Heterogeneous Access: Survey and Design Considerations

Amandeep Singh[†], Gaston Ormazabal[†], Sateesh Addepalli[*], Henning Schulzrinne[†]
[†]{aman, gso, hgs}@cs.columbia.edu
[*]sateeshk@cisco.com

***Abstract*—As voice, multimedia, and data services are converging to IP, there is a need for a new networking architecture to support future innovations and applications. Users are consuming Internet services from multiple devices that have multiple network interfaces such as Wi-Fi, LTE, Bluetooth, and possibly wired LAN. Such diverse network connectivity can be used to increase both reliability and performance by running applications over multiple links, sequentially for seamless user experience, or in parallel for bandwidth and performance enhancements. The existing networking stack, however, offers almost no support for intelligently exploiting such network, device, and location diversity.**

In this work, we survey recently proposed protocols and architectures that enable heterogeneous networking support. Upon evaluation, we abstract common design patterns and propose a unified networking architecture that makes better use of a heterogeneous dynamic environment, both in terms of networks and devices. The architecture enables mobile nodes to make intelligent decisions about how and when to use each or a combination of networks, based on access policies. With this new architecture, we envision a shift from current applications, which support a single network, location, and device at a time to applications that can support multiple networks, multiple locations, and multiple devices.

***Index Terms*—Heterogeneous access, pervasive mobility, multihoming, multipath, disruption tolerance, policy management**

I. Introduction

People have become more connected than ever. Mobile communications and the Internet are at the center of this phenomenon. With more mobile devices connecting to the Internet, the bandwidth consumption of mobile broadband will soon surpass fixed line broadband [1]. Users expectations to communicate in this always connected world will only increase in future. Let us see how Alice would use Internet services in this more connected world of the future – Alice is on a video conference call with Bob on her office phone. It is getting late, she leaves her office and transfers the call to her mobile phone. She takes the subway to her home, there are no interruptions in the call when the subway is underground. Bob shares an important document with Alice and asks her to review it, as soon as Alice reaches home she transfers the video conference to her Internet enabled TV and the shared document to her desktop. In this example, we have seen how Alice wants to communicate with Bob without any interruptions across networks and from multiple devices. To make Alices call possible, we envision a shift from current applications which support a single network, location, and device at a time to applications that can support multiple heterogeneous networks, locations, and devices seamlessly.

The latest communication devices have significant mobility and processing power that provides phone, computing, video, and data communication all based upon IP protocol, for example, smartphones, tablets, and laptops. These devices have multiple network interfaces, such as Wi-Fi, WiMAX, LTE, and possibly wired LAN. Cellular networks provide pervasive mobility with a large coverage area as compared with other networks like Wi-Fi which only provide limited coverage. Also, there are multiple service providers in the same geographic area in most of the regions around the world. The devices may experience sequential connectivity across technologies or multiple concurrent networks. Despite this multiplicity of networks and devices, protocols are largely still assuming the use of a single network interface and address for mobile devices. For example, mobile IP is predicated on the idea of maintaining a single network address across network attachment changes. The HTTP protocol also assumes a single destination address at a time. In fact, all applications based on the standard socket API only support one network interface at a time. For an efficient use of network resources across multiple network technologies there is a need of an unified network architecture where network entities (NEs) and mobile nodes (MNs) are working together to provide high throughput, resiliency, better energy management, and above all a seamless user experience.

Multi-homing refers to the ability to connect to multiple networks at the same time. It provides *network* independence and offers fault-tolerance (reliability), flow redirection, and load balancing functionalities. When a single connection can be spread over multiple interfaces of paths it is defined as multipath connectivity. Multipath provides on-demand bandwidth, and also, fault-tolerance. Internet applications can make better use of these multiple networks by specifying policies for network usage especially for applications which require on-demand network resources. For example, to reduce time and cost, a mobile device can use a cheaper Wi-Fi network for downloading a file instead of using an expensive LTE connection, which can be saved for time critical applications like voice calls.

Mobility can be terminal, network, session, or personal [3]. Terminal mobility provides *location* independence with continuous network access when a MN moves from one network to another. Network mobility provides efficient network access in

transportation industry, for example, Wi-Fi enabled cars, buses, trains and more recently, aircrafts. Having mobility support in a heterogeneous environment results in better connectivity, and cost savings for both the users and the service providers. Figure 1 illustrates a heterogeneous mobile environment where a MN can connect to any available network based upon its location.

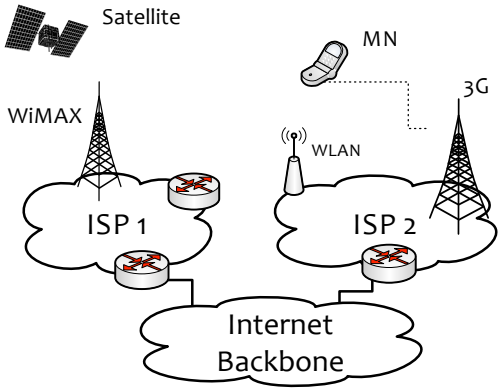

Fig. 1. Heterogeneous Environment

Increasingly, users are consuming rich Internet services such as Skype, Netflix from multiple devices such as smartphones, tablets, laptops, and more recently, connected vehicles. Users may want to move current active application from one device to another, and continue using the service from the second device. But, application clients are not truely mobile, as TCP connections are reset whenever there is a change in the client IP address. Session and personal mobility [3] provides *device* independence by supporting a single application context across multiple devices. For example, Alice is watching an online video on her smartphone, she can seamlessly transfer the application session along with her credentials to another device and resume the video where she left it.

Having mobility can cause temporary disruptions due to reduction in radio signal strength, physical channel congestion or network unavailability. Users can move around places where there is no wireless Internet access at all or the user cannot readily access any of the options, for example, while traveling internationally. Thus, connectivity can be intermittent with disruptions from seconds to hours or days. Delay tolerant networks (DTN) protocols provide communication support in mobile ad-hoc networks and sensor based networks where continuous end-to-end connectivity may not be possible [17]. Having DTN support in the future Internet architecture will provide better resiliency and user experience. Furthermore, DTN support can enable home devices to communicate with each other in an opportunistic manner to share data, and resources in an energy efficient manner [22].

Several solutions have been proposed in the last decade, which decouple location and identifier of end nodes to support multihoming and mobility functionalities. In this work, we propose a unified mobile Internet architecture design by understanding requirements for each networking stack layer, which transitions the current networking stack into the mobile era by providing multihoming, mobility, multipath, disruption tolerance and user control support.

In this work, we first summarize and evaluate presently available relevant solutions in view of our goals, and propose a architecture design that is required to achieve the vision of a unified mobile Internet. In Section II, we identify the design goals that are essential for our vision. In Section III, we survey some of the important proposals related to future Internet architecture. We restrict ourselves to current Internet protocol suite enhancements leaving out the clean-state Internet achitecture approaches [2]. In Section IV, we propose our initial design of the architecture. Section V discusses the future vision of the proposed architecture.

II. Design Goals

As all voice, multimedia, and data services are converging to IP, a generalized approach is required to understand mobility communication requirements impacting application users, developers, and network providers. The proposed architecture design goals include:

1) **Seamless user experience:** Users should not experience any degradation of service when a change in network point of attachment, network type, service provider or device occurs. Real-time applications, such as an active voice or video call, should not experience any disconnections or delays upon handover, either from one network service to another (e.g., LTE to Wi-Fi), or upon changing devices (e.g., from a mobile phone to a laptop).
2) **User and mobile node independence:** Users and their mobile nodes (MN) should be decoupled from physical network infrastructure. Networks are heterogeneous in terms of access technologies (Wi-Fi, LTE, WiMAX, satellite), service providers, performance and cost. Users can access Internet applications from many locations (home, office, outdoors), devices (PC, smartphone, tablet) and from more than one network at a time.
3) **Policy management:** Users should be able to control their network access. Application developers should be able to select either static (pre-configured) or dynamically configured network access policies. For example, a user may decide to download OS upgrades, or upload videos, when network bandwidth is plentiful or cheap. Conversely, voice or video services, or interactive games, may require low-latency mobile connections.
4) **Multi-device support:** MNs should support multi-device communication sessions. MNs should be able to discover neighboring nodes, and their corresponding shared services, such as, audio, video (camera, display), and storage, in a secure manner, to enable ubiquitous computing for a next-generation multimedia experience. For example, a user can transfer a video feed from a mobile device to a fixed TV, while maintaining security associations, and application context.
5) **Secure communication:** There should be a well-defined security model for connection (resource access), context

(protocols), and content (data) abstractions without introducing additional latency. For example, cryptographic credentials can be reused to reduce redundant operations.

6) **Disruption tolerance:** Applications should continue to operate in the absence of network connectivity both short and long term, to provide resiliency and better user experience. For example, a file transfer to and from a cloud-based storage service can be controlled by user-defined policies that take into account network conditions and priority, delaying packet delivery until a suitable network becomes available.
7) **Network intelligence:** MNs should leverage network-based resources seamlessly, for making better network selection decisions, and storing data packets for future deliveries. These resources may provide additional computation and storage, helping MNs to reduce energy consumption. For example, a service can store a users daily route geographic map of nearby networks, and based upon usage-pattern analysis, the service can delay data transfers, until either a high bandwidth, or a low cost network, becomes available.
8) **Backward compatibility:** Existing legacy applications should work without any modifications. The standard networking socket API should be adhered to, since it is the most programing interface.

The overall objective of the proposed architecture is to improve the quality and continuity of the mobile user experience. With these comprehensive goals, a shift from the traditional network-centric to a user-centric approach, in network architectures, is implied. In this architecture, users and applications are able to determine what kind of network connectivity they want to use in terms of availability, resource consumption, bandwidth, cost, and QoS.

III. Related Work

In the past decade, various proposals have been made to improve present functionalities of Internet networking stack. We believe providing multihoming, general mobility - terminal, network, session and personal, multipath, and disruption tolerance in a secure manner are the core functionalities that need to be addressed by any new Internet achitecture. To understand previous work, we perform a layer by layer analysis of presently available solutions starting from application layer. In view of our design goals, we evaluate each solutions support for mobility, multihoming, multipath, disruption tolerance, and policy management. For transport and network layer support, we will see how almost all proposals split the overloaded IP address with the locator/identifier [4] abstraction. For data link layer support, we will see a similar kind of abstraction provided by IEEE 802.21 Mobile Independent Handoff (MIH) [60] framework. For physical layer, Dynamic Spectrum Access (DSA) solutions are discussed for efficient spectrum access.

Later we will discuss three different integrated architectures, namely, Architecture for Ubiquitous Mobile Communications (AMC) [75], and OpenRoads architecture [77]. We will see how these proposals decouple MNs from physical network infrastructure to provide a service layer abstraction. In the last section, we will evaluate these proposals in view of our initial design goals.

Application Layer

Application layer support is relatively easy to deploy compared to other layers as it is mainly installed as a user-space component in an OS. We will discuss general mobility using Session Initiation Protocol (SIP) [5]. Application frameworks for ubiquitous computing, and multihoming and multipath support using session shim-layer solutions. In the last section, we will discuss disruption tolerance support in application layer.

A. Mobility

Session Initiation Protocol (SIP) implements location/identifier split at the application layer where clients, user agents (UAs), register their location with registrar server, which acts as a rendezvous point. SIP supports terminal, session, and personal mobility [3]. Session mobility defines session/context transfer from one physical device to another. SIP provides session mobility using two methods, third-party call control (3PCC) and the REFER method [3]. Dutta et al. [6] implement a SIP based application layer mobility architecture in heterogeneous wireless networks for both real-time and non-real-time communications.

Shacham et al. [7] present a ubiquitous device personalization architecture based upon SIP. A new model called *room presence* is proposed, where each discrete location such as a office room or home has its own presence and all users currently occupying it register their services using the Service Location Protocol [8] with a centralized Directory Agent (DA). For personalization, the architecture defines a profile document that lists a set of rules that apply for any room in which the user wants the devices to be configured. When a user enters a room, the local devices that are authorized to see the user profile, configure themselves according to the user profile. It also introduces a new multi-device system (MDC), which is essentially a virtual device created through joining the features of two or more existing devices registered as one single device in SLP DA. All sensible combinations of virtual devices are formed by booting a Multi-Device System Manager (MDSM) that searches for devices in its local vicinity.

Hansen et al. [9] propose a session mobility solution for application migration using a SOCKv5 [10] proxy. It defines two new network elements, a Migration Server (MS) for client services registration and migration management, and a Mobility Anchor Point (MAP), a modified SOCKv5 proxy present in the network data path between the client and the other endpoint of the communication. When the client device (source) wants to migrate an application to a new device (destination), the MS starts the migration on both the devices and MAP changes the tuple $\{$application ID, device ID$\}$ from source endpoint to the new destination endpoint.

The Transparent Extensible Session Layer Architecture (TESLA) [11] introduces a user space shim sub-layer be-

tween application and transport layer to provide session-layer services like application-controlled routing and traffic engineering. In case of disconnections and mobility events it maintains the end-to-end connection by maintaining the same initial IP address even after a new IP address is acquired. Additionally, it can also provide optional encryption functionality. TESLA does not modify the host networking stack as it provides application transparency by dynamically extending the protocol suite using dynamic library interposition. But, it requires both hosts to use same configuration of the TESLA protocol stack. Figure 2 illustrates the general session shim sub-layer functionalities. We observe that session shim sub-layer requires both hosts to have similar networking stack to support policy management, mobility, multihoming, and in some cases, application security. There is no standard session intialization mechanism for configuration exchange, which makes the deployment of session shim sub-layer solutions a challenge. For example, if one host is configured to use encryption the correspondent node must be configured to decrypt the payload.

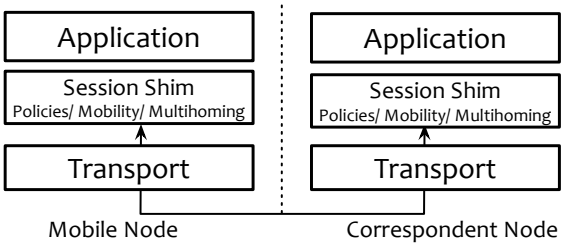

Fig. 2. Session Shim Sub-layer

A number of projects have investigated ubiquitous computing by introducing a new middleware layer on top of which new applications can be implemented. The Panoply [12] project introduces a model based on *spheres of influence* where each sphere encapsulates policy and provides well defined boundaries for interactions in a secure manner. Similarly, the Gaia (Active Spaces) [13] project also creates a component-based middleware layer that provides a dynamic services framework to applications to communicate in a ubiquitous environment. However, these solutions do not consider multihoming and disruption tolerance as their design goals.

B. Multihoming

Multihoming support in application layer can be implemented independently by applications, which can make them complex to manage. Also, it can result in performance degradation due to lower layers blocking simultaneous connections. To provide a generic framework, middleware architectures provide networking APIs [15], [16] to support application policies and multihoming support. Strawman [15] architecture implements a kernel-space session layer to provide individual flow striping to improve the performance of applications. It allows striping over multiple connections, maximum throughput, and minimum delay. However, it does not support mobility natively.

C. Disruption Tolerance

Often applications handle network timeouts as communication errors. Ott [18] argues that a future Internet architecture should be designed with challenging networking conditions – disconnections and delays, as a part of the design rather than considering them as network errors. For enabling DTN [17] support, an incremental deployment approach using network proxy, overlay network, and delay-tolerant forwarding as a native routing and forwarding infrastructure is suggested.

The CHIANTI [19] project proposes a modular proxy-based architecture for supporting legacy applications in the presence of intermittent connectivity and changing connectivity characteristics. The CHIANTI architecture defines a tunnel protocol to communicate between the client and the proxy. The tunnel protocol carries application protocol (with optional enhancements) packets on top of IP and transport layer. Different protocols can be used as tunnel, for example, DTN bundle protocol over TCP/UDP [20]. To further improve latency, failover and scalability of proxies, Opportunistic Connection Management Protocol (OCMP) [21] architecture proposes sharing of client state among proxies in an overlay network.

To share data and services in a disconnected network, mobile users have to rely on node and service discovery methods. Moghadam et al. [22] present a modular application development framework, Seven Degrees of Seperation (7DS), to support opportunistic communication in a disconnected dynamic environment. 7DS provides the necessary transport and application layer functionalities for mobile nodes to exchange information in store-carry-forward manner. Intentional Networking [23] provides custom socket APIs to enable application specific communication policies (intentions) and disruption tolerance support in heterogeneous mobile networks.

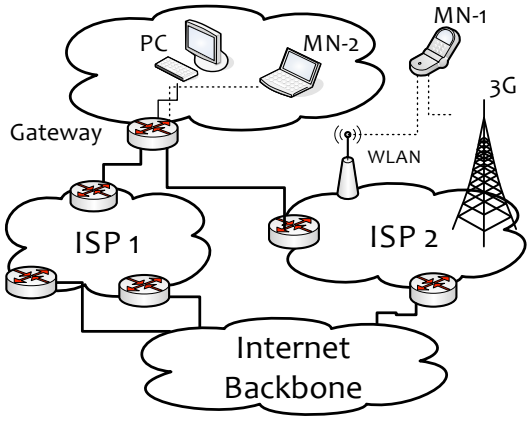

Fig. 3. End-host vs End-site

TRANSPORT AND NETWORK LAYER

Transport and network layer supports are implemented in kernel-space, which has a system wide effect. Transport and network layer solutions can be divided into end-host and end-site based upon the location of changes in networking stack. End-host solutions require host networking stack changes and

they do not require any network infrastructure changes. End-site solutions are deployed at the site's exit routers enabling multihoming and network mobility support for the whole site. Figure 3 illustrates end-host vs end-site multihoming and mobility support. End-site with Desktop PC and MN-2 can access a multihomed network, and end-host – MN-1 manages mobility and multihoming independently.

End-host Solutions

End-host solutions require host networking stack changes. For the transport layer, we will look at Multipath TCP (MPTCP) [25] and Stream Control Transmission Protocol (SCTP) [31]. For network shim sub-layer solutions, we will look at Host Identity Protocol (HIP) [38], End-to-End Connection Control Protocol (ECCP) [49], Site Multihoming by IPv6 Intermediation (SHIM6) [50], Identifier-Locator Network Protocol (ILNP) [54], and Mobile IPv6 [57].

Transport Layer

A. Multipath Transmission Control Protocol

The MPTCP [25] protocol is an extension of TCP that allows simultaneous use of different paths by resource pooling [29] their capacities into a single connection resulting in higher throughput and fault tolerance. MPTCP provides a dynamic locator/identifier abstraction without requiring any additional support from network. Initially, both hosts establish a basic connection and exchange their MPTCP capabilities. If multiple addresses are present, additional sub-flows can be added to the already established connection. Each sub-flow handles congestion control independently, and data packets are distributed between the sub-flows according to the available bandwidth on each path.

The MPTCP protocol supports mobility in an opportunistic manner [26] without requiring any new network entity. After establishing initial connection, if connectivity can be achieved using another interface, a second sub-flow will be established. If connectivity is lost for one sub-flow, the remaining sub-flows can continue without interruption. A single MPTCP connection can simultaneously communicate using both IPv4 and IPv6 addresses, so it is possible for a mobile host to move seamlessly between IPv4 and IPv6 networks.

The MPTCP proxy can be used to enable MPTCP hosts to connect to legacy hosts. The MPTCP proxy acts as a fixed anchor point associated with the mobile host, which supports MPTCP. Major improvements to current MPTCP design have been proposed for wireless access networks [27], which can enable user policies and traffic engineering capabilities. Besides multihoming and mobility, MPTCP can reduce energy consumption in mobile devices [28] by selecting more efficient network path for data transfers.

B. Stream Control Transmission Protocol

The SCTP [31] protocol is a connection oriented reliable transport protocol that supports multihoming natively. Like user datagram protocol (UDP), SCTP uses messages (chunks) for communication. It employs two different types of chunks, data chunks to transmit actual data and control chunks (HEARTBEATs) to monitor peers and path status. Upon establishing a connection between two hosts, a SCTP association is created. Multiple paths can exist in a single association. SCTP uses a primary path to transfer data and other secondary paths are used for fault tolerance. SCTP supports multi-streaming – ability to send independent streams of chunks in parallel inside a single association, which increases the availability and avoids head-of-line blocking [36]. In case of any transmission errors in the primary path, SCTP can automatically change the data transmission path to one of the secondary paths. Concurrent Multipath Transfer (CMT) [32] extension adds support for using multiple paths for data transfer in SCTP.

The SCTP Dynamic Address Reconfiguration (SCTP-DAR) [34] extension adds mobility support, which allows each endpoint to update the IP address list for an existing SCTP association. This update procedure typically happens during the handover of a mobile terminal since one interface is used at a time. For supporting multipath data streams, mobile Concurrent Multipath Transfer (mCMT) [35] can be deployed. SCTP has support for multihoming, mobility and multipath, but, it requires NAT update [37] to work properly inside a NAT-enabled site, which makes the depolyment a challenge.

To take advantage of SCTP both sides need to support it. On the other hand, MPTCP can support one-side implementation as multipath functionality is used only when initial handshake is complete. Additionally, existing applications require changes to use SCTP while MPTCP is an extension of TCP requiring no changes to existing applications. SCTP also suffers from middlebox issues – firewalls may not allow SCTP traffic.

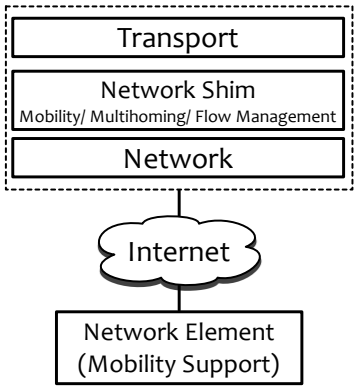

Fig. 4. General Network Shim Sub-layer based Architecture

Network Shim Sub-layer

This section discusses several proposals, which introduce a new shim sub-layer between network and transport layer. The shim sub-layer provides multi-homing, mobility, multipath and flow management functionalities. For supporting mobility (terminal and network) these solutions introduce, in general, a new network element or propose enhancements to existing ones (e.g. DNS). Figure 4 illustrates a general network shim

sub-layer based network architecture and the supported functionalities.

A. Host Identity Protocol

The HIP [38] protocol is designed to support mobility, multihoming, and baseline end-to-end security natively. HIP decouples the transport layer and IP layer (IPv4 and IPv6) by introducing a new host identity layer between them. The host identity layer introduces a new namespace, host identity, which is used by the transport layer to connect to other nodes in the network and the traditional network layer IP addresses (locators) are used for routing. Each host can support multiple identities but each identity is uniquely defined by an identifier – a public key of an asymmetric key-pair, which can be used for host authentication as well. From a functional point of view, HIP integrates IP-layer mobility, multihoming, security, NAT traversal, and IPv4/6 interoperability in a novel way [39].

The Host Identity Hash (HIH) function is used to reduce a public key into a static IPv6 compatible 128-bit Host Identity Tag (HIT). In order to support IPv4 applications, HIT can be further reduced to a locally unique 32-bit Local Scope Identifier (LSI). HIP uses Encapsulated Security Payload (ESP) [42] to carry data packets in each direction. HIP clearly separates control and data protocol signalling, which results in a clean abstraction and multiple data protocol support. For identity resolution HIP can work in three different modes – opportunistic, using distirbuted hash tables (DHTs) [40] or new resource record for DNS [41].

Since IP addresses are used only for routing the packets they become irrelevant after the packets have reached the destination interface. Therefore, HIP can support mobility and multihoming by controlling what IP addresses are placed in outgoing packets. The host uses UPDATE messages with LOCATOR parameter [39] to inform Correspondent Nodes (CN) of additional locators at which the host can be reached. Additionally, the host can also declare a particular locator as a preferred locator. The HIP protocol introduces a new network entity called the Rendezvous Server (RS) to account for the cases when both hosts are moving or the destination's current locator is not known. Each HIP host publishes its host identifier in the RS, which maintains the mapping for identifiers and locators. HIP requires a basic extension [47] to support current NAT systems. Latest HIP implementations are using UDP based control protocol to support hosts behind NAT systems.

To enable network mobility, HIP introduces a Mobile Router (HIP-MR) [44] entity. HIP Simultaneous Multiple Access (HIP SIMA) [45] extension allows user define policies and enables flows to use different paths independently of each other based upon cost, bandwidth and other defined parameters. Multipath HIP (mHIP) [43] enables multipath support in HIP. Existing applications do not need any modifications to communicate with HIP enabled hosts. T. Koponen et al. [46] propose an application mobility mechanism in which service instances can obtain identities in addition to hosts. Services can delegate other services running on different physical hosts to impersonate them by using valid delegation certificates.

B. End-to-End Connection Control Protocol

The ECCP [49] protocol provides multipath and multihoming support by abstracting transport layer into data-delivery and connection control sub-layers. ECCP handles connection control sub-layer functionality that can also be reused with other transport protocols. ECCP takes the middle ground between MPTCP and HIP by being end-to-end and not introducing a new namespace (HIP). Similar to MPTCP, ECCP handles multipath by starting multiple subflows using IList parameter within a flow, which is identified by a flowID instead of five-tuples. It provides independent flow migration and policy management for better load balancing requirements.

Once the ECCP connection is established, the communicating nodes share a 64-bit nonce that is required for all future control messages. The 64-bit nonce prevents connection hijacking and disruption by spoofed control messages. To provide backward compatibility with NAT, ECCP packets are encapsulated as UDP payload (with a constant port number). ECCP-aware NAT boxes maintain their mapping using flowIDs instead of usual five-tuples. ECCP provides opportunistic mobility support using RSYN packets and *version numbers*. Version numbers are used to avoid acting on the past mobility events.

C. Site Multihoming by IPv6 Intermediation

The SHIM6 [50] protocol is a multihoming protocol supporting only IPv6 hosts. SHIM6 introduces a new shim sub-layer, SHIM6, between network and transport layer, which manages the locator/identifier split. Upper Layer Identifier (ULID) is used by the transport layer and locators (IP addresses) are used for routing. SHIM6 provides the functionality to map ULID with locators. ULID can be a normal IPv6 address or a Cryptographically Generated Address (CGA) [51] or a Hash Based Address (HBA) [52]. ULID is the initial default address at the time of path initialization process.

To enable multipath communication SHIM6 uses Context Forking (CF) [50] mechanism, which enables upper layer protocols to use different locator pair for the same destination. For handling mobility events, end-hosts use UPDATE REQUEST message to update CN's locator list with the newly added locator. The CN can reply with Update-Ack message on successful updates. It does not support NAT, as control packets containing locator values will be different from actual source address. SHIM6 has little support for traffic engineering as end-nodes can only set the primary locator for traffic.

The SHIM6 protocol is easier to deploy than HIP – Non-SHIM6 hosts can easily communicate with SHIM6 enabled hosts as they can simply drop the SHIM6 initialization request reverting back to normal IPv6 functionality. Socket APIs that provide simple on/off functionality, location management, feedback from upper layers have been proposed for general *network shim sub-layers* [53].

D. Identifier-Locator Network Protocol

The ILNP [54] protocol implements the locator/identifier split by emphasizing the use of Provider Allocated (PA) addresses over Provider Independent (PI) addresses. Locator is used to route traffic while the identifier is used as a node identifier. ILNP protocol can support both IPv4 and IPv6 network protocols. In IPv6 version (ILNPv6), the 128-bit IP address is divided into 64-bit locator and 64-bit identifier. In IPv4 version (ILNPv4), the current source and destination addresses represent locators and the 64-bit identifier is carried in the IP-Option header. The 64-bit identifier may be globally unique, but, it must be locally unique. Globally unique addresses are preferred, which removes the the need for Duplicate Address Detection (DAD).

Addresses can be generated using local MAC address or using cryptography, similar to host identities (HIT) in HIP. ILNP introduces a new network shim sub-layer for connection management. Transport layer protocols depend only upon identifiers, for example, TCP pseudo header includes only identifier to calculate TCP checksum. DNS record is updated to return identifiers and locators instead of IP addresses [56].

The ILNP protocol supports multihoming and mobility natively. Hosts can send ICMP locator update message [55] to all the active CNs and update DNS via secure dynamic DNS updates upon change in locator values. As the transport layer depends only upon identifier, ILNP can also be deployed on end-site routers without any changes. Non-ILNP hosts can communicate directly with ILNP enabled hosts as they can ignore the IP-Option header. ILNP is still a working draft with no currently available open implementation.

E. Mobile IPv6

The MIPv6 [57] protocol provides mobility support to IPv6 enabled hosts by maintaining the same IP address across networks. The MIPv6 protocol introduces essential enhancements while removing Mobile IPv4 [58] shortcomings. MIPv6 removes tunneling overhead by sending the new care-of address via binding update messages using Destination Option IPv6 extension header to all active CNs. Current MIPv6 protocol standard allows a MN to have multiple care-of addresses. But, only one primary care-of address is registered with HA and CNs. Also, MIPv6 requires HA to maintain its home IP address.

The Multiple Care of Address (MCoA) proposal [59] extends MIPv6 to allow the registration of multiple care-of addresses creating multiple binding cache entries. To distinguish between multiple bindings, a MN creates and manages a new Binding Identification (BID) number for each new binding and sends it inside a Binding Update message. The flow bindings extension [61] allows multiple flows (multipath) between two nodes. The flow binding is defined by a set of packet matching traffic selectors based upon source and destination IP addresses, transport port numbers, and other fields in IP and higher layer headers. The flow binding extension allows independent user defined policies for each binding flow.

The Network Mobility (NEMO) Basic Support protocol [60] extends MIPv6 to support mobility of an entire network of nodes moving together. NEMO introduces an additional Mobile Router (MR) entity along with the HA in the network. Each MN is connected to the MR. A MR can allow another MR to attach to its network, creating arbitrary levels of nested mobility. A NEMO enabled network is considered multihomed when a MR has multiple egress interfaces connecting to the Internet or when there are multiple MRs or multiple global prefixes in the network.

The MIPv6 protocol requires host networking stack changes. To reduce the complexity of host stack and expand the support for range of mobile devices, network based localized mobility management (NETLMM) [62] is proposed. Proxy Mobile IPv6 (PMIPv6) [63] is a network mobility management protocol for local domains in which hosts do not participate in mobility management. It is designed to assist both, IPv6 and IPv4, mobile nodes that do not support mobility natively. PMIPv6 introduces two network entities, Local Mobility Anchor (LMA) and Mobile Access Gateway (MAG). The LMA acts as the Home Agent for a MN and the MAG as an access router capable of managing the signalling for a MN attached to its link.

End-site Solutions

End-site multihoming solutions do not require end-host changes. They provide reliability, Internet routing scalability via traffic engineering, and bandwidth availability to the whole site. IPv4 can support multihoming using Border Gateway Protocol (BGP), which requires Provider Independent (PI) addresses. The problem with this approach is that global routing tables become enormously large hence difficult to manage. Network Address Translation (NAT) can also be used for multihoming but transport layer survivability is not guaranteed. For the Locator/Identifier split mechanism different approaches have been pursued, namely *map-and-encap* and *address rewriting*.

A. Network Prefix Translation

The Network Prefix Translation (NPTv6/NAT66) [64] uses *address rewriting* mechanism to support multihoming natively for IPv6 enabled hosts. It is easy to deploy and does not require any change to the hosts. It provides a stateless translation functionality by defining a two way checksum neutral algorithmic function. NPTv6 applies a complimentary change to other part of the IPv6 address that is not used for routing in the external network to keep the transport layer checksum same. As the translation is stateless, sessions using IPsec ESP encryption can cross it.

NPTv6 transfers the complexity of the network to application layer – if an application wants to know its outside addresses, it must use mechanism like DNS name when referring to themselves or use different mechanism to determine all the outside addresses. NPTv6 provides hair pinning behavior – if the translator receives a datagram on the internal interface

that has a destination address that matches the site's external prefix, it will translate the datagram and forward it internally.

B. Internet Routing Overlay Network

The Internet Routing Overlay Network (IRON) [65] architecture proposes a new virtual overlay network on which specific routers manage virtual prefixes. Provider independent prefixes are leased to end-nodes (clients). The IRON uses the existing IPv4 and IPv6 global Internet routing system as virtual Non-Broadcast Multiple Access (NBMA) links for tunneling inner network protocol packets. It introduces three new network elements, IRON client (*Client*), IRON server (*Server*), and IRON relay (*Relay*).

A Relay is an overlay network router that acts as a relay between the IRON and the native Internet. A *Server* router provides forwarding and mapping services for network prefixes owned by Clients. A *Client* is a router or a host that logically connects a MN to a *Server* via a bi-directional tunnel. After the *Client* selects a *Server*, it forwards initial outbound packets by tunneling them to the *Server*, which in turn, forwards them to the nearest *Relay* within the IRON overlay that serves the final destination.

The IRON supports multihoming natively – a *Client* can register multiple locators with its *Server*. The *Client* can assign metrics with its registrations to inform the *Server* about a preferred locator. Traffic engineering can be performed at *Server* and *Client* levels using user or network policies.

The IRON can support network mobility with *Client* router registering its location with *Server* whenever mobility event takes place. If a *Client* is moving away significantly from the current IRON server, a new registration can be performed with the nearest IRON server. To support terminal mobility, the *Client* router functionality should be implemented in the host. Also, IRON supports only those *Clients* behind NAT that use transport protocol with NAT traversal (e.g. UDP). IRON is an experimental proposal with no currently available open implementation.

C. Location Identification Separation Protocol

The Location Identification Separation Protocol (LISP) [66] is also an address family agnostic network based *map-and-encap* protocol. In LISP, a host is identified with Endpoint Identifier (EID), and source and destination hosts communicate via EIDs. On arrival of outbound packets, the source exit router, Ingress Tunnel Router (ITR), maps destination EID to a Routing Locator (RLOC) that corresponds to an entry point in the destination domain. This is the *map* phase of map-and-encap. In the *encap* phase, ITR encapsulates the packet and sets the destination address to the RLOC returned by the mapping infrastructure. On arrival of inbound packets, the destination router called Egress Tunnel Router (ETR), performs decapsulation and packets are delivered to destination EID. LISP separates its operations into data plane (map-and-encap) and control plane (EID-to-RLOC mapping system). This separation allows different mapping systems (LISP-ALT, LISP-TREE, LISP-MS) to be used along with data plane operation. ITR can define weights and priorities to each external connection, which enables traffic engineering capabilities and user defined policies for flow selection. Communication with non-LISP domain is done via Proxy Tunnel Routers (PTR).

The LISP-Alternative-Topology (LISP-ALT) [67] is one of the proposed mapping systems that describes EID-to-RLOC mapping without introducing any new protocols. LISP-ALT uses Border Gateway Protocol (BGP) and its multi-protocol extension along with the Generic Routing Encapsulation (GRE) protocol to construct an overlay of devices that advertise EID prefixes only. LISP-ALT can suffer from initial packet delay when contacting the authoritative ETR.

The LISP Mobile Node (LISP-MN) [68] enables mobility in MNs by implementing ITR and ETR functionalities directly in end-hosts. For communicating with non-LISP sites, all packets are encapsulated and routed to the Proxy Egress Tunnel Router (PETR). When LISP-MN moves to a new network, it receives a new RLOC address, which gets registered with the mapping system. Corresponding ITRs and Proxy Ingress Tunnel Routers (PITRs) must also update their cache. To support NAT, LISP-MN requires an additional mapping server, NAT traversal router (NTR). Implications of LISP-MN working inside a LISP enabled end-site are not addressed.

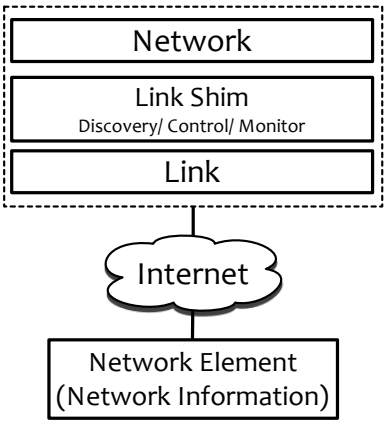

Fig. 5. Link Shim Sub-Layer

LINK LAYER

The Link layer solutions introduce a shim sub-layer support to provide handover optimizations, network discovery, control and monitoring of interfaces, and network access policies to upper layers. The shim layer can be deployed in both end-host and end-site solutions. For reducing network handover delays by providing network infomation apriori, and providing network access policies, the link shim sub-layer protocols introduce, in general, a new network element. Figure 5 illustrates a general link shim sub-layer based network architecture and the supported functionalities. We will discuss two main solutions that provide seamless heterogeneous network access to MNs – IEEE 802.21 Media Independent Handoff (MIH) framework and Access Network Discovery and Selection Function (ANDSF).

A. IEEE 802.21 Media Independent Handoff

For providing seamless connection handover in heterogeneous and homogeneous networks, including IEEE 802 and cellular networks, IEEE has standardized the 802.21 Media Independent Handover (MIH) [69] framework. The 802.21 standard defines a Media Independent Handoff (MIH) function – a abstraction layer above link layer, which provides a common interface to upper layers hiding technology specific primitives. To handle each technology, MIH maps the generic interface to a specific media dependent Service Access Point (SAP) whose aim is to collect information and to control link behavior during handovers. This abstraction is used by the IP layer (MIH User) or any other upper layer to better interact with the underlying technologies.

The 802.21 standard allows both, network and host controlled handovers. It defines three different types of MIH services based upon their association semantics: (1) Event Services (ES), (2) Command Services (CS), and (3) Information Services (IS). The ES is used to deliver events to upper layers or within link layer. CS enables MIH function to receive commands sent from higher to lower layers for network management. The IS provides a framework to acquire additional network information within a geographical area to facilitate handover decisions. An enhancement [70] to IS framework, using Location-to-Service Translation (LoST) protocol, proposes a new architecture for acquiring geographic network information. It defines three-layers – (1) LoST layer for specific geographic region, (2) ISP IS servers, and (3) Independent Evaluator IS servers. In a multi-party heterogeneous environment, 802.21 can be functional only if service providers have prior contracts with other service providers, for example, Verizon customers using AT&T network or a nearby Boingo Wi-Fi hotspot.

B. Access Network Discovery and Selection Function

Access Network Discovery and Selection Function (ANDSF) [71] is a cellular technology standard with similar functionalities as MIH framework by 3GPP. The ANDSF framework defines how a MN can connect to the Evolved Packet Core (EPC) [72] using a non-3GPP access (DSL, WLAN, WiMAX, CDMA2000) interface. It provides seamless vertical handover and allows operators to provide a list of preferred networks with access policies. The policies can be defined for a single IP flow or for all traffic for a given Packet Data Network (PDN).

The ANDSF information is represented by the ANDSF Management Object, a XML document compatible with Open Mobile Alliance-Device Management (OMA-DM) standard. The document specifies – MN location, Discovery information, Inter-System Mobility Policies (ISMP), and Inter-System Routing Policies (ISRP). A MN can send its location to the ANDSF server based upon geographical coordinates, cellular area or a WLAN location (SSID). Discovery information may be sent by the ANDSF server, which allows the MN to map its current location to a list of alternative access networks.

The ISMP consists of priortised rules that control network access. Similar to ISMP, ISRP is a list of priortised rules that control the network access based upon – (1) per PDN basis (all network flow) or (2) per IP flow basis (specific flow control).

Physical Layer

To efficiently use the available wireless spectrum based upon the location of a mobile node the Dynamic Spectrum Access (DSA) [73] solutions through opportunistic access to the licensed bands (e.g., TV Whitespaces) without interfering with the existing users. DSA enables on-demand high bandwidth for mobile users. Mobile nodes can opportunistically performs (1) spectrum sensing, and (2) spectrum management with corresponding Base Station (BTS) by using cognitive radio methods [73] or by quering a remote geospatial spectrum database [74].

Mobile Communication Architectures

A. Architecture for Ubiquitous Mobile Communications

The Architecture for ubiquitous Mobile Communications (AMC) [75] proposes a scalable next generation wireless system (NGWS) that integrates heterogeneous wireless systems and provides best network selection based upon user's network access needs. The network interoperating agent (NIA) eliminates the need for direct service level agreements (SLAs) among service providers. The NIA provides authentication, billing, and mobile management for customers. The interworking gateway (IG) is used to manage physical network resources. It also provides authentication, authorization, and accounting (AAA) along with mobility management for the NIA.

B. OpenRoads Architecture

The OpenRoads [77] architecture decouples service providers from physical infrastructure using OpenFlow [76] protocol resulting in a flattened physical infrastructure comprising of all heterogeneous mobile access technologies and a service provider that uses this flattened infrastructure via standardized APIs.

The OpenFlow protocol is used to control switches and routers centrally. It defines a flow-table in an OpenFlow switch and a action associated with it (datapath) and a remote controller (control) which communicates securely with switches using OpenFlow protocol. OpenFlow defines a software defined network (SDN) where the logic for network operations (policy management, routing etc.) are all done in software (controller). OpenFlow also provides a means to control the datapath elements (set power levels, allocation of channels, port blocking etc.) using SNMP or NetConf protocols.

The OpenRoads architecture provides a virtual layer (transparent proxy) that appears as a controller to the datapath and as a private network to the controllers called FlowVisor [78]. FlowVisor is used to create network slices and provide isolation between them. Slices can comprise of whole network,

Protocols	Pros	Cons
MPTCP	• Opportunistic mobility. • Multipath and Multihoming support. • Supports IPv4 and IPv6. • No modification to existing applications. • Backward compatible with TCP.	• Specific solution for transport layer protocol (TCP).
SCTP	• Allows multi-streaming of data in a single association. • Mobility, Multihoming and Multipath support.	• Both ends should support it. • Middlebox support is limited. • No locator/identifier split. • Requires application modifications. • NAT requires update.
HIP	• Mobility, Multihoming and Multipath support. • Baseline end-to-end security. • Host identification and authentication. • Supports IPv4 and IPv6. • No modification to existing applications.	• Both ends should support it. • Base exchange and payload overhead. • Identity namespace management.
ECCP	• Mobility, Multihoming and Multipath support. • Supports IPv4 and IPv6. • No modification to existing applications.	• Both ends should support it.
SHIM6	• Simpler than HIP.	• Supports only IPv6. • No mobility support. • No NAT support.
ILNP	• Supports end-host and end-site functionality. • Multihoming and Mobility support. • Supports IPv4 and IPv6. • NAT Support.	• New DNS records for mobility support. • No implementation.
IRON	• Incremental deployment with a business model.	• Uses an overlay network. • No NAT support for TCP.
Mobile IPv6	• Improved performance compared to Mobile IPv4. • Multiple flow support.	• Encapsulation overhead when using Home Agent without route optimization.
LISP	• Flexible mapping system. • Supports IPv4 and IPv6.	• Initial packet overhead. • Encapsulation overhead. • Replication of ITR and ETR to support mobility. • Requires additional mapping server for NAT support.
NPTv6	• Simple implementation	• Supports only IPv6. • No mobility support.

TABLE I
TRANSPORT AND NETWORK PROTOCOLS EVALUATION

a part of network or shared switches. For slicing datapath configuration OpenRoads uses SNMPVisor, that runs alongside FlowVisor.

A mobility manager application enables mobility in OpenRoads without any host changes. Mobility manager maintains the same IP address of the MN across multiple networks by controlling the DHCP servers. As OpenFlow is independent of the physical layer, vertical handoff between different radio networks is transparent. Multihoming support in OpenRoads is possible but it requires further research. Policy management can be implemented in the mobility manager as it has control of physical infrastructure. OpenRoads require all infrastructure switches (especially edge routers) to be OpenFlow enabled to support mobility.

EVALUATION

In the above discussed solutions, we observe that most protocols provide mobility, multihoming, multipath, some support

for flow management, and legacy host support via proxys either natively or by extensions. Table 1 lists the pros and cons of the above discussed protocols. In view of our design goals, we evaluate each protocol in terms of support for multihoming, mobility, multipath, and flow management.

The MPTCP protocol provides multipath, terminal mobility and mobile device energy savings. It provides multipath support using resource pooling principle. We believe resouce pooling should be decoupled from the core functionality as a separate function similar to MIH function. A separate function will enable any future transport protocol to leverage this functionality. MPTCP supports existing applications without any changes. SCTP provides multihoming and multi-streaming functionality natively. Mobility support can be added to SCTP using extensions. To support SCTP, existing applications require modifications and the limited firewall support makes its deployment difficult. HIP provides multihoming, terminal and network mobility, and multipath support. In addition to these functionalities, it also provides native baseline end-to-end security, which other discussed solutions do not provide. HIP incurs base exchange and payload overhead due to native ESP traffic. The ECCP protocol logically decouples the connection control (similar to resource pooling) from the data delivery mechanism of transport protocols. ECCP can be used with multiple transport protocols. SHIM6 adds native multihoming support in IPv6, it does not support IPv4, and there is no native mobility support. ILNP is inspired by HIP and SHIM6 protocols. It supports multihoming and mobility natively. It can be deployed at either end-host or end-site. With no current available implementation its performance cannot be evaluated. MIPv6 is an enhancement to MIPv4 protocol. MIPv6 removes the requirement Foreign Agent (FA). In MIPv6, a MN can directly communicate with CN without incurring tunneling overhead using binding messages via Destination Option IPv6 extension header. Multiple care-of-address extension adds multihoming support in MIPv6. To support legacy hosts, a network based mobility protocol called PMIPv6 can be used.

With end-site approaches, support for multihoming and traffic engineering is implemented at the end-site routers. The LISP protocol provides multihoming and traffic engineering support using flow priority and weight attributes. LISP adds encapsulation and initial packet overhead. For mobility support, it requires ITR and ETR functionalities to be implemented in end-hosts, which is contrary to the expectation that this functionality can be implemented at the end-site without any changes to MN. Also, the end-host networking stack becomes more complex and scalability can become a major issue.

The OpenRoads architecture uses a network based protocol called OpenFlow to partition physical resources. To support mobility, OpenRoads mobility manager maintains the same IP address for the MN irrespective of its location. For supporting millions of nodes, scalability can become a major issue with this approach. With NIA and IG components, AMC architecture provides the right abstractions for the physical infrastructure and service providers. NIA manages the service provider and each infrastructure provider is managed by IG. OpenRoads introduces some novel ideas like FlowVisor, which network operators can use to share physical network resources among multiple service providers. Integrating ideas of OpenRoads and AMC architecture to provide a hybrid architecture where FlowVisor manage the physical resources and NIA manage the service providers will be a more scalable solution. Unlike other projects, we have attempted to give a comprehensive view of the current state of the art technologies that can be used for the future Internet architecture.

Despite the use of IPv6 address space, NAT will remain an essential part the Internet architecture. It is essential for any new architecture to support NAT. We observe in all discussed protocols, NAT support requires further research, NAT changes or additional network elements. We believe an end-host solution that supports NAT and which can also be deployed at end-sites with minimal changes is an ideal solution for the future Internet architecture.

IV. Architecture

In the previous section, we saw several proposals for improving present Internet communications at application, transport, network, link and physical layers of the Internet protocol suite. Almost all the solutions introduce a new type of shim sub-layer functionality and new APIs to provide clear abstractions for upper layers. We believe, for any future networking stack, all new layer functionalities should provide APIs for event propagation machanism to upper layers because having knowledge of lower layer's environment enables certain essential house-keeping mechanisms, for example, handling network errors and enabling adaptive network flows. Similar to shim sub-layer functionality, new socket APIs should also provide essential feedback about network events to applications.

Managing heterogeneity, in general, is a complex problem with no single solution to it. In the last sections, we saw solutions for providing multihoming and mobility support. The AMC [75] architecture provides support for managing multiple physical infrastructure providers. To manage heterogeneity at both, device and network sides, we propose a unified network architecture that provides pervasive mobility across networks and devices. On the MN side, the architecture manages heterogeneous network interfaces and on the network side, it manages heterogeneous infrastructure providers seamlessly.

Mobile Node

Figure 6 illustrates an enhanced Internet protocol suite where network, link and physical layers are decoupled by the corresponding control functions (CFs). The CFs are controlled by the control middleware, which also provides enhanced BSD socket networking APIs. It also supports policy management functionality to control all networking layers, where user policies can have an influence on the network access. Similar to IEEE 802.21 MIH framework [69], upper layers can also influence lower layers using commands via CF and also, lower layers can influence upper layers using events.

The Physical Control Function (PCF) provides DSA functionality, which allows spectrum channel selection for efficient wireless communication based upon location and network conditions. The Link Control Function (LCF) decouples network and link layer functionality. It provides handover delay optimizations in both, homogeneous and heterogeneous networks, link status (up/down), link control and monitoring functionalities. The PCF and LCF are together, similar to PHY and MAC functionalities. These functions provide standard control APIs, similar to MIH Function, and inteface specific functionality is implemented using MIH Service Access Points (SAP). Any (present or future) link layer technologies can be added without changing upper layers. The control middleware provides control plane access.

Similar to LCF, we define Network Control Function (NCF) that provides, at least, multihoming, mobility, multipath, end-host identity and flow management support. Having terminal mobility at this sub-layer will have a system wide effect and all applications will work seamlessly. All transport layer protocols, current and future ones, can benefit from these functionalities without having to implement them individually. NCF also provides end-host identity that is essential for having locator/identifier split to free-up the overloaded IP addresses. Additionally, it can also provide end-host verification, for example, HIP [38] performs the initial base exchange between end-hosts using corresponding host identities.

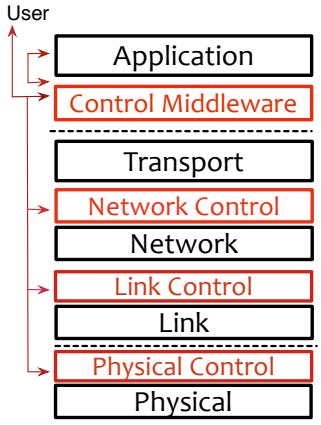

Fig. 6. Enhanced Internet Protocol Suite

The Control Middleware (CM) provides sockets API enhancements and manages control functions, which allows applications to select, add, and remove interfaces for any new or existing connections. The OS vendors can support session mobility and disruption tolerance in the control middleware or by introducing a networking library on top of the socket APIs. We can argue that disruption tolerance can reside in NCF sub-layer, but handling disruption tolerance in the control middleware provides greater flexibility for implementing application specific policies. Furthermore, future applications, as an option, can specify only the QoS requirements and the control middleware decides what kind of transport protocol best suits these requirements. Similarly, all applications can benefit from session mobility without requiring any custom protocols to achieve it.

The CM is comprised of a policy engine (PE) that makes dynamic decisions, based upon control event inputs from various attribute managers, as shown in Figure 7. The PE evaluates a state-vector of these current control events against pre-defined policies, resulting in a modification of system behavior. The state-vector defines a context at any given time, for example, location, time of day, and network type, cost, bandwidth, and latency. The CMs attribute managers include: The network manager (NM) that maintains and monitors all active network interfaces information. The NM also provides network information to the PE and executes the network handover decisions. The security manager (SM) maintains networks and devices access credentials, and end-to-end communications public/private key pairs. The location manager (LM) provides MN location information to the PE based upon GPS coordinates, or indoor positioning parameters, such as, Wi-Fi network identifier. The service-sharing manager (SSM) provides a centralized service registration function for local network discovery. The system manager maintains system parameters such as CPU, bandwidth and battery usage, to enable application-specific usage constraints, such as maximum bandwidth limit or battery utilization. And the queue manager maintains application specific queues to store data packets when there is no network connectivity, or a policy enforces no network usage, enabling application disruption tolerance support. The data store (DS) provides a structured key-value repository for each attribute manager respectively. Additionally, the socket proxy provides legacy application support by intercepting socket system calls.

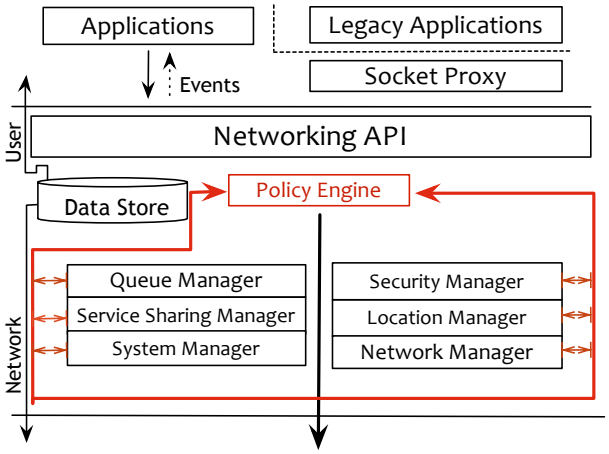

Fig. 7. Control Middleware

The enhanced Internet protocol suite provides better control of network access based upon networking conditions and user-context. We envision innovative future *mobile* applications that can make use of network, location, and device heterogeneity efficiently. This *context-aware* networking stack is the basis of our unified mobile architecture.

Network

Currently, there are two possible solutions for managing multiple service providers on a single mobile device. In the first case, all networks are managed by the end-host without any service provider interventions, in this case, customers receive two separate bills, for example, current dual sim cellular phones. In the second case, all networks are managed by a single service provider and end-hosts do not have any knowledge about it, for example, Boingo Wi-Fi service. In this case, customers receive a single unified bill. Also, in the first case, the end-host manages multiple IP addresses without any assistance from the network operators, and in the second case, the IP address may or may not be managed by the service provider depending upon the service type. Both these solutions have their own strengths and weaknesses. The first solution simplifies the service provider tasks and complexity is transferred to the end-host, and for the second solution, service provider manages all the complexity for an end-host. In both these solutions, scalability can be a major issue – device stack can become complex due to more hardware and software components. Also, it is difficult to manage same IP address across multiple infrastructure providers to support session continuity. Furthermore, these solutions do not fully exploit the network infrastructure diversity around them.

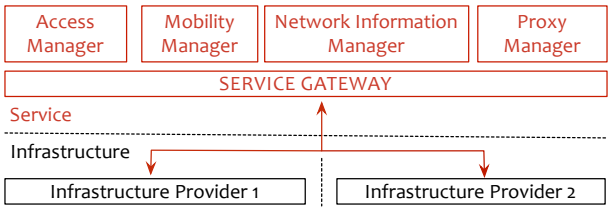

Fig. 8. Network Architecture Components

With a hybrid approach, where a single third-party entity manages end-users, and physical network operators are managed independently results in a more scalable service and seamless heterogeneous mobility. Customers receive a single bill and can connect to the best available network based upon the network conditions and cost. Service providers can manage their network capacity efficiently by load-balancing traffic among multiple physical infrastructure providers, resulting in service network decoupling into two logical planes – service and infrastructure. This decoupling, as shown in Figure 8, allows a service provider to offer services across multiple infrastructure providers, resulting in better management of heterogeneity and service scalability. Moreover, a single infrastructure provider can host multiple service providers, thus maximizing the infrastructure resource utilization, using for example, software defined network mechanisms.

Service Plane

The enhanced networking stack may be assisted with network-based external services to provide reliable mobility and additional network information support. The Mobility Manager (MM) service, acting as an independent centralized location registrar, assists the NCF to provide node reachability, while the node moves across domains, and acts as a rendezvous point, when both communicating nodes are moving concurrently. The Network Information Manager (NIM) service assists the LCF with additional network information such as nearby networks presence, bandwidth, cost, latency, and spectrum information for optimal network discovery and selection. The NIM service helps reduce handover delay and battery energy consumption. It may also use analytics to provide even more refined network selection capabilities. The Proxy Manager (PM) service enables incremental deployment of the system by converting the enhanced networking stack packets into the traditional stack packets. These services provide common functionalities across heterogeneous administrative domains, organically defining a separate logical service plane.

The identity, billing and accounting services can also be abstracted into a single service management logical entity, Access Manager (AM), which can be also deployed in the common service plane described above. This results in network providers that are responsible for only technology-specific infrastructure access, while all the service functionality can be deployed in the service plane. Additionally, a Service Gateway (SG) may provide an application layer firewall to further secure the interface between the service and the infrastructure plane, to prevent and mitigate general security attacks.

Infrastructure Plane

The infrastructure plane defines the physical network, which provides network access to authorized MNs. Infrastructure providers maintain their own independent DHCP servers. A *home network* is defined as the default network which users connect to when they switch on their mobile devices. A *foreign network* is defined as a different infrastructure provider other than the default home network. Currently, as many infrastructure providers are also service providers, all service plane components can reside in a single provider or multiple third-parties, which provide individual or combination of services. Service providers with no physical infrastructure can also provide services by using single or multiple physical infrastructure providers. As server virtualization created several cloud computing platforms, we envision network infrastructure virtualization creating several virtual service providers and providing innovative services to the customers.

Security

Each physical infrastructure provider functions independently, managing their own security infrastructure. For AAA functionality, the infrastructure plane communicates with the service plane's AM. A Federated Identity Management [79] scheme must be used to authenticate and authorize infrastructure access in foreign networks. When a MN moves to a foreign network, the MN's identity is authorized by the AM. We will discuss a complete security architecture for heterogeneous mobile environments in our future work.

V. Conclusion

Several recent proposals that enhance the Internet networking stack functionalities to provide multihoming, multipath, mobility, security, and disruption tolerance support were discussed and evaluated. By abstracting common design patterns, a unified context-aware architecture has been presented, as an evolution of the traditional networking stack. This enhanced stack is comprised of control functions for each corresponding layer and a control middleware that abstracts network complexity and provides a policy-based decision making system. Moreover, the architecture also abstracts the service providing network into two separate logical planes, infrastructure and service. This decoupling allows a service provider to offer services across multiple infrastructure providers, resulting in better management of heterogeneity and service scalability. The service plane provides user management, and enhances the control middleware with network-based reliable mobility, and additional network information support across heterogeneous networks.

References

[1] Cisco Visual Networking Index, white paper, Cisco Systems Inc., Jun. 2011.

[2] J. Rexford and C. Dovrolis, *Future Internet Architecture: Clean-Slate Versus Evolutionary Research*, Communications of the ACM, Vol. 53 No. 9, pp. 36-40.

[3] H. Schulzrinne and E. Wedlund, *Application-Layer Mobility Using SIP*, SIGMOBILE Mobile Computing and Communications Review, Vol. 4, No. 3, Jul. 2000, pp. 47-57.

[4] D. Thaler, Identifier-locator Split: Architectural Discussion, MobiArch Workshop 2008.

[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, *SIP: Session Initiation Protocol*, IETF RFC 3261, Jun. 2002.

[6] A. Dutta, F. Vakil, J. C. Chen, M. Tauil, S. Baba, N. Nakajima, and H. Schulzrinne, *Application Layer Mobility Management Scheme for Wireless Internet*, IEEE International Conference on Third Generation Wireless and Beyond (3G wireless), 2001.

[7] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer, *Session Initiation Protocol (SIP) Session Mobility*, IETF RFC 5631, Oct. 2009.

[8] E. Guttman, C. Perkins, J. Veizades, and M. Day, *Service Location Protocol*, Version 2, IETF RFC 2608, Jun. 1999.

[9] K. H. Hansen, H. C. Nguyen, and H. P. Schwefel, *Session mobility solution for client-based application migration scenarios*, International Conference on Wireless On-Demand Network Systems and Services, Jan. 2011, pp. 76-83.

[10] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones, *SOCKS Protocol Version 5*, IETF RFC 1928, Mar. 1996.

[11] J. Salz, A. C. Snoeren, and H. Balakrishnan, *TESLA: A Transparent, Extensible Session Layer Architecture for End-to-end Network Services*, In 4th Usenix Symposium on Internet Technologies and Systems, Mar. 2003.

[12] Panoply Project, http://www.lasr.cs.ucla.edu/panoply/panoply.html.

[13] Gaia-Active Spaces for Ubiquitous Computing, http://gaia.cs.uiuc.edu/index.html.

[14] Habib A, Christin N, and Chuang J, *Taking advantage of multihoming with session layer striping*, In Proc. INFOCOM, 2007, pp. 16.

[15] S. Pack, K. Park, T. Kwon, and Y. Choi, *SAMP: Scalable Application-Layer Mobility Protocol*, IEEE Communications Magazine, Jun. 2006.

[16] A. Qureshi and John Guttag, *Horde: Separating Network Striping Policy from Mechanism*, MobiSys 2005.

[17] K. Fall and S. Farrell, *DTN An Architectural Retrospective*, IEEE Journal on Selected Areas in Communication, Vol. 26, No. 5, Jun. 2008.

[18] J. Ott, *Delay Tolerance and the Future Internet*, Wireless Personal Multimedia Communications, 2008.

[19] Challenged Internet Access Network Technology Infrastructure (CHIANTI),http://www.chianti-ict.org/.

[20] K. Scott and S. Burleigh, *Bundle Protocol Specification*, IETF RFC 5050, November 2007.

[21] A. Seth, S. Bhattacharyya, and S. Keshav, *Application Support for Opportunistic Communication on Multiple Wireless Network*, Nov. 2005.

[22] A. Moghadam, S. Srinivasan, and H. Schulzrinne, *7DS A Modular Platform to Develop Mobile Disruption tolerant Applications*, NGMAST 2008.

[23] B. D. Higgins, A. Reda, T. Alperovich, J. Flinn, T. Giuli, B. Noble, and D. Watson, *Intentional Networking: Opportunistic Exploitation of Mobile Network Diversity*, MobiCom 2010.

[24] J. Widmer, R. Denda, and M. Mauve, *A Survey on TCP-Friendly Congestion Control*, IEEE Network, May 2001, pp. 28-37.

[25] A. Ford, C Raiciu, M. Handley, and O. Bonaventure *TCP Extensions for Multipath Operation with Multiple Addresses*, IETF RFC 6824, Jan. 2013.

[26] C. Raiciu, D Niculescu, M. Bagnulo, and M. Handley, *Opportunistic Mobility with Multipath TCP*, ACM MobiArch, 2011.

[27] G. Hampel, and T. Klein, *Enhancements to Improve the Applicability of Multipath TCP to Wireless Access Networks*, IETF Internet Draft, work in progress, June 2011.

[28] C. Pluntke, L. Eggert, and N. Kiukkonen, *Saving Mobile Device Energy with Multipath TCP*, ACM MobiArch, 2011.

[29] D. Wischik, M. Handley, and M. Bagnulo Braun, *The Resource Pooling Principle*, ACM SIGCOMM CCR 58(5), 2008, pp 4752.

[30] M. Scarf and A. Ford, *MPTCP Application Interface Considerations*, IETF RFC 6897, Mar. 2013.

[31] R. Stewart, *Stream Control Transmission Protocol*, IETF RFC 4960, Sept. 2007.

[32] J. Iyengar, P. Amer, and R. Stewart, *Concurrent multipath transfer using SCTP multihoming over independent end-to- end pathbibitem117 G.N. Stone, B. Lundy, and G. G. Xie, Network Policy Languages: A survey and a New Approach, IEEE Network, Vol. 15, Jan 2001. s*. IEEE/ACM Transactions on Networking 14(5), 2006, pp. 951964.

[33] M. Riegel, and M. Tuexenhttp://tools.ietf.org/html/rfc6953, *Mobile SCTP*, IETF Internet Draft, Nov 2007.

[34] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama, and M.Kozuka, *Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration*, IETF RFC 5061, Sept. 2007.

[35] C. Huang, M. Lin, and L. Chang, *The Design of Mobile Concurrent Multipath Transfer in Multihomed Wireless Mobile Networks*, The Computer Journal, Vol. 53 No. 10, 2010.

[36] M. Scharf and S. Kiesel, *Head-of-line Blocking in TCP and SCTP: Analysis and Measurements*, IEEE GLOBECOM, 2006.

[37] Q. Xie, R. Stewart, M. Holdrege, and M. Tuexen, *SCTP NAT Traversal Considerations*, IETF Internet Draft, Nov. 2007.

[38] P. Nikander, T. Henderson, C. Vogt, and J. Arkko, *End-host mobility and multihoming with the Host Identity Protocol (HIP)*, IETF RFC 5206, Apr. 2008.

[39] P. Nikander, A. Gurtov, and T. R. Henderson, *Host Identity Protocol (HIP): Connectivity, Mobility, Multi-homing, Security, and Privacy over IPv4 and IPv6 Networks*, IEEE Communications Surveys and Tutorials, Vol. 12, No. 2, May 2010.

[40] J. Ahrenholz, *Host Identity Protocol Distributed Hash Table Interface*, IETF RFC 6537, Feb. 2012.

[41] P. Nikander and J. Laganier, *Host Identity Protocol (HIP) Domain Name System (DNS) Extension*, IETF RFC 5205, Apr. 2008.

[42] P. Jokela, R. Moskowitz, and P. Nikander, *Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)*, IETF RFC 5202, April 2008.

[43] T. Polishchuk and A. Gurtov *mHIP: TCP-Friendly Secure Multipath Transport*, In Proc. of 5th International Conference on Access Networks (ACCESSNETS'10), Nov 2010.

[44] J. Melen, J. Ylitalo, P. Salmela, and T. Henderson, *Host Identity Protocol-based Mobile Router (HIPMR)*, IETF Internet Draft, May 2009.

[45] Pierrel S, Jokela P, Melen J, and Slavov K, *A Policy System for Simultaneous Multiaccess with Host Identity Protocol*, In Proceedings of IEEE Workshop on Autonomic Communications and Network Management (ACNM), 2007.

[46] T. Koponen, A. Gurtov, and P. Nikander, *Application Mobility with HIP*, In Proc. Of NDSS Wireless and Security Workshop, 2005.

[47] M. Komu, T. Henderson, H. Tschofenig, J. Melen, and A. Keranen, *Basic Host Identity Protcol (HIP) Extensions for Traversal of Network Address Translators*, IETF RFC 5770, Apr. 2010.

[48] M. Komu and T. Henderson, *Basic Socket Interface Extensions for the HIP*, IETF RFC 6317, Jul. 2011.

[49] M. Arye, E. Nordstrom, R. Kiefer, J. Rexford, and M. J. Freedman, A Formally-Verified Migration Protocol For Mobile, Multi-Homed Hosts, IEEE International Conference on Network Protocols (ICNP), 2012.

[50] E. Nordmark, and M. Bagnulo, *Shim6: Level 3 Multihoming Shim Protocol for IPv6*, IETF RFC 5533, June 2009.

[51] T. Aura, *Cryptographically Generated Addresses (CGA)*, IETF RFC 3972, March 2005.

[52] M. Bagnulo, *Hash-Based Addresses (HBA)*, IETF RFC 5535, June 2009.

[53] M. Komu, M. Bagnulo, K. Slavov, and S. Sugimoto, *Sockets Application Program Interface (API) for Multihoming Shim*, IETF RFC 6316, July 2011.

[54] R. Atkinson and SN Bhatti, *ILNP Architectural Description*, IETF RFC 6740, Nov. 2012.

[55] R. Atkinson and SN Bhatti, *ICMP Locator Update message for the ILNPv4* , IETF RFC 6745, Nov. 2012.

[56] R. Atkinson, SN Bhatti, and S. Rose, *DNS Resource Records for ILNP*, IETF RFC 6742, Nov. 2012.

[57] C. Perkins, D. Johnson, and J. Arkko, *Mobility Support in IPv6*, IETF RFC 6275, July 2011.

[58] C. Perkins, *IP Mobility Support for IPv4*, IETF RFC 3220, Jan. 2002.

[59] R. Walikawa, V. Devarapalli, G. Tsirtsis, T Ernst, and K. Nagami, *Multiple Care-of Addresses Registration*, IETF RFC 5648, Oct. 2009.

[60] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, *Network Mobility (NEMO) Basic Support Protocol*, IETF RFC 3963, Jan. 2005.

[61] G. Tsirtsis, H. Soliman, N. Montavont, G. Giaretta, and K. Kuladinithi, *Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support*, IETF RFC 6089, Jan. 2011.

[62] J. Kempf, *Problem Statement for Network-Based Localized Mobility Management (NETLMM)*, IETF RFC 4830, Apr. 2007.

[63] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, *Proxy Mobile IPv6*, IETF RFC 5213, Aug. 2008.

[64] M. Wasserman and F. Baker, *IPv6-to-IPv6 Network Prefix Translation*, IETF RFC 6296, Jun. 2011.

[65] F. Templin, *The Internet Routing Overlay Network (IRON)*, IRTF RFC 6179, Mar. 2011.

[66] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, *Locator/ID Separation Protocol (LISP)*, IETF RFC 6830, Jan. 2013.

[67] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis, *LISP Alternative Logical Topology (LISP+ALT)*, IETF RFC 6836, Jan. 2013.

[68] D. Farinacci, D. Lewis, D. Meyer, and C. White, *LISP Mobile Node*, IETF Internet Draft, work in progress, Oct. 2012.

[69] IEEE Standard for Local and metropolitan area networksPart 21: Media Independent Handover Services, Jan. 2009.

[70] K. Andersson, A. G. Forte, and H. Schulzrinne, *Enhanced Mobility Support for Roaming Users: Extending the IEEE 802.21 Information Service*, WWIC 2010.

[71] Access Network Discovery and Selection Function (ANDSF) Management Object (MO), http://www.3gpp.org/ftp/Specs/html-info/24312.htm .

[72] 3GPP Evolved Packet Core (EPC), http://www.3gpp.org/The-Evolved-Packet-Core

[73] I. F. Akyildiz, W. Lee, M. C. Vuran, and S. Mohanty, NeXt Generation/ Dynamic Spectrum Access/ Cognitive Radio Wireless Networks: A Survey, Computer Networks Journal, Vol. 50, 2006, pp. 2127-2159.

[74] A. Mancuso, S. Probasco, and B. Patil, Protocol to Access White-Space (PAWS) Databases: Use Case and Requirements, IETF RFC 6953, May 2013.

[75] I. F. Akyildiz, S. Mohanhttp://tools.ietf.org/html/rfc6953ty, and J. Xie, *A Ubiquitous Mobile Communication Architecture for Next-Generation Heterogeneous Wireless Systems*, IEEE Radio Communications, Jun. 2005.

[76] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, *OpenFlow: Enabling Innovation in Campus Networks*, ACM SIGCOMM Computer Communication Review, Vol. 38, No. 2, Apr. 2008.

[77] K. Yap, M. Kobayashi, D. Underhill, S. Seetharaman, P. Kazemian, and N. McKeown, *The Stanford OpenRoads Deployment*, WiNTECH 2009.

[78] K. Yap, R. Sherwood, M. Kobayashi, N. Handigol, T. Huang, M. Chan, N. McKeown, and G. Parulkar, *Blueprint for intoducing Innovation into the Wireless Networks we use every day*, OpenFlow Technical Report, Oct. 2009.

[79] H. Gomi, M. Hatakeyama, S. Hosono, and S. Fujita, A Delegation Framework for Federated Identity Management, Proceedings of the 2005 workshop on Digital Identity Management.