVERSITY OF  
CAMBRIDGE

Computer Laboratory

# IP Design Philosophy

---

- ▶ IP enabled internetworking by homogenising the network and transport layers
  - ▶ A uniform general-purpose set of protocols
- ▶ By standardizing the middle, layers above and below were free to evolve.
- ▶ **Clearly, IP was correct!**
  - ▶ Hard to imagine faster growth...



# Looking forward

---

- ▶ IP has been very successful, *but...*
  - ▶ Is it likely to be the eternal Internet protocol?
- ▶ Can we realistically expect v6, or any other potential replacement to be more successful?
- ▶ Most importantly: *What is really achieved* in deploying an incrementally more scalable protocol?
  - ▶ If you think v6 has deployment problems, wait till v8!



# Alternate approaches

---

- ▶ Two things seem important:
- ▶ **Less homogeneity.**
  - ▶ An end-to-end issue, IP is an interface and it can't anticipate everything.
- ▶ **Focus outside the scope of the IPv4 Internet.**
  - ▶ Address transition.
  - ▶ Allow specialized networks
  - ▶ Do not presume to have a drop-in replacement

# Our Position:

---

An Inter-networking architecture  
must allow communications  
between dissimilar networks  
**without** mandating a standardized  
data path.



# Plutarch

---

- ▶ Aim to provide a minimal control plane to allow dissimilar networks to arrange communications.
- ▶ An *extensible* Inter-network.
- ▶ Two fundamental concepts: Contexts and Interstitial Functions.
- ▶ Develop a management/control service to address naming and connections at an inter-network granularity.





# Not completely new...

---

“We call a network which builds coherent user level semantics from a regionalized infrastructure and qualitatively heterogeneous communication technologies a Metanet.” – Wroclawski, Metanet Whitepaper (1997)

“In particular [the Yellow Book] aims to provide endpoint communication across multiple independent networks.”

– Bennett, INDRA Note 967 (1980)



# Core Network Dissimilarities

---

- ▶ **Naming**

- ▶ Google, not DNS!

- ▶ **Addressing**

- ▶ Two part address: (network address, opaque address)

- ▶ **Transport**

- ▶ Mapping across protocols. Congestion/flow control etc.

- ▶ **Routing**

- ▶ Again at a network granularity – like BGP!



# Contexts

---

- ▶ A *Context* is an area of the network that is homogenous in some regard.
  - ▶ Principally naming, addressing, routing and transport.
- ▶ Two purposes:
  - ▶ ***Locational***: serve as descriptors allowing end-to-end services to be composed through network closures
  - ▶ ***Mechanical***: describe a set of communication mechanisms within which an endpoint might bind for a session



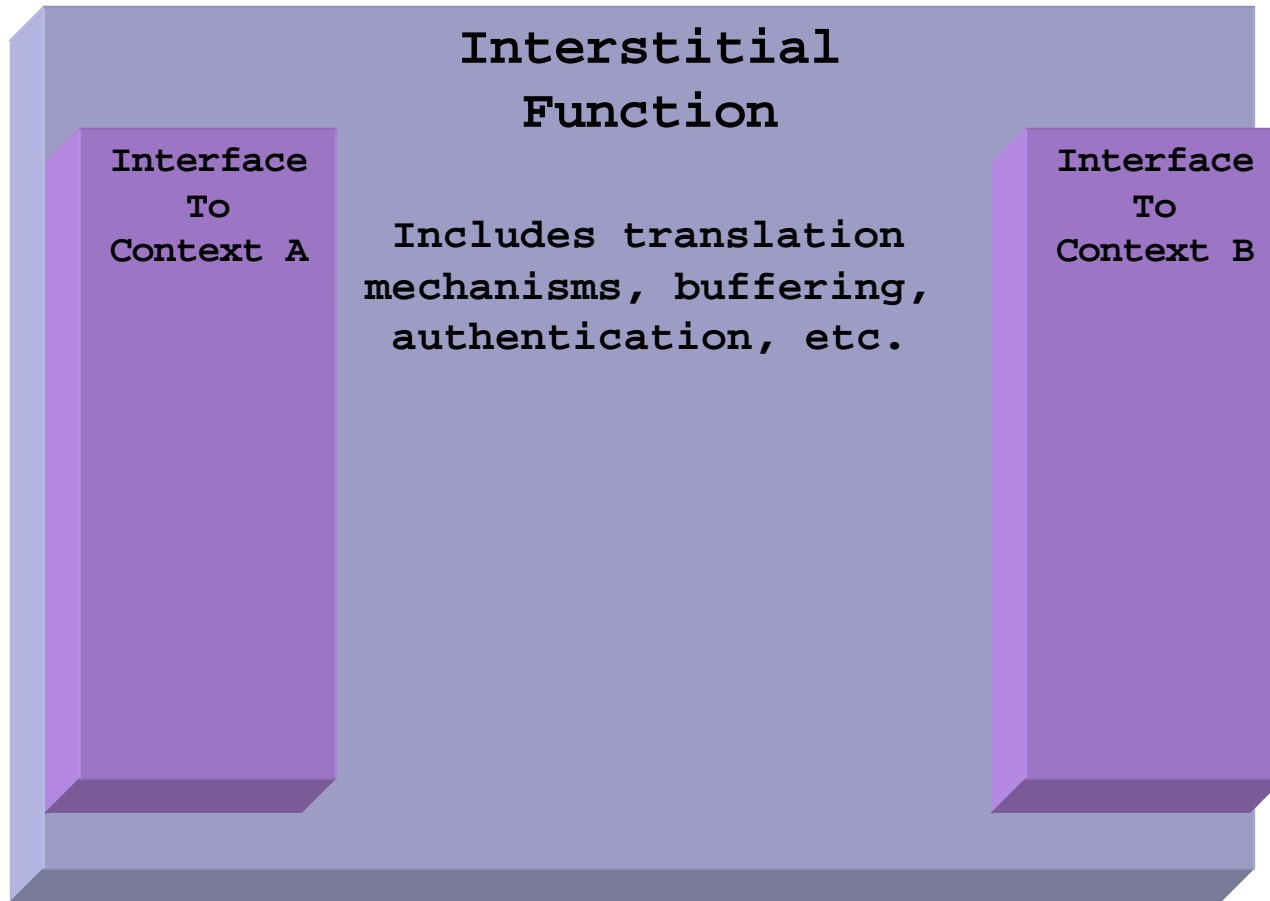
# Interstitial Functions (1)

---

- ▶ Exist at the borders between contexts.
- ▶ Allow data to cross contexts.
- ▶ Already have such creatures
  - ▶ NAT boxes (IP nets), BGP routers (AS domains)
- ▶ Not just IP though!
  - ▶ Dissimilar transport (IPv4 <-> ATM)
  - ▶ High-level overlays

# Interstitial Functions (2)

---



# Example: Border Nets

---

- ▶ Attempting to connect from a GPRS laptop to a sensor net via the Internet.
- ▶ Both ends are behind opaque gateways.
  - ▶ Same as the two-NAT problem... connection impossible
- ▶ Three stages to communication:
  - a) Name/Address lookup
  - b) Chained context instantiation
  - c) Application binding / Communication

# Ex: a. Lookup

---

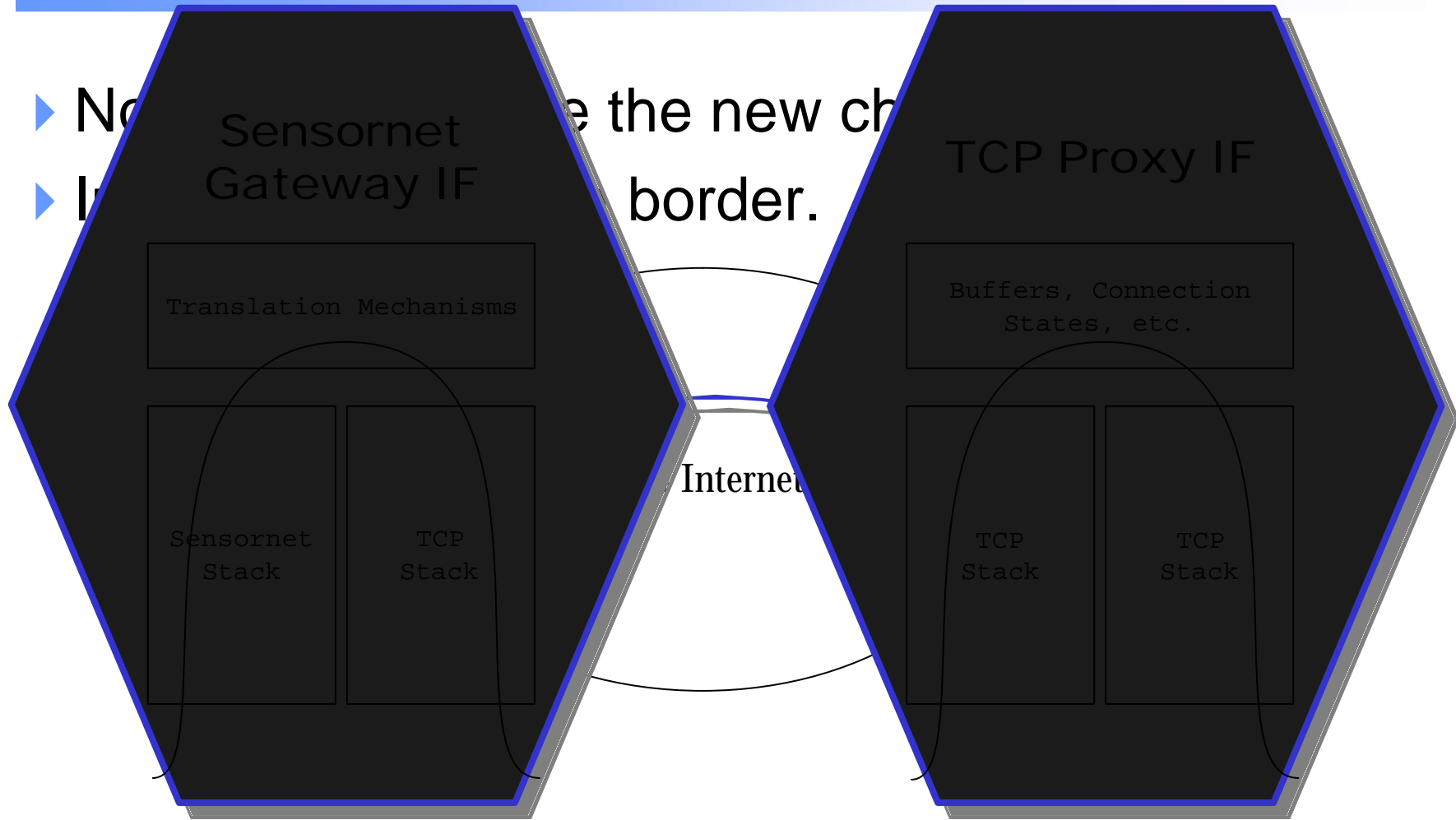
- ▶ Want a distributed service that passes queries and advertisements across contexts.
- ▶ Search on name=value pairs.
- ▶ i.e.  

```
find_route(destContextName=myExperimentalSensorNetwork,  
           pathProperties=(protocol=QueryProtov1.2,  
                           connection=reliableByteStream))
```
- ▶ Properties act as hints... endpoint selects.
- ▶ Get back a list of candidates, and pick one.



# Ex: b. Instantiation

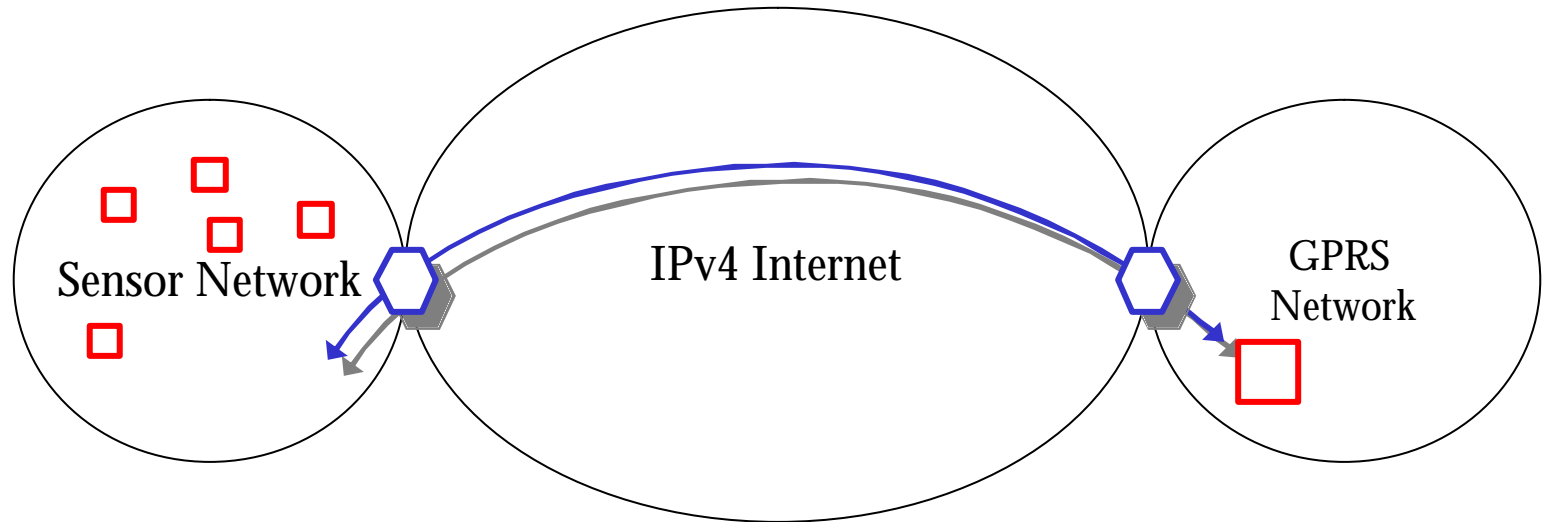
- ▶ No change to the new client
- ▶ IP at the border.





# Ex: c. Apps Bind and Talk!

- ▶ Finally, app binds to the newly created context.



- ▶ End-to-end protocol-specific tunnel.
- ▶ Midpoint customisations



# Future work

---

- ▶ Go beyond ranting position paper.
- ▶ Many unsolved issues.
  - ▶ lookup,
  - ▶ semantics of IFs
  - ▶ Impact / issues on end-to-end service.
  - ▶ Failure
  - ▶ Garbage collection?
- ▶ Build something!



# Review

---

- ▶ IP got us here through homogeneity
- ▶ Want to extend all that – embrace heterogeneity
- ▶ Key components to our architecture are contexts and IFs.
- ▶ Also important is the accompanying infrastructure to make it work.



# Conclusions

---

- ▶ Increasingly difficult to extend IP to support demands of new applications and environments.
- ▶ We propose *Plutarch* an architecture that eschews homogeneity, allowing independent networks to work together for end-to-end communications.

(the end)



UNIVERSITY OF  
CAMBRIDGE

Computer Laboratory

# Isn't this Active Networks?!

---

- ▶ (a.k.a “What about untrusted code?”)
- ▶ Finite set of common IFs.
- ▶ Can be served from trusted repositories.
- ▶ IFs not carried inside packets.



# Where are these IFs going to run?

---

- ▶ IFs live on gateway nodes – these nodes must be accessible from both contexts involved.
- ▶ Small matter of deployment. ;)
  
- ▶ Need a platform for gateways to execute IFs in a reliable, accountable manner...
- ▶ (Have I mentioned XenoServers?)



# Backup slides start here.

---



UNIVERSITY OF  
CAMBRIDGE

Computer Laboratory

# The argument is circular!

---

- ▶ Plutarch is by no means the be-all-end-all solution!
- ▶ There may be a general purpose protocol that solves all these problems.
  
- ▶ However, if that is the case, we would expect that it would *emerge* within Plutarch and grow to make all other contexts redundant.



# Naming and Addressing

---

- ▶ We figure this is one of the hardest parts
  - ▶ (but not the only hard part)
- ▶ Allowing heterogeneity means a diverse name/address space.
- ▶ But this is what we have already... v4 is stretched, and v6 primarily suggests more bits
- ▶ We imagine (context, internal name) pairs

