

Chapter 2

A Survey of Novel Internetwork (and Naming) Architectures

When all think alike, then no one is thinking
Walter Lippman

Internet architectures may be broadly categorized into either incremental or radical proposals. Incremental architectures, such as [119, 130, 158], generally aim at addressing particular limitations of the current Internet architecture through patching, while radical architectures, such as those supported by the FIND [10] and FIA [9] initiatives, tend to adopt a clean-slate approach to designing a “better” Internet, without being necessarily restricted by the current Internet model.

We classify network architectural proposals based on the service model they intend to support. This highlights a growing interest in data-centric network services. This chapter serves two main purposes: (1) to survey the literature and highlight commonalities and differences across the spectrum of solutions, and (2) to present a reference for the rest of the book, and introductory tutorial. We start in Sect. 2.1 by classifying architectures based on the service model they are intended to support. In Sect. 2.2, we proceed to overview several clean-slate architectural instances that we shall refer to throughout the rest of the book.

2.1 Classifying Network Architectures

Before discussing our classification approach, we recall some general definitions. A *computer network* is an inter-connection of computers over which information¹ flows. The *network architecture* is the conceptual design and the fundamental operation structure of a computer network. Based on these definitions, one may clearly recognize the obvious defining structures of a computer network: computers and inter-connections, communication, and information structures.

¹Information, content, and data are used interchangeably within this chapter, unless otherwise specified, to represent data abstractions recognized by the network.

2.1.1 Classification Approach

How do we approach the classification problem given the complexity of the design space? In other words, what should the defining element(s) of our classification model be? We start by recognizing that every design is intended to support a set of goals, which generally encapsulate the pressing needs/requirements of users.² Generally speaking, the design process then involves converging on a set of defining structures, and proceeding to optimize those. The outcome is an architectural design that is comprised of the following abstraction levels: (1) the *outer-architecture* represents what the network user can see. This is analogous to the network service interface or *Instruction Set Architecture (ISA)* which defines the addressing modes, the data object semantics, and the available operations; and (2) the *inner-architecture* represents the internal operation structure of the network including the low level substrate structure and the functional aspects to support the outer-architecture.

We believe that both abstraction levels provide useful and complementary insights regarding the architectural landscape. Hence, to help answer our question of what the defining element(s) of the taxonomy should be, we found it useful to classify some of the existing literature based on their supported service model (or the types of service the network provides to its users). This view has helped us in understanding the underlying goals behind an architectural design. The high-level classification, which we refer to as the service-model perspective, is briefly discussed next.

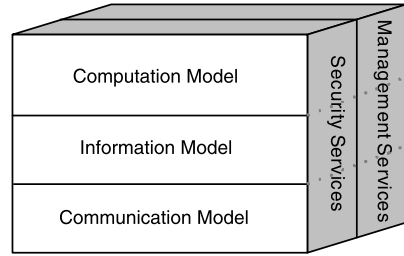
2.1.2 Service-Model Perspective

Classifying architectures from this perspective is motivated by several factors. First, the service-model approach implicitly accounts for the needs of the users relative to a network, which is the ultimate goal of any network design. For example, the Internet’s simple “best-effort delivery” service model came about to satisfy a set of goals, as explained in [49], primarily allowing multiplexed utilization of resources (which led to packet switching, domain, gateways), survivability (which led to end-to-end state), etc. Second, most network architectures tend to be naturally categorized and described relative to their service models. For example, we find in the literature the “data-oriented” network architecture [99], the “delay-tolerant” architecture [5], the “differentiated services (diffServ)” architecture [35], and so on. Finally, such a classification could enable future reasoning about—and evaluation of—the degree to which a particular architecture satisfies the service requirements of the users. One such evaluation methodology based on utility was proposed in [145].³

²Within the discussion, a *user* is the general term used to abstract any entity that utilizes the network services.

³In [145], Shenker defines utility as the degree to which a network service model matches the needs of the network users, i.e., how good an architecture is, is measured by the happiness of its users.

Fig. 2.1 Generalized service-model view



The generalized service-model perspective is depicted in Fig. 2.1. The communication, information, and computation⁴ models represent the building blocks that collectively define, together with the security and management services, the general service model of any network architecture. By building blocks we mean that every architecture must provide these three models, whether explicitly or implicitly.⁵ On the other hand, security and management services are not building blocks (since one can easily come up with architectures that do not provide any security or management services), and they operate across the communication, information, and computation models.

- *Communication model*: This service model represents the communication and control services offered by the network. For instance, delivery services whether “best-effort”, QoS-aware [35], aware of disruption [5], and/or geographic location [78] all belong to this model. Communication paradigms whether connection-oriented (e.g., ATM) or connectionless (e.g., SMDS, X.25) are classified under this model as well.
- *Information model*: This model deals with the information services that the network provides to its users. The networking community currently recognizes the need for network built-in information services (naming, searching, archiving, distribution, security, and analysis services) to support a multitude of applications and their requirements.
- *Computation or Programmability model*: This model represents the level of programmability support within the network. The programmability services might potentially span all the other service models, allowing for example the programmability of the communication model and/or the information model etc. Programmable networks [44, 155], for example, provide an explicit computation model.

The security and management services provided by the network are generally, but not necessarily, offered in-band with the rest of the service models. For example, secure communication services include secure end-to-end tunneling and transport (IPSec or SSL), secure identity (HIP [118]) etc.

⁴We abuse terminology referring to the terms computation and programmability interchangeably hereafter.

⁵For example, the Internet provides an information model implicitly (the datagram and information transparency) but not explicitly.

In fact, it is possible (and maybe convenient) to fully classify the literature based on the service-model view depicted in Fig. 2.1 if each of the constituent service models is further divided into its defining elements. The major focus of this section is to illustrate some of the prominent architectural work that represents critical points of the aforementioned service spectrum. Additionally, we believe that the independent contributions to the field are converging, and this section aims to highlight such phenomenon by means of a survey. Table 2.1 presents, in matrix form, such a survey of the literature, limited to general internetwork architectures. Hence, we do not consider overlays, scoped architectural work (such as naming, or routing architectures) and we do not provide an exhaustive list of internetwork technologies which is not the goal of this section.⁶ The work is divided into two parts. The first part overviews some of the independent contributions to the field, while the second part is solely concerned with the FIND [10] and FIA [9] work illustrating the community's view of what the future Internet should look like. More details about each of the architectures of Table 2.1 are provided in Sect. 2.2. Note that Table 2.1 marks the service models only as those are made *explicit* in the architectural description, and consequently it does not contradict our previous claim that communication, information, and computation models being building blocks. Several insights may be gathered by observing the matrix. First, there seems to be a growing consensus about the need for expanding the network's service model beyond the communication space, especially as researchers start thinking of designing a future Internet. Additionally, and most importantly, one can clearly notice the emphasis on information services, which is intuitive given the prevailing information-centric usage models with the current Internet.

The next section further elaborates on the service models of each of the architectures of Table 2.1.

2.2 Technical Reference

This section is intended as a technical reference for the rest of the book. We elaborate on the architectures of Table 2.1 as we refer to them frequently throughout the book.

2.2.1 Communication-Oriented

2.2.1.1 DARPA Internet

The design principles of the DARPA Internet [46] are clearly outlined in Clark's seminal paper [49] which we review here (see Chap. 1 for more history). The paper

⁶The majority of internetwork technologies (ATM, X.25, XNS, DECnet etc.) would be classified in our matrix as communication-oriented. For a comprehensive list of the internetwork technologies, we refer the reader to [48].

Table 2.1 Matrix view classification of internetwork architectures based on their explicit service-model classes, ordered by publication date

	Year	Comm	Info	Comp	Description
Independent proposals	ARPAnet [46, 49]	1970s	x		providing best-effort delivery of datagrams among globally identified attachment points
	Active Nets [155]	late 1990s	x	x	provide a framework for dynamic creation and deployment of network services at runtime
	TRIAD [77]	2000–1	x	x	exposing a “content layer” that provides transparent access and distribution of named content
	I3 [121, 149]	2002	x		providing a general-purpose rendezvous overlay that decouples host identity from location using flat, persistent names, and quasi pub-sub communication semantics
	Plutarch [54]	2003	x		provide a communication model that inherently allows interoperation of semantically disparate domains without mandating uniformity across them nor a global namespace
	FARA [50]	2003	x		provide an abstract network model that builds on the Internet’s “best effort” service model adding clean separation of endpoint names from network addresses
	NIRA [165]	2003	x		providing an Internet-style communication service that inherently supports user-selected domain-level routes similar to Plutarch service model, but with global naming
	TurfNet [140]	2004	x		providing an Internet-style communication service that utilizes flat labels for network identification and routing
	ROFL [42]	2006	x		providing data access (locating and retrieving data) and data distribution independent of location using an anycast primitive and flat naming
	DONA [99]	2007	x	x	

Table 2.1 (Continued)

	Year	Comm	Info	Comp	Description
Clean-slate: FIND [10] & FIA [9]	2006	x			establishing communications among user devices using persistent, user-relative names and namespaces that are constructed bottom-up with no reliance on global namespaces
	2006	x	x		providing reliable delivery (push/pull) of content (large data units or files) to mobile/stationary endpoints using in-network storage/caching
	2006	x	x		providing multipoint-to-point bulk data transfer/distribution among hosts (endpoints+intermediaries) with in-network storage/caching
	2006	x	x	x	providing information transfer in response to user (endpoint) specified service expressions through in-network processing/data handling
	2006	x	x	x	providing a network query interface to users for expressing intent and implementing operations through a declarative framework for managing in-network information and state (router and host state, and data streams)
	2006	x	x		provides a transient network substrate that enables identification and communication among entities based on global, and persistent (location-independent) identifiers
	2006	x		x	providing a tussle-resistant communication service, delivery of functional datagrams, that equips providers with usage control over their networks through policy enforcement, and users with policy-aware control over their traffic forwarding
	2006	x			providing a distributed location service and a spatial routing primitive for location-centric communication
	2010	x	x		providing content-centric publish-subscribe primitives that allows the network to natively match interests (subscriptions) to named content (publications) by name

highlights the connection between the intended goals of the DARPA Internet and design decisions that govern its current operation. The primary goal of the Internet was to allow multiplexed utilization of its resources, which influenced by the networks (ARPAnet) and technologies (packet switching) at that time, led to the adoption of the current Internet structure: domains, packet switching, and gateways connecting networks. The survivability requirement resulted in maintaining flow state at end nodes while keeping the core stateless with respect to ongoing flows. The requirement for alternative transport services in terms of latency and reliability has led to the TCP/IP split and the introduction of UDP. The need to support various networks resulted in making a minimum set of assumptions regarding the underlying function provided by the network which is “the network can deliver a packet or datagram”.

The Internet service model can be simply stated as providing *best-effort delivery of datagrams among globally identified attachment points*.

Communication Design Space

- *Topology*: The topological boundaries within the Internet are referred to as Autonomous Systems (AS) or domains. The domain is an authoritative boundary that maintains local communication policies. Domains are composed hierarchically through customer-provider or peering relationships with a distinguished core set of domains referred to as Tier1.
- *Addressing*: The Internet employs a global hierarchical addressing architecture, IPv4. The address space is composed hierarchically to match the topology, rendering the network address a topological forwarding directive. An IP address is formed of two parts: the network address and the host address.⁷ The address space is finite.
- *Naming*: DNS naming is an out-of-band service that is not part of the core network services, i.e., the network does not recognize a domain name. Hence, the Internet model does not constrain the naming architecture design space. Whether a centralized, global, hierarchical naming architecture (DNS style), or a distributed, flat naming architecture (e.g., OpenDHT) ends up being deployed is irrelevant to the Internet architecture as long as there exists a mechanism to translate a name to an IP address.
- *Routing and Forwarding*: The Border Gateway Protocol (BGP) [132] is the de-facto standard for inter-domain routing on the Internet. BGP is a policy-based path vector protocol which empowers domains with control over route selection and propagation. Forwarding is thus based on the policies of the domains and the protocol allows for arbitrary preference functions over routes. While the core forwarding function on the Internet is a best-effort delivery service, guaranteed delivery is provided as an end-to-end service using TCP.

⁷Classless Inter-Domain Routing (CIDR) separates the two parts at arbitrary boundaries using the forward slash notation as in 192.168.1.1/16. This means the first 16 bits of the 32 bit address, 192.168, designates the network address.

Information Design Space The only information abstraction recognized by the Internet architecture is the datagram. More complex information models must be composed out of the simple datagram. The type of content being delivered over the Internet’s best-effort service whether a static file or an time-sensitive voice stream is irrelevant to the architecture.

2.2.1.2 Plutarch

Briefly, Plutarch [54] proposes a framework for next generation networks that embraces heterogeneity within and handles it through contexts and interstitial functions. Contexts are like ASs that implement their internal addressing, naming, routing, and transport mechanisms. Interstitial functions map between the set of functionalities of different contexts. Plutarch is a response to the shortcomings of the current Internet protocol model that unifies all underlying network types through a one-size-fits-all networking and addressing mechanism. This (i.e., IP) has resulted in a semantic bottleneck that is leading to ossification of architecture where it is increasingly difficult to introduce anything but incremental changes. Network Address Translators (RFC 1631), and Resilient Overlay Networks [24] are examples of unclean solutions adopted as a result.

The service model of Plutarch, what users expect from the network, is not addressed in the paper. However, the aim of the work is to *provide a communication model that inherently allows interoperation of heterogeneous networks and mechanisms without mandating uniformity across networks (or a global namespace).*

Communication Design Space

- **Topology:** heterogeneous networks (contexts); **boundary:** a context is a region of the network that is homogeneous in some way, “a set of bindings with reference to which names may be resolved”; **Composition:** composition of contexts is not hierarchical, it is either adjacent (border) or containment (nesting); there is no notion of a global or unique “root context”; different namespaces can exist per context; no overlap of contexts.
- **Naming and Addressing:** local naming and addressing within contexts; inter-context name translation at gateways through “interstitial functions”; **Lookup:** route-query to chained-context-descriptor mapping (out-of-band); distributed route-query search across contexts (e.g., flooding).
- **Routing:** route over the discovered chained-contexts.
- **Compatibility with the Internet:** Plutarch subsumes the Internet as a context.

2.2.1.3 TurfNet

The proposal TurfNet [140] is a network architecture for future dynamically composable networks. The architecture is based on the concept of composing autonomous heterogeneous networks, referred to as turfs, dynamically without sharing

a global addressing mechanism or network protocol. Composition of networks takes two forms: (1) merging or horizontal composition, and (2) hierarchical independent or vertical composition. Within each AS (turf), independent addressing, routing and resolution mechanisms (control plane functionality) are available and are local to that turf. Across the turfnet, no common network protocol and no shared address space is required. However, a global name space is required to allow communication between the turfnet.

The service model of TurfNet is similar to Plutarch except for the designated global namespace.

Communication Design Space Separate naming/addressing:

- Topology: hierarchical ASs or turfs; composed horizontally (merging) or vertically (customer-provider/peering).
- Addressing: local address space per turf.
- Naming: Global namespace; Lookup: name to address mapping (in-band, i.e., lookup creates forwarding state), recursive lookup creates address and protocol translation state (soft state) within boundary routers up to root ASs.
- Forwarding/Routing: local forwarding within turf using local addresses; intra-turf: hosts external to the turf are mapped to turf address space; mapping soft state maintained by boundary gateways; end-to-end path composed of up-path to root (created during lookup) and down-path to destination turf (created through registration); routing = address + protocol translation at boundary gateways.

2.2.1.4 Internet Indirection Infrastructure (I3)

Internet Indirection Infrastructure (I3) [149] is a general-purpose indirection overlay network over IP that uses rendezvous servers where clients can register triggers using persistent names. It decouples the sending and the receiving actions where clients send traffic to the overlay and the latter takes care of forwarding the traffic to other interested clients that registered triggers in the system. A trigger is a simple binding of a persistent name to an IP address. In addition to mobility, several communication primitives are enabled as a result including unicast, multicast, and anycast by registering different types of trigger. Traffic in I3 goes through the overlay indirection servers thus degrading the routing efficiency in contrast to direct routing over IP, and potentially causing triangle routing and doubling the amount of traffic. The simple service model of I3 makes it vulnerable to a multitude of security attacks. The proposal Hi3 [121] is an attempt at making use of an overlay (I3) in conjunction with a secure direct end to end approach mainly the Host Identity Protocol (HIP) [118]. This would allow for efficient end-to-end traffic with HIP (mobility, multi-homing, DOS resistance) that leverages an independent secure, integrated rendezvous infrastructure (I3) as an overlay to route the HIP control traffic (stability).

The I3 service model may be stated as *providing a general-purpose rendezvous overlay that decouples host identity from location using flat, persistent names, and quasi pub-sub communication semantics.*

2.2.1.5 FARA

FARA [50] is an abstract network model in which the current Internet architecture is generalized and remodeled to enable clean separation of endpoint names from network addresses. The FARA abstraction recognizes communication among pairs of entities via logical links referred to as associations on top of a communication substrate. An entity in FARA is the endpoint of communication and smallest unit that can be mobile, such as a process, a thread, or a cluster of devices. An association is a logical communication link between a pair of entities representing persistent communication state. Entities maintain local association state and may have multiple concurrent associations. A packet belongs to one association and carries an association ID (AID) that enables the receiving entity to correctly demultiplex the packet to its association. As to the communication substrate, it represents underlying infrastructure that is able to deliver packets on behalf of associations. Addressing, routing, forwarding are mechanisms employed by the substrate and are left unspecified by FARA. However, FARA assumes connectionless point-to-point communication between entities. An entity supplies the substrate with a packet and header that contains a destination Forwarding Directive (FD). The latter contains enough information that the substrate can use to deliver the packet all the way to the destination entity that contains the association. Separating entities from the substrate can lead to seamless mobility of entities and independent evolution of applications and substrate mechanisms.

In M-FARA instantiation, and unlike the traditional Internet, no global address space is assumed, i.e., no IP addresses. However, there is a set of local address spaces referred to as “addressing realms”. Topological information is represented in terms of these private realms that a packet will traverse in transit towards its destination. The FD here is a set of sub-FDs specifying the realms on the path. In the case that the source entity and the destination entity belong to the same realm, the destination FD has NO topological information. When the source and destination belong to different realms, the destination FD must specify the topology which can be very complex if the private realm addressing is flat. M-FARA’s addressing introduces a 2 level hierarchy with a designated globally known “core”. Thus the destination FD in this case will consist of (FDup, FDdown) pair of FDs specifying how to reach the core from the source and then how to reach the destination from the core. This design supports mobility across realm boundaries.

2.2.1.6 NIRA

NIRA [165] is a novel inter-domain routing architecture that equips users with the ability to choose domain-level routes. The user control over the sequence of providers (ISPs) that packets take introduces competition among the backbone ISPs, thus driving innovation and lowering service cost. NIRA’s support for user choice introduces several problems including practical provider compensation, route discovery and representation, and security, which are addressed in the paper.

The service model of NIRA [165] may be stated as *providing an Internet-style communication service that inherently supports user-selected domain-level routes*.

Communication Design Space NIRA reuses many of the Internet's design decisions:

- Topology: strictly hierarchical ASs (domains) with distinguished core (customer-provider and peering relations); concept of domain hierarchy from edge user to core referred to as up-graph.
- Addressing: globally unique, hierarchical address space (IPv6 reused); address encapsulates topological location and provider hierarchy; (scheme: provider rooted hierarchical addressing).
- Routing: valley-free; unicast: 2-segment route (uphill + downhill), one from source to core (uphill) and one from core to destination (downhill); downhill discovered through name service; proactive detection of route failures; BGP-like inter-domain routing within the core, and path vector inter-domain routing over provider hierarchy.
- Naming: naming system required to map endpoint names to downhill route segment; name system design not mandated (may be DHT or DNS style for example).

2.2.1.7 ROFL

Routing On Flat Labels (ROFL) [42] addresses the challenge of how to scalably route on flat labels with no location identifiers. In contrast to the previous proposals, ROFL eliminates location identifiers altogether, and solely depends on persistent identifiers to route inter and intra-domain traffic. The advantages of such approach in addition to mobility and multi-homing include fate sharing (no control path since no need to contact resolution infrastructure), simple address allocation, better capture of identities at network layer, and independence from any external resolution systems.

The service model of ROFL [42] may be stated as *providing an Internet-style communication service that utilizes flat labels for network identification and routing*.

Communication Design Space ROFL eliminates the address dependence on location, hence the network address itself becomes the name.

- Topology: hierarchical ASs; AS up-graph required (domain's provider hierarchy).
- Addressing/Naming: Namespace: flat circular namespace similar to Chord [150]; Name semantics: unique persistent and global identifiers, semantic free as in [29], self-certifying (HIP [118] public key hash); Naming system: global DHT formation and maintenance as nodes join and leave.
- Routing/Forwarding: compact routing; no name/address resolution since routing on flat DHT; ID translated into source route during forwarding and route follows successor pointers; DHT formed over routers and static hosts.

2.2.1.8 Postmodern Internet

The postmodern Internet architecture [34] is a reaction to the rigidity of the current Internet's network layer with respect to different and possibly conflicting policies of stakeholders. The architecture aims to provide a minimalist network layer that anticipates tussle, and accommodates for it through flexibility in introducing policies. Users need to control how their traffic is delivered, while providers try to control their network usage. Some example policies include inter-domain routing policies, packet filtering policies, policies of who can specify forwarding and who has access to what, service policies, etc.

The service model of the architecture can be stated as *providing a tussle-resistant communication service, delivery of functional datagrams, that equips providers with usage control over their networks through policy enforcement, and users with policy-aware control over their traffic forwarding.*

Communication/Computation Design Space

- Topology: hierarchical virtual realms (trust boundaries).
- Packets and forwarding: functional datagrams (smart packets containing functional blocks—how, what, where, knobs and dials); user control over forwarding paths/directives (FDs) when aligned with provider policies; (mechanisms: Resolution from “destination specs” to LinkIDs to reach destination).
- Addressing and Routing: both decoupled from forwarding and not restricted (not part of network layer), i.e., no global addressing mechanism required; (mechanism: use globally unique Link IDs instead of node IDs to determine paths; inter-realm LinkID routing information dissemination).
- Transparency: in-network packet processing and rewrites of functional blocks.

Information/Security Design Space Functional packets recognized by network; network entities (hosts and realms) have trusted identities generated by decentralized PKI infra; signed packets; accountability and path signatures.

2.2.1.9 Geometric Stack

The proposal [78] calls for making geographic/spatial location information an inherent service of the network, for the latter to better address the needs of dense wireless/mobile access networks (geographic routing/addressing/tracking/dissemination). A novel stack is proposed to provide communication through physical space rather than network space (topology) enabling a multitude of applications that utilize geo-routing, geo-casting, and localization.

The service model of [78] may be stated as *providing a distributed location service and a spacial routing primitive for location-centric communication.*

Communication Design Space Packet-switched; location information available over some coordinate systems with translation among them:

- **Topology:** hierarchical topology; high-speed wired backbone with wireless edge networks; nodes associated with home areas.
- **Addressing:** address is geographic identifier (unicast: host ID + last position; geocast: set of coordinates of a zone).
- **Forwarding and Routing:** linear geometric routing (along 1-D paths called trajectories); source node specified path equation and network decides on forwarding through next-hop local forwarding by intermediate nodes (not source routing but path specification).
- **Naming:** host identifier (e.g., MAC derivative); **Lookup:** distributed service to resolve identifier to location (DHT based), location-based service.

2.2.2 Information-Oriented

The architectures we discuss in this section aim to achieve a similar objective, which is efficient network information support (typically in-band), a service that the current Internet model lacks. While sharing the general goal, the design decisions (and mechanisms) employed by each proposal remain different, which is what we try to illuminate next. There is a direct coupling between the information and the communication models, and the latter is generally intended to provide the necessary requirements for efficient data access (time, space, disruption, disaster, etc.).

2.2.2.1 TRIAD

The original Internet architecture provides transport mechanisms that are transparent to the applications or services employing them [49]. As the Internet usage models become more content-oriented (web traffic, multimedia, or p2p traffic), more intelligence is overlaid on top of the traditional Internet design to provide faster and more reliable content access as is the case with Content Distribution Networks (CDN). TRIAD [77] is a novel architecture that treats content as first-class, shifting the communication paradigm from host-centric to content-centric communication. The proposal exposes the limitations of current content distribution models, whether scalability, latency, or architectural openness and consistency. TRIAD addresses the content problem by making explicit a *content layer* that can efficiently *route* towards content. The content layer spans the core of the network by extending traditional IP routers to support *name-based routing*.

TRIAD's service model can be stated as *exposing a "content layer" that provides transparent access and distribution of named content*.

Information Design Space

- Type of information: datagrams/packets, services, and content.
- Naming: semantics: URL names compatible with DNS, URL split into domain name of content server and file name; service names are persistent whereas content names (service name + file name) is not; namespace: global hierarchical namespace; naming system: distributed naming infrastructure with no single point of failure.
- Routing: name-based routing: inter-domain content routers (CRs) route based on names towards content servers (caches for closest replicas, transformers); routing is a distributed in-band search operation; single-source multicast support.
- Tussle: Content routers (CRs) are provided as ISP infrastructure (similar to BGP routers now) and are thus controlled by the domain's authority, hence ISP control over directory service. Additionally, coupling naming and routing at the domain level can potentially lead to more tussle.

Communication Design Space Host-to-content; Delivery of content depends on the communication model (TRIAD reuses HTTP/TCP/IP transport). TRIAD does not mandate the Internet model though.

- Addressing and routing: Addressing is global hierarchical (IPv4 reused). Address is only a forwarding directive used for transient routing/forwarding of information and not for lookup.
- Naming and routing: There is an explicit separation between name and address/forwarding directive. All network entities (hosts) are identified by names (DNS). Name is end-to-end identifier, information model handles routing based on names.
- Transport: in-band lookup and connection setup/transport (using DRP as an alternative to TCP), hence symmetric search and delivery paths (lookup is as available as delivery).
- Mobility: Indirection-based host mobility, Name-based routing abstracts topological location, but since endpoints (hosts) do not advertise their names (for scalability), endpoint acquires new name in visiting domain and inserts a redirection in its home domain.
- Compatibility with Internet: highly compatible with Internet infrastructure, requires extensions at directory level.

2.2.2.2 DONA

The Data Oriented Network Architecture [99] is a clean-slate redesign of the naming and resolution mechanisms on the Internet. Similar to TRIAD [77], DONA is a reaction to the evolution of the Internet usage models. Initially, the Internet was designed to enable host-to-host communication (FTP, telnet where source explicitly specifies address/locator of destination) over a transparent forwarding engine. This

model has significantly changed since into a data-centric model where users access content and services independent of the location of content, services and of the users. DONA proposes replacing DNS names with flat, self-certifying names and replacing the name resolution mechanisms with a name-based anycast primitive that lives over IP. The main design requirements for DONA are persistence, availability and authenticity. The architecture itself is a synthesis of ideas from HIP [118], TRIAD [77], and SFS [114].

DONA's service model can be stated as *providing data access (locating and retrieving data) and data distribution independent of location using an anycast primitive and flat naming*. The design decisions of DONA are very similar to those of TRIAD, except for the naming architecture. We simply point the differences.

Information Design Space

- Naming: Semantics: persistent, self-certifying, flat name (HIP [118]).
- Tussle: ISP control over user's activity (content lookup and registration activity); ISP physical control over Resolution Handlers (which are similar to content routers in TRIAD).
- Security: Authentication and integrity of information (mechanism: PKey cryptography).

2.2.2.3 USwarm

Universal Swarm (USwarm) [161] is a proposal that applies swarming techniques (such as BitTorrent) to design a universal data transfer architecture that learns from p2p architectures and eliminates selfishness of peers through incentives. The architecture is a response to the data transfer shortcomings on the current Internet. The whole Internet is modeled as a single swarm that employs a distributed metadata resolution system to resolve data-to-peers(s) that can serve the data (analogous to BitTorrent Tracker). An intentional naming system is employed as well to resolve intentions to metadata. Hence 2 level resolution is proposed: intention to metadata to provider peers.

The service model of USwarm can be stated as *providing multipoint-to-point bulk data transfer/distribution among hosts (endpoints + intermediaries) with in-network storage/caching*.

Information Design Space

- Type: datagrams, data, metadata.
- Naming: Bittorrent model—semantics: self-certifying (publisher, hash) tuple, principal name globally unique and authentic, metadata uniquely specifies data and contains name plus block ids; infrastructure out-of-band intentional resolution system IRS (map intent to metadata); distributed p2p lookup/search.
- Routing/search: locality-aware distributed tracking that involves peers and intermediaries to locate content (i.e., resolve metadata to peers); more control over routing decisions for ISPs (traffic engineering) and users.

- Transport: multipoint to point transport of information; incentive-aware.
- Security: authenticity and integrity of data.

Communication Design Space Internet style point-to-point communication employed.

2.2.2.4 UIA

The User Information Architecture (UIA) [11, 63] is a clean-slate redesign of the naming and resolution mechanisms on the Internet. The current Internet reliance on global naming makes it hard for two devices to communicate even when they are in close proximity. As a result, instead of relying on global namespaces and infrastructure-based connectivity, UIA constructs user-local namespaces bottom-up and embraces ad-hoc connectivity for communication among user devices. Users assign personal names to their devices and to other devices they meet. These user-specific, persistent names constitute the user's namespace and need only be unique within that user's namespace. UIA provides mechanisms for namespace creation, name resolution, and local and remote connectivity among the users' named devices. Communication could rely on ad-hoc or infrastructure-based networks depending on the physical paths available between the named devices.

The service model can be stated as *establishing communications among user devices using persistent, user-relative names and namespaces that are constructed bottom-up with no reliance on global namespaces.*

Communication Design Space

- Naming: devices are identified with flat cryptographic identifiers (EIDs); users assign persistent, user-relative names to their devices; the user-relative names are bound to the device EIDs and form the user's namespace; each user's namespace contains user-relative names/bindings of the user's devices and of other users' devices that are merged with the user's namespace when devices are introduced; the user namespace is organized as a tree with the user as its root and other users' spaces forming subtrees.
- Routing: several options are proposed for routing on EIDs whether using ad-hoc gossip-based discovery protocols and routing for devices on a local network, or using the Internet for remote connectivity, or even using clean-slate DHT-based routing similar to other identity-based routing proposals; given that users, devices, and user-relative names are the main currency of UIA, and given that devices are linked following social introductions, UIA proposes using social structure to expedite routing on EIDs.
- Security: EIDs are cryptographic hashes of public keys; name-based authentication and access control using distributed PKI infrastructure; security is built bottom-up to follow the namespaces.

2.2.2.5 Postcards from the Edge

The proposal “Postcards from the Edge” [166] is a clean-slate cache-and-forward architecture for a future internet. The architecture is a response to the revolution in access technologies, primarily wireless and mobility, that overwhelm the Internet’s basic design assumptions. Given the continuously increasing capacity and decreasing cost of in-network storage, the authors propose an architecture that provides uniform reliable transport of large files across heterogeneous access networks and in the face of intermittent connectivity.

The service model can be stated as *providing reliable delivery (push/pull) of content (large data units or files) to mobile/stationary endpoints using in-network storage/caching*.

Information Design Space

- Type of info: large data units (files).
- Naming of content: Semantics: globally unique (UFID.FQDN) names for files; name service: out-of-band service (i.e., lookup and transport are not coupled), hierarchical name resolution system (DNS style); Lookup: distributed name-to-cache(s) mappings.
- Routing/search: no routing based on names; out-of-band search/lookup to name service (i.e., lookup name then contact host similar to Internet); Rendezvous push/pull through post-office nodes (every node knows PO current PO nodes where former can pick its content).

Communication Design Space Host-to-host; builds on top of Internet best-effort IP service for addressing and routing of control traffic. IP is not essential to the design though.

- Topology: hierarchical topology formed of high-speed wired backbone connected to edge access networks.
- Naming of endpoints: location aware rendezvous service; Lookup: distributed name to post-office mapping; Security: endpoint associated with home autonomous system for authentication when mobile.
- Routing: hop-by-hop routing on location information; Supports Type-of-Service (TOS) to distinguish between transport and caching.
- Mobility: artifact of rendezvous.
- Compliant with Internet: builds on IP, hence highly compliant.

2.2.2.6 Information Transfer and Data Services

The proposal ITDS [164] is a response to the simple store-and-forward model of the current Internet. The architecture calls for in-network support of a broader range

of services by dealing with information abstractions⁸ rather than simple bit transfers. Hence the communication model proposed is aware of information rather than simply being a bit-plumbing medium. Additionally, in-network data processing is proposed to implement dynamic user service requirements.

The service model of [164] may be stated as *providing information transfer in response to user (endpoint) specified service expressions through in-network processing/data handling*.

Information/Data and Computation Design Space

- Type: data, information; semantics: various transfer characteristics (streaming, random access, interactive, canned, ...).
- Processing: general-purpose computation model on routers to support data services; data services already offered by network and not dynamically deployed by users (contrast to active nets for example); service specifications are mapped to computation model rather than dynamically introduced; service composition, control.
- Storage: in-network storage/caching possible on routers.

Computation Design Space User specifies intent and network maps computation to resources, hence, limited user control.

Communication design space Can operate on top of various communication models (e.g., Internet); ITDS focuses on information transfer models “on top of the network”.

2.2.2.7 WiKi

The proposal Wireless Knowledge Infrastructure (WiKi) [39] addresses the limitations of the current Internet in supporting the needs of applications and services given the huge proliferation of wireless, mobile, and ubiquitous computing. WiKi takes a clean-slate approach to designing a future declarative network in which in-network state (router, network, and host state) is treated as distributed data that can be queried by users through declarative languages. Such approach separates logical representation from actual implementation making the network more flexible to change, and more informed about its operations. Hence, WiKi realizes a knowledge plane [51] for the Internet.

The service model of WiKi may be stated as *providing a network query interface to users for expressing intent and implementing operations through a declarative framework for managing in-network information and state (router and host state, and data streams)*.

⁸The proposal explicitly differentiates between information (e.g., “the requested web page did not change”) and data/content (e.g., the actual web page data).

Information Design Space

- Type: internal network information (state), data streams, continuous/static queries; cross layer, cross domain views of data streams; archives.
- Processing: integration, aggregation, fusion, joins, etc.
- Security: policy enforcement; distributed monitoring; access control.

Computation Design Space Built-in distributed WiKi runtime (proxies); query processing and optimization; user control: user has control over computation (e.g., route selection/protocol, service composition) through queries (mechanism: Network Datalog language to specify computation).

Communication Design Space Wireless/mobile endpoints query WiKi proxy nodes (infrastructure or overlay) that perform query processing; can operate as overlay.

2.2.2.8 NDN

Motivated by the content-centric usage models on the Internet, content-centric networking [90] or Named Data Networking (NDN) [16] by Jacobson et al. recognizes content, instead of hosts, as a first-class network citizen. The NDN architecture introduces new content-centric primitives into the Internet's core; primitives for content discovery, retrieval, and caching allow for more efficient, secure, and location-independent information sharing. NDN re-architects the protocol stack to route interest packets towards named data objects. Interest packets are routed by name using simple name-aware forwarding and caching primitives.

The service model of NDN may be stated as *providing content-centric publish-subscribe primitives that allows the network to natively match interests (subscriptions) to named content (publications) by name.*

Information Design Space

- Objects: two types of packet, interest packets (subscriptions) and data packets (publications); data packets consume interest packet; matching on longest prefix name as in IP.
- Naming: content names are multi-part and hierarchical; names are globally routable and are aggregated by prefix in a manner similar to IP for scalability.
- Security: to provide integrity and self-authentication, each content packet is individually signed along with its name binding.

Communication Design Space

- Routing: can operate as an overlay on top of IP; only interest packets are routed towards the content provider, while content is symmetrically forwarded on the reverse path; supports multicast primitives and disconnected semantics.

2.2.3 Computation-Oriented

The most notable work on programmable networks is perhaps the Active Networks. A detailed survey on active networks is provided by Tennenhouse [155], and a more general one on programmable networks is provided by Campell [44]. The area of active networks has extensively explored the idea of programmable networks with the ability to introduce change into networks dynamically. Some pointers to prominent work in this field can be found at DARPA's site <http://www.sds.lcs.mit.edu/darpa-activenet/>, and at the IEEE issue [3, 12]. This section reviews an example of programmable networks instead of surveying the literature.

In [156], the authors introduce the concept of programming the network by shifting the computing paradigm on current networks from an end-to-end system to a system in which each fine-grained data element participates in the computation. Within active networks, traditional packets are replaced by “capsules” which contain programs and content simultaneously. Nodes on the network (routers, switches, servers—firewall) can dynamically execute the capsule programs safely and efficiently. The capsules are loaded into a transient environment on the node and are safely executed/interpreted eliminating the ability of the capsule programs to stray beyond the restricted environment and thus to compromise the shared resources on the network. The programming abstraction provided by active networks allows user-driven customization of the infrastructure to enable faster deployment of new services. It also enables for adaptive protocols on the network, thus tailoring the infrastructure for user/application needs. Logically, active networks shift the intelligence in the network from the node to the capsule. A capsule for example will compute its path within the network (might need to access routing tables on nodes) instead of the capsule (packet) being dumb. PLAN [85] is a functional programming language for active networks used within packets. It is resource limited and secure enabling a smart means of communicating between nodes. PLANet [84] is an active network implementation that utilizes PLAN to implement network layer on top of an IP-free link layer. PLANet uses active packets and allows active extensions, used to operate the network, to be downloaded to routers (for example to implement services like DNS, address resolution, routing). An active packet within PLANet needs to explicitly specify the destination of evaluation, avoiding the evaluation of the packet program on every node on the packet's path. This is mainly due to the reason that evaluation is computationally expensive. Programs in packets are marshalled at the source node and unmarshalled only at point of evaluation. In general a packet is only forwarded by an intermediary node (by executing an *routeFun* attribute specified by the packet) until it reaches the intended destination on which it is evaluated. Addressing within PLANet is based on 48 bit addresses (implemented with 32 bit IP appended to 16 bit port number) assigning one address per network interface on a node.

2.3 Conclusion

This chapter presented a survey of novel network architectures. We have shown that while the communication structure is necessary for defining and representing a modern network architecture, it is in general insufficient. Information and computation structures are other building blocks that need to be properly understood within modern networks. We highlighted naming as a core service within an inter-network architecture and we reviewed the basic objects and abstractions involved. As we shall see in the next chapter, most of the naming architectures are composed of simple underlying objects and primitives.



<http://www.springer.com/978-1-4471-4551-6>

Internet Naming and Discovery

Architecture and Economics

Khoury, J.S.; Abdallah, C.T.

2013, XIV, 142 p., Hardcover

ISBN: 978-1-4471-4551-6