D3 – Initial Overall Architecture

The Original Technical Design Goals and Principles

1. **Interconnection of Existing Networks** (rather than imposing a new unified global network that could become obsolete later)
2. **Survivability:** the entities communicating should be able to continue without having to re-establish or reset the high level state of their conversation.
3. **Support for multiple Communication Service Types:** the Internet should support, at the transport layer, a variety of applications that could be distinguished by differing requirements for such things as bandwidth, latency and reliability.
4. **Support for a variety of Physical Networks**

The Original Technical Design Goals and Principles

5. **Distributed Management:** Internet resources should be able to be operated and managed by distributed stakeholders.
6. **Cost Effectiveness:** desire for efficient use of network resources.
7. **Simple Host Attachment:** desire for keeping the complexity of host protocol stacks low, so that deployment was not hindered.
8. **Resource Accountability:** desire for understanding and monitoring the usage of network resources.

Design Principles for the Trilogy Architecture

- Information Exposure
- Separation of Policy and Mechanism
- The Fuzzy End to End Principle
- Resource Pooling

Information Exposure

- *The data (or transaction) that uses up scarce networking resources, through being sent (or acted on) should **integrate control information** (e.g. a metric) that reflects its resource usage in 'real time'.*
  - *The control information should provide sufficient information about resource usage to support an efficient and accountable allocation of resources.*
  - *The control information may be used as a variable in a pricing structure.*
- Examples:
  - forwarding of packets by a router ('data')
  - creation /maintenance of a connection by a middlebox ('transaction').

Information Exposure

- Can be seen as extending the original Internet Design Principle of **connectionless datagrams** (signalling goes in-band i.e. along with the packet) in two ways, in order to take account of the abstract goal of Design for Tussle.
  - Firstly, by including the idea of accountability for usage of scarce resources, and
  - secondly by recognising that transactions as well as packets consume resources.
- *Some open questions with this Design Principle*
  - For a transaction, it may not be immediately obvious how much resource the transaction will consume, e.g. how much processing will it take to process the message? At one level, uncertainties are risks, which can also be handled as a cost.
  - Adding the integrated control information in itself consumes some resource. It is assumed that sufficient benefit is gained, for instance in terms of the efficiency with which resources are used.

Separation of Policy and Mechanism

- *Allow a network entity **local choice** according to its priorities (policy). Have a **common protocol** or method through which the policies interact to determine how networking resources are shared.*
  - *Constrain conflicting policies via the Information Exposure Design Principle.*
- Examples:
  - Kelly's view of congestion control
  - Inter-domain routing with BGP

Separation of Policy and Mechanism

- analogous to the original Internet Design Principle of 'IP over everything and everything over IP'. That defined a simple IP layer to transfer data, with freedom about the applications above and the networking technologies below.
- *Some open questions with this Design Principle*
  - So far, detailed considerations have focussed on resource-related examples of the Design Principle. We need to work out how to capture reachability-related examples more formally.
  - Does the Information Exposure principle really give sufficient constraint on policies so that they don't conflict with each other? Detailed case studies are needed, including reachability related ones.

The Fuzzy End to End Principle

- *Allow the endpoint to explicitly delegate some functions into the network, so the end is effectively a distributed system. This may imply some state in the network, which should be "soft and hinty".*
- Examples: Application prioritisation, Protection, Content caching and targeting, Finding an appropriate peer

The Fuzzy End to End Principle

- closely related to the original Internet end to end Design Principle
- *Some open questions with this Design Principle*
  - The main open issue has already been mentioned, i.e. whether the Design Principle applies only to delegation to an application-level intermediary, or also to other functions.
  - Another open issue is how to extend this Design Principle to something about more general delegation or relocation of control functions (a possible example might be BGP communities). This is a known desire, see [Trilogy08] but we don't yet have a theory about the set of techniques to achieve it.

Resource Pooling

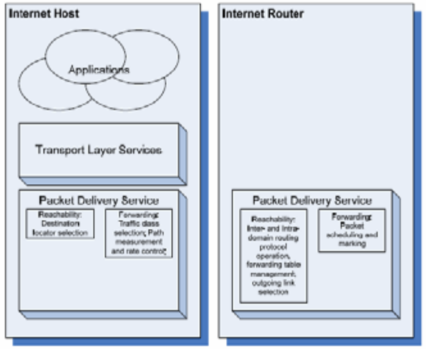
- *Allow sufficient pooling of resources to be effective.*
  - *Ensure that the resource pooling mechanisms don't conflict with each other.*
- making the network's resources behave like a single pooled resource, i.e. separate resources appear to act as one large resource.

Resource Pooling

- Our Resource pooling Design Principle can be seen as extending and formalising the original Internet Design Principle that resilience should be achieved through redundancy and diversity rather than through super-reliable individual components.
- *Some open questions with this Design Principle*
  - What abilities does the network need to have in order to achieve effective pooling of a particular resource? For example, re-bind mappings so packets can be re-routed.
  - How to ensure that one resource pooling mechanism doesn't conflict with another? Some worked examples are needed; we have an idea how to resolve the specific conflict mentioned above [Handley08]

Initial Architecture

- Packet delivery functionality
  - Reachability plane: hop-by-hop outgoing link selection
  - Forwarding plane: how the transmission resource on each link will be apportioned between packets
- Transport Services (functions that are implemented in a pure end-to-end fashion)
  - Examples: reliability, flow control, message framing, …



Reachability plane

- Are source locators required?
- What other packet information influences the path?
- Are locators from a single global namespace?
- What reachability information or control is shared, beyond the locator spaces themselves?
- What additional identifier spaces are used to manage the global topology, and how?

Forwarding plane

- The fundamental packet delivery service is best-efforts:
- Nodes along the delivery path mark property information about the path into packets, either explicitly using bits in the header, or implicitly
- implicit requirement that an end system generating a sequence of packets must be able to depend on consistency for their paths
- What is the level of path property information provided by the network?
- How does the forwarding plane distinguish traffic types?