A Survey of Mobile VPN Technologies

Abdullah Alshalan, *Student Member, IEEE*, Sandeep Pisharody, *Student Member, IEEE*, and Dijiang Huang, *Senior Member, IEEE*

***Abstract*—Virtual private network (VPN) is the traditional approach for an end-to-end secure connection between two endpoints. Most existing VPN solutions are intended for wired networks with high-speed, highly reliable connections. In a mobile environment, these network connections are less reliable. This affects traditional VPN performance resulting in frequent application failure, data loss, and reduced productivity. Mobile VPN bridges the gap between what users and applications expect from a wired network and the realities of mobile computing. In this survey, we provide a taxonomy of VPN designs, present a study of existing mobile VPN solutions, and highlight the advantages and disadvantages and applications of these methods.**

***Index Terms*—Virtual private network, VPN, mobile VPN, MVPN.**

I. Introduction

GLOBAL computing industry is quickly evolving toward having powerful cloud computing resources aimed at providing services over the Internet, with mobile devices behaving as the user interface into this cloud. In such an environment, having a way to securely connect these mobile terminals to a cloud computing resource like MobiCloud [1] is of great importance, not just for information assurance and protection of intellectual property, but often for regulatory and compliance reasons.

Classical Virtual Private Network (VPN) connections establish secure connections between a remote user and a home network by encrypting packets sent though the Internet rather than building a true private network [2]. These VPN connections however, are best suited for stationary devices which, unlike mobile devices tend to have a stable network connection [3]. Most mobile devices are susceptible to intermittent connection loss while switching from one network to another or experiencing a gap in coverage [4]. For example, a cellular phone might switch between WiFi and 4G, or between one WiFi and another. Such connection losses or connection changes can cause the VPN connection to break causing the mobile applications utilizing the VPN to either timeout or crash.

Given the ever increasing popularity of remote workers and Bring-Your-Own-Devices (BYOD) in work places along with the ubiquitous presence of wireless networks that these devices have access to, it is prudent to have a mobile VPN solution that can provide a VPN experience that does not require the user to reset and reconfigure the VPN session upon switching between networks. According to a survey done by Dimension Data [5], 79% of over 1600 surveyed IT and security professionals ranked mobility as a top priority. In the same survey, 71% of the respondents expressed that data security is the major concern of mobility. While there have been several mobile VPN protocols studied and several commercial mobile VPN solutions in the market, to the best of our knowledge, there has been no survey done that compares these protocols and solutions against each other. Saha *et al.* [6] provides a survey about several mobility protocols that support micromobility and macrombility to the IP layer. Their survey, however, does not pertain to mobile VPNs and only covers mobility support in the IP layer. Subsequent work in application layer mobility and mobility based on Host Identity Protocol are not included in that survey. Similar argument can be made against the survey done by Akyildiz *et. al.* [7] which survey layer 3 and layer 2 mobility management protocols in IP wireless networks. Zhu *et al.* [8] provides a survey of mobility support in the Internet, however this survey is more generalized than our mobile-VPN-specific survey.

The goal of this survey is to explicitly state requirements from a mobile VPN, study the various technologies that were tailored to a mobile environment, compare their characteristics, strengths, weaknesses and applications; with an emphasis on security and robustness in the context of a mobile device with frequent network disruption/handover. In Section II we go over some of the background information including network nomenclature, underlying network protocols and classification methodologies commonly used for VPNs. We present in Section III criteria for classification of mobile VPN. Section IV lists the requirements that we believe are essential for a mobile VPN. In Section V we discuss protocols that we believe are the building blocks for an effective mobile VPN solution. In Section VI we compare and contrast these technologies. We discuss some commercial mobile VPN products available in the market with two case studies in Section VII. We conclude this survey by discussing open issues in Section VIII before we present a summary in Section IX. For the reader's convenience to navigate this manuscript, we provide Table I serving as a road-map for this survey.

Manuscript received October 13, 2014; revised August 11, 2015; accepted September 29, 2015. Date of publication November 2, 2015; date of current version May 20, 2016. This work was supported in part by the NSF Secure and Resilient Networking (SRN) Project (1528099), in part by the NATO Science for Peace and Security Multi-Year Project (MD.SFPP 984425), and in part by the ONR YIP Project (N00014-10-1-0714). The work of A. Alshalan was funded by a scholarship from King Saud University. The work of S. Pisharody was supported by a scholarship from the NSF CyberCorps Program (NSF-SFS-1129561).

The authors are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: abdullah.alshalan@asu.edu; sandeep.pisharody@asu.edu; dijiang.huang@asu.edu).

Digital Object Identifier 10.1109/COMST.2015.2496624

TABLE I
ROAD-MAP OF PAPER LAYOUT

Section	Description
II) Background	This section provides technical background of VPNs, network components, protocols used as building blocks of (mobile) VPNs namely: PPP, L2TP, GRE, IPIP, IPsec, TLS, MIP, SIP, RTP, SRTP. We conclude this section with a summary of VPN classification. *Readers familiar with these protocols may skip this section.*
III) Mobile VPN Classification Criteria	In this section we provide criteria to classify mobile VPNs namely: *tunnel establishment, layer of mobility and security protocol. We conclude with a taxonomy of mobile VPN classification.*
IV) Mobile VPN Design Requirements	In this section we provide the requirements of mobile VPNs which are: *1) seamless network roaming, 2) security, 3) application persistency, 4) performance.*
V) Mobile VPN Technologies and Solutions	In this section we survey mobile VPN technologies which we divide into three categories: A)Network mobility-based: MIPv4 VPNs, MIPv4 w/2HA VPNs, MIPv6 VPNs, BGP/MPLS VPNs, MOBIKE VPNs, NEMO and Cellular VPNs (CDMA 2000, UMTS). B) Application mobility-based: SIP-based VPNs, WTLS-based VPNs, MUSeS, FastVPN. C)HIP-based.
VII) Commercial mobile VPN solutions	We provide the reader with a brief summary of currently available mobile VPN products. We also present two case studies of commercial mobile VPNs: Radio IP's Mult-IP and Columbitech Mobile VPN.
VIII) Future Work	Identify six open issues in mobile VPNs.
IX) Summary	Conclusion and summary of this paper.

II. BACKGROUND

In this section we provide background information in an attempt to make this survey more comprehensive and self-contained. In the remainder of this section we briefly define VPNs. After that, we provide definitions of network components that are used in mobile VPN solutions. We then provide a technical background section discussing, briefly, the protocols that are used in mobile VPN solutions and technologies. We conclude the section with a discussion about VPN classification for comprehensiveness.

A. Virtual Private Network (VPN)

VPNs are networks built as an overlay on the public infrastructure of one or more providers, so as to permit access between a defined set of devices [9]. A VPN can be simply thought to be an authenticated and encrypted tunnel to serve as a virtual leased line over a shared public infrastructure [9], [10]. According to the strictest definitions, a VPN does not have to ensure encryption of data. Per the Virtual Private Network Consortium (VPNC), a secure VPN is a user encrypted and authenticated connection *a*) between two segments of the same private network, *b*) from a computer to a private network, or *c*) between two computers [11]. Since VPN is between authorized users/devices, strong access control is essential for a secure VPN [12]. As is common in the industry, this survey will use VPN and secure VPN interchangeably.

B. Network Components

Several network components are common to most VPN solutions discussed in this survey. They are briefly discussed here:

- A mobile node (MN) is any network device that is not tethered to one static network. In most cases, the MN can physically move around. However, a node that has multiple network interfaces and has the capability to switch between them is for all intents considered to be a MN. Common examples of a MN are a cellular phone or a tablet. Since MN is typically on the move and any time it strays too far from the location of its access point, it switches to another network for its connectivity, thereby obtaining a new IP address.
- The Network which the MN originally joins the network is called the Home Network.
- The Home Agent (HA) is a router on a MN's home network. It maintains information about the MN's location at all times, and is responsible for delivering packets to the MN when it is away from its home network.
- The Foreign Agent (FA) is a router in the MN's visited network that provides routing services to the MN in the visited network.
- Correspondent Node(s) (CN) is a node with or without any mobile functionality, which communicates with the MN.

C. Protocols

This section will briefly present the different protocols that constitute the basis of the mobile VPN solutions discussed in Section V.

1) Point-to-Point Protocol (PPP): PPP is used to provide a virtual direct link over a multitude of physical mediums between two endpoints allowing encapsulation of network-layer datagrams into frames regardless of the nature of the physical medium. Authentication in PPP is achieved through Password Authentication Protocol (PAP) or Challenge-Handshake Authentication Protocol (CHAP) [13], [14]. Encryption can be added on by configuring PPP Encryption Control Protocol (ECP) [15] or by securing the data using a higher-layer protocol like IPsec.

2) Layer 2 Tunneling Protocols (L2TP): There are two main variations of the L2TP protocol: L2TPv2 and L2TPv3. L2TPv2 is a data link tunneling protocol used to tunnel multiple PPP sessions. Multiple PPP sessions are differentiated by using a session ID. L2TPv3 can be used to tunnel not only PPP frames but also IP packets. This eliminates the overhead of encapsulating IP packets within PPP frames [16]. Both L2TPv2 and L2TPv3 provide two channels within the tunnel: a reliable control channel and an unreliable data channel [16], [17]. L2TP does not provide any security features and relies on protocols like IPsec for message authentication and encryption [18].

3) Generic Routing Encapsulation (GRE): GRE is a protocol used to tunnel an arbitrary network layer protocol over any network protocol [19]. When encapsulating protocol X over protocol Y, GRE takes a packet in format X and considers it

to be the payload packet. It first encapsulates this packet in a GRE packet and then encapsulates this GRE packet in protocol Y, which is considered the delivery protocol. RFC2890 extends the GRE header to allow multiplexing different packet flows within one GRE tunnel with in-order delivery of packets [20]. This allows GRE to encapsulate PPP frames [18].

4) IP Encapsulation Within IP (IPIP): IP-in-IP or simply IPIP is defined in RFC2003 to allow for encapsulating an IP packet within another IP packet [21]. A tunnel can be established between two endpoints, an encapsulator and a decapsulator, in order to tunnel packets going from a source in the network of the encapsulator to a destination in the decapsulator's network. The inner IP header, added by the source, will contain the original source and destination IP addresses. The outer IP header, added by the encapsulator, will contain its IP address as the source IP and the decapsulator's IP address as the destination IP. It is then the responsibility of the decapsulator to process the IPIP packet and forward the original (inner) packet to the destination. Minimal IPIP was introduced in RFC2004 in response to the problem of the overhead of adding another IP header especially to packets with small payload [22].

5) Internet Protocol Security (IPsec): IPsec consists of guidelines for a series of protocols that secure communications at the network layer of the OSI stack [23]. Originally developed as a security extension for IPv6, IPsec was later adapted into working with IPv4; making it well-suited for use in both platforms [9].

The following protocols are commonly used as part of the IPsec suite:

- Encapsulation Security Payload (ESP) provides data source authentication, data integrity, data confidentiality and anti-replay protection of IP packets by encapsulating the data to be protected between the ESP header and trailer [24], [25].
- Authentication Header (AH) supports data source authentication and data integrity, but does not offer any form of confidentiality. This makes it a lot simpler than ESP, but also less commercially attractive. AH authenticates the entire datagram, unlike ESP, which does not authenticate the leading IP header or any other information that comes before the ESP header [9].
- Internet Key Exchange (IKE) protocol for negotiating IPsec connection settings and authenticating the end points. It defines the security parameters, negotiates keys, and manages the IPsec communication channels [10], [26].

IPsec can be implemented in two different modes: Transport mode, and Tunnel mode. In the former, the IP payload is encrypted, while the header is left intact. In Tunnel mode, the entire IP packet including the headers are encrypted, and encapsulated within new IP headers.

6) Transport Layer Security (TLS): TLS is a protocol designed to provide security to applications communicating in a client-server model. It is an upgrade to the commonly used SSL protocol. Unlike IPsec, TLS resides between the Application layer and the transport layer in the TCP/IP stack. Therefore, it provides confidentiality and integrity only to the application data whereas IPsec, depending on the selected security protocol, extends its security features to both the transport layer header and the IP header [27].

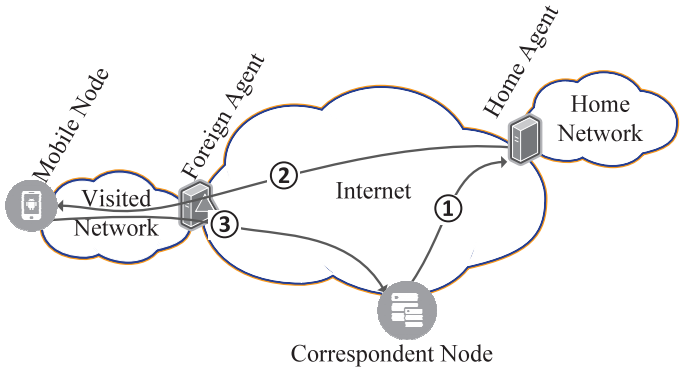

Fig. 1. Mobile IP.

Wireless TLS (WTLS) is an optimized version of the regular TLS protocol for a low-bandwidth wireless network with substantial latency [28]. WTLS provides compressed data structures to reduce packet sizes; and uses a new certificate format that compresses the TLS certificates. WTLS enhances TLS greatly by giving the client and server the ability to resume a previous session instead of negotiating new security parameters, thereby enhancing mobility.

7) Mobile IP (MIP): MIP is a popular protocol for extending mobility into IP. It enables session continuity when an end point travels among heterogeneous networks and ensures this mobility is transparent to applications [29]. MIP is thus ideal for providing mobile VPN connectivity independent of the underlying access technology.

The MN has two IP addresses: care-of address (CoA) and home address. The home address is constant and used to communicate with the CN. The CoA is a temporary address assigned by the visited network, and is used to build a tunnel from the HA to MN. The original packet is the encapsulated in the IP packet with CoA [29]. The general structure is shown in Figure 1.

Packets sent to the home address of the MN are intercepted by the HA (Step 1). The HA tunnels the packets to the MN (Step 2) since it is aware of the MN's location. After decapsulating the packet, the MN can send all further communication directly to the CN (Step 3). As long as the MN notifies the HA of its new CoA when there is a change in location, this process provides strong authentication technique for the MN [29]. However, security in MIP has two major limitations: *a*) Outgoing packets from the HA to the MN may be dropped by filters in visited network because of the use of the home address as the source address; and *b*) problems arise when the FA is not capable of reading a MN's request to its HA. This prevents the set up of the tunnel resulting in connectivity loss [9].

8) Session Initiation Protocol (SIP): SIP is an application-level signaling protocol used for controlling sessions such as voice and video calls. It contains design elements that are based on a request/response transaction model, wherein each transaction consists of a client request and server response of a particular method or function. SIP uses most of the header fields, encoding rules and status codes of HTTP, providing a

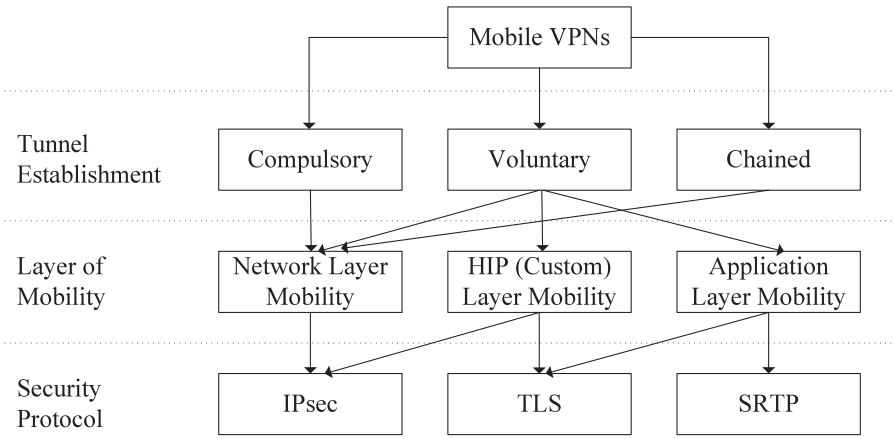

Fig. 2. Mobile VPN classification taxonomy.

readable text-based format. SIP works as the signaling portion of several other protocols [30].

Each resource of a SIP network is identified by a uniform resource identifier (URI), based on the general standard syntax also used in Web services and e-mail. The URI scheme used for SIP is `sip : username : password@host : port`. If secure transmission is required, the scheme `sips :` is used as the prefix instead of `sip :` to ensure that each hop over which the request is forwarded must be secured using Transport Layer Security (TLS) [30].

9) Real-Time Transport Protocol (RTP): RTP is a application layer protocol designed to provide end-to-end network transport functions suitable for applications transmitting real-time data, such as audio and video over multicast or unicast network services regardless of what the underlying network and transport protocols are [31]. Secure RTP (SRTP) intends to provide encryption, message authentication and integrity, and replay protection to the RTP data [32].

D. VPN Classification

Over the years, there have been several attempts to classify VPNs in order to gain a better understanding of their capabilities and application. We discuss a few of them briefly here, before picking a classification criteria that suits mobile VPNs.

- Mobility - Using mobility as a classification criteria, VPNs can be divided into Stationary VPNs and Mobile VPNs. As the name suggests, stationary VPNs are designed to work for nodes that do not move during a VPN session, such as a PC or a router; with mobile VPNs on the other hand, being able to support mobility for a MN whose IP may change periodically due to its geographic motion or due to it switching between available networks.
- Deployment - From a network architecture perspective, VPNs can be categorized into the following groups: a) Site-to-site VPN; b) Remote access VPN; and c) Peer-to-peer VPN. Site-to-site VPNs are used to establish a tunnel between two VPN gateways in order to virtually connect two separate networks such that resources in both networks can be available at the other end. Remote access VPNs are set up to allow remote users to connect to a private network through a VPN gateway. Peer-to-Peer VPNs are used to enable nodes in a P2P network to communicate amongst each other securely. P2P VPNs can be set up either in a centralized or a decentralized architecture [33].
- OSI Layer - Depending on what layer of the OSI stack the VPN service is provided on, VPNs can be divided into: a) Layer 1 VPN; b) Layer 2 VPN; and c) Layer 3 VPN. Layer 1 and Layer 2 VPNs provide the ability to create multiple virtual networks over the same physical network or over Asynchronous Transfer Mode (ATM) or Frame Relay (FR) circuits [34], [35]. Traditional VPNs such as ATM and FR networks, enable security at the Data Link layer (Layer 2) of the TCP/IP stack [9]. These solutions were not designed to support mobility of the end points [36]. Layer 3 VPN protocols such as GRE, IPsec, IPIP and BGP/MPLS offer the VPN end points the ability to route IP traffic between them [36]. Since mobile communications introduce problems because traffic is generally to and from a MN, in their basic forms, L3VPNs are not very forgiving of mobility [12].

III. Mobile VPN Classification Criteria

Based on their inherent characteristics and use cases, there are several criteria to classify mobile VPN technologies. In this Section, we introduce and describe a few such classification criteria in detail. We also provide a mobile VPN taxonomy based on these criteria illustrated in Figure 2, and we summarize the classification criteria in Table II.

A. Classification Criteria - Tunnel Establishment

VPNs can be categorized into the following three groups based upon tunnel establishment criteria: a) Voluntary VPN; b) Compulsory VPN; and c) Chained VPN tunnel.

1) Voluntary VPN: In a voluntary VPN, an end-to-end tunnel between a remote node and a VPN gateway is setup voluntarily or as dictated by need of the remote user. In this model, no intermediate nodes or entities are involved. The remote node has to have certain capabilities like IPsec, TLS

TABLE II
SUMMARY OF MOBILE VPN CLASSIFICATION CRITERIA

Classification Criteria	Class	Advantages	Disadvantages
Tunnel Establishment	Voluntary	- Simple Client/Server model. - No need for intermediary devices; thus, no compromise to the end-to-end encryption, and no SLA's required.	- Need for NATing or IPv6. - Packets cannot be inspected for QoS. - Requires encapsulation over lossy wireless links.
	Compulsory	- No encapsulation needed between MN and service provider. - No VPN support needed in MN. - Ability to provide QoS.	- Data is partially transmitted over insecure channel. - Intermediary devices have to be trusted. - Need for SLA which may not be suitable for low-budget organizations.
	Chained	- Eliminates the need for insecure channel between the service provider and the MN. - Allows for QoS and traffic shaping.	- Intermediary devices have to be trusted. - Need for SLA.
Layer of Mobility	Network Layer Mobility	- Solves the problem of IP address change in network layer transparently from the above layers of TCP/IP stack.	- Transport layer protocols may time out if the recovery of the IP layer is not done in a timely manner especially during long gaps in network coverage.
	Application Layer Mobility	- Supports mobility by creating session binding above the IP layer so it is not affected by IP address changes.	- More overhead on the mobile VPN to keep track of the session information.
Security Protocol	IPsec	- better performance than TLS.	- Requires establishing the security association from scratch unless MOBIKE is used.
	TLS and its variants	- NAT-friendly since they do not include IP addresses is part of the security association.	- Performance is not as good as IPsec. - Does not authenticate the sender's IP address.

etc. For example, when a MN sets up a voluntary VPN connection, the service provider (wireless carrier) is unaware of this tunnel. The remote user (whether stationary or mobile) can establish said VPN connection to any private network after gaining Internet access from the service provider. This model has the following advantages and challenges [18]:

Advantages:

- It is a simple client/server model of VPN. A tunnel can be established with relative ease as long as the client has access to a public IP network such as the Internet and a VPN client software, and the server has a VPN server software.
- The service provider and any intermediate nodes do not have to be party to setup the VPN. Once the tunnel is setup, the intermediate nodes do not have visibility into the encrypted traffic. As a result, they do not have to be trusted.
- Since the intermediate nodes have no visibility into end-to-end traffic, there is no need for Service Level Agreements (SLA) or any other legal documents ensuring the confidentiality of the data.

Challenges:

- The remote end requires a publicly routable IP address which can be a challenge given the limitation of available public IPv4 addressing. However, to overcome this challenge, voluntary VPNs can utilize NAT or IPv6. It is important to note that the NAT solution is inherently incompatible with certain designs of IPsec, and careful attention needs to be given to the network design before attempting to use such a solution.
- Any kind of traffic prioritizing and shaping using Quality-of-Service (QoS) by the service provider would not work since QoS would require packet inspection.
- Since additional encapsulation is required to setup a voluntary VPN, setting up and maintaining a VPN over a lossy wireless channel may lead to shoddy customer experience.

2) Compulsory VPN: A VPN established between the service provider and a private network, which is used only when the remote user wants to access resources in the private network is termed to be a compulsory VPN, simply because the user is forced to use the tunnel between the carrier and the private network's gateway. In this model, the MN does not need to support any tunneling or security protocols such as IPsec or TLS since it is completely unaware of the tunneling. The advantages and challenges of this model are as follows [18]:

Advantages:

- No encapsulation is needed between the MN and the service provider. The overhead of encapsulation and encryption is eliminated especially over lossy radio link.
- No VPN support needed in the MN. This would save the MN's resources such as battery and CPU.
- Ability to provide QoS to differentiate level of service for voice, video and data services.

Challenges:

- The lack of an end-to-end VPN connection means that the data is partially transmitted through an insecure data channel, and is thus open to snooping.
- The service provider has to be trusted since the radio link connection is terminated at the service provider. Even if some encryption is used over the radio link, traffic will be decrypted before it is encapsulated and forwarded to the private network through the compulsory VPN tunnel.
- There is a need for an SLA and the service provider has to be involved, thereby leading to additional costs. This may make it unsuitable for small businesses and academic institutes.

3) Chained VPN Tunnel: The chained VPN tunnel model uses concatenated tunnels provided by the service provider but extends it past the base station to the remote user. Such a

model allows for QoS and traffic shaping. However, the service provider still has to be trusted. This model however eliminates any insecure data channels that exist in the compulsory VPN model [18].

B. Classification Criteria - Layer of Mobility

Mobile VPNs can be classified based on which layer of the TCP/IP stack is mobility taken care of. Based on this criteria mobile VPNs can be categorized into: a) Mobile VPNs based on network layer mobility. These mobile VPNs address the mobility problem in the network layer. They solve the problem of IP address change by network level means such as redirecting traffic to a HA in Mobile IP and support for mutli-homing MOBIKE. b) Mobile VPNs based on application layer mobility. These mobile VPNs support mobility by creating session binding above the IP layer so it is not affected by IP address changes. SIP-based mobile VPNs and TLS-based VPNs are examples of this category.

C. Classification Criteria - Security Protocol

Security is an essential part of VPNs. Encryption, authentication and message integrity provided by VPNs can be broadly categorized into: a) protocols based on network layer security such as IPsec; and b) protocols based on application layer security such as SRTP and SSL and its variants TLS, DTLS and WTLS.

IPsec applies security to IP datagrams on network layer, so there is IP address binding between the two endpoints of an IPsec tunnel. However, application layer security protocol are more suitable for mobility and more NAT-friendly since they do not include IP addresses as part of the security association [37].

IV. Mobile VPN Design Requirements

A mobile VPN is, by professional consensus, thought to have the following requirements:

1) *Seamless Network Roaming (SNR)*: When the MN performs a vertical handoff (MN uses a different network interface such as switching from cellular interface to WiFi) or horizontal handoff (MN switches from one network to another while using the same medium; e.g. switching from one WiFi network to another) and receives a new IP, the VPN functionality should remain intact without user involvement. In both scenarios, the physical IP address for the VPN tunnel (outside address) changes.
2) *Security*: The mobile VPN should enforce a mechanism for authenticating the user, providing encryption of the data traffic along with integrity assurance.

In addition, several sources of the solutions studied make the following demands of a mobile VPN solution:

3) *Application Session Persistence (ASP)*: Open application connections remain active when the network connection changes or is interrupted, or when the user manually puts the device in sleep mode.
4) *Performance*: Not all data needs to be encrypted. Split tunneling can be employed so that data that is not sensitive may be sent through unencrypted channel enhancing the performance of the VPN as well as preserving MN battery. Moreover, encryption algorithms can be chosen based on user requirements and device status, and may be done adaptively. Adaptive compression techniques may also be applied. Additionally, it should conserve system resources by providing location updates proportional to mobility [38].

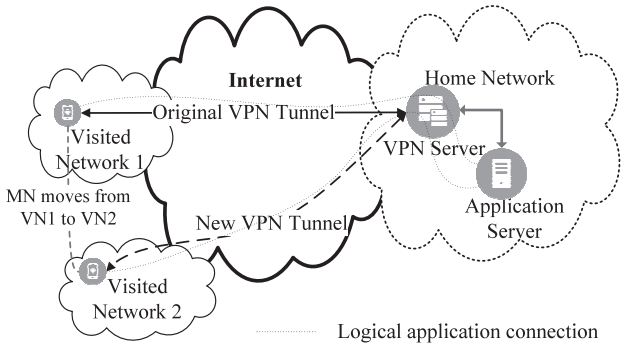

Fig. 3. VPN mobility scenario.

For the purpose of this survey, we assume that a mobile VPN solution is required to maintain the VPN session between a VPN client and a VPN server despite interruption of network connectivity, or when the MN moves between networks and obtains new IP addresses. Figure 3 shows how a MN connected to a home network can travel between networks, get new network information and still appear to maintain the same session from an application perspective. In essence, the main goal with a mobile VPN solution is to provide the application layer transparency to network layer disruptions so as to maintain independence of the end-to-end application sessions from issues caused by mobility.

For conceptual models and design methodology of mobile VPNs, the reader may find [10] useful.

V. Mobile VPN Technologies and Solutions

Mobile VPN is a broad class of protocols that seek to deliver secure IP mobility [3]. An ideal protocol would satisfy requirements set forth in Section IV. Of the several products and protocols that seek to adapt VPNs for mobility, we study a few different approaches. They are discussed in the remainder of this section. Figure 4 shows a taxonomy of the mobile VPN technologies discussed in this section. We also provide Table III to compare these solutions.

A. Mobile VPN Through Network Mobility

In this section we discuss several mobile VPN technologies that support mobility at the network layer.

1) Mobile IPv4 Based VPNs: This type of mobile VPN relies on two protocols IPsec and MIPv4 explained in sections II-C5 and II-C7 respectively. A MN first obtains an IP address for its Home Network and registers it with an HA. When the MN roams and connects to a foreign network, it

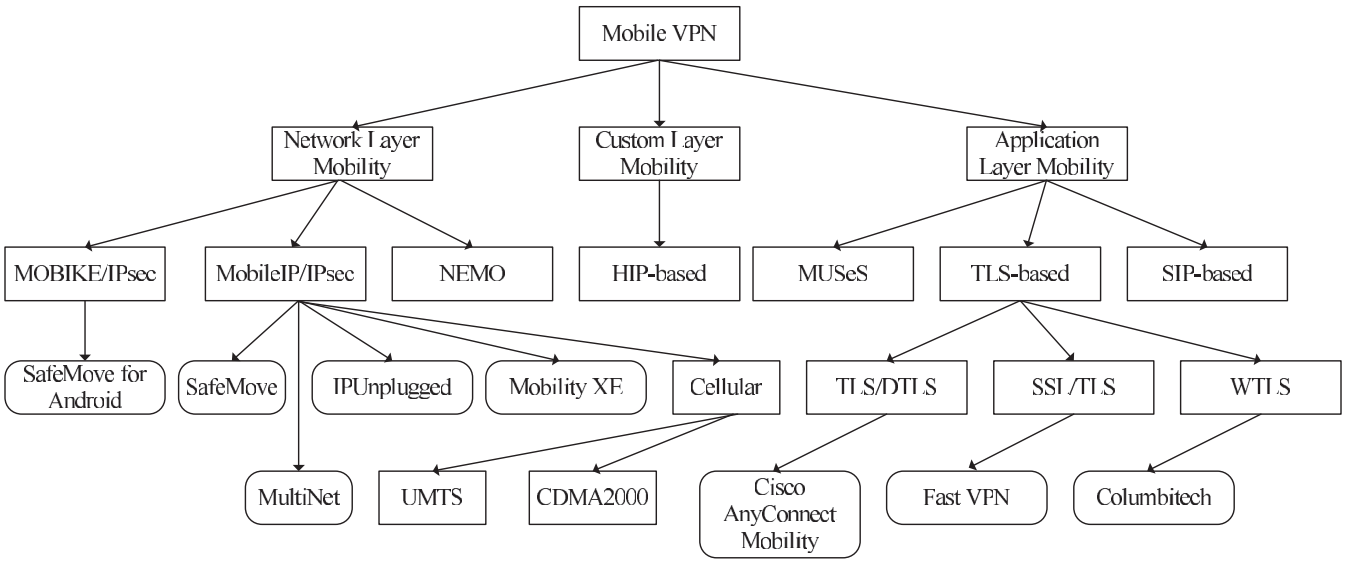

Fig. 4. Mobile VPN technologies and solutions taxonomy

obtains a new IP address and registers with a FA. As shown in Figure 5, the FA establishes an IPsec tunnel between itself and the HA and informs the HA that FA's IP address is the new CoA of MN1. All packets sent from a CN to MN1 go at first to the HA which then send them to the FA though the IPsec tunnel. The FA has the capability to realize which MN these packets are destined to and therefore it will forward them to MN1. This is the compulsory approach of this mobile VPN. A voluntary approach is achieved by having the MN acting as its own FA as the case for MN2 in Figure 5.

The IPsec tunnel is established between MN2 and the HA. MN2 will register its newly obtained IP address with the HA. Just like the compulsory approach, packets destined to MN2 has to be routed to the HA first which causes the triangular routing anomaly.

Authors in [39] present a benchmark for the performance of authentication and encryption algorithms used in IPsec-based mobile VPNs.

2) Mobile IPv4 With Two HA Based VPNs: Incorporating MIP into IPsec based VPN gives rise to several technical issues. When a MN moves away from its home network, it must establish an IPsec tunnel with the VPN gateway using the CoA it received after moving. Since all packets including MIP messages are encrypted by IPsec, the FA cannot decrypt them, thereby rendering it unable to relay the MIP messages [40]. This problem can be avoided by having a mechanism with two HAs, one for internal and one for external networks [41]. The MN would use the internal HA (i-HA) if it is in the home network and an external HA (x-HA) when it moves out of its home network. This device adds another layer of MIP which is underneath IPsec as shown in Figure 6. Upon receiving a new CoA, the IPsec tunnel will not have to be broken, and the FA would be able to decrypt the messages as well. When the MN ventures out to visit a network, it would follow a registration process as in Figure 7.

This solution, proposed in IETF RFC 5265, has several merits. First, there is no modification required to the MIP and IPsec standards. Modifications to the MN are slight [42]. The solution, however, leads to problems determining: *a*) where the x-HA should be placed; *b*) the trustworthiness of the x-HA; *c*) how to protect traffic going to the x-HA; and *d*) the performance impact of having three extra headers to the payload [42].

Benenati *et. al.* [43] build on the work of Feder *et. al.* [44] and use a variant of this IETF solution, along with multiple tunneling protocol standards to offer a transport layer solution across 3G and WLAN. The proposed solution provides a solution for mobility between interconnected WLAN and 3G networks. The authors consider integration of a WLAN system with an existing 3G network either as a wireless Ethernet extension (Tight internetworking) or as complementary to the 3G network (Loose internetworking), with the essential difference being the amount of shared infrastructure between the 3G network and the wireless providers. At a minimum, the Authentication, Authorization, and Accounting (AAA) server is shared between the two technologies. Further, in their solution, the authors of [43] assume that the MN is intelligent enough to engage the proper protocols while using a minimal set of credentials for authentication, which are inherently different for various 3G and WLAN technologies. The specifications in IETF RFC 5265 can be adapted for VPN protocols other than mobile IPsec, provided the MN has IPv4 connectivity with an address suitable for registration. Instead of an IPsec gateway, if an TLS gateway or SSH node was used, it could adapt into mobile TLS or mobile SSH VPN connection [38].

In [45], Dutta *et al.* present a framework named Secure Universal Mobility (SUM) that utilizes the dual HA concept. Their framework suggests a make-before-break approach to reduce the delay incurred while reconstructing the two MIP tunnels and the IPsec tunnel. Based on signal strength, a MN can initialize the handover process before it actually moves from one network to another. This includes activating the target interface and obtaining an IP from the target network. This approach only works if the MN is in the range of both the current network and the future network.

TABLE III
COMPARISON OF MOBILE VPN TECHNOLOGIES

VPN Solution	Mobility Layer	Devices required	Protocol Modifications	Advantages	Disadvantages	MobileVPN Requirements (section IV)
BGP/MPLS Mobile VPN	Network	Diameter Server, Provider Network Server, HA, AAA	Foreign Customer Equipment Address in MIP header	• Adds a mobility quotient to otherwise tried and tested VPN protocols.	• Requires specialized equipment and configuration on part of the ISP. • VPN drops when MN moves, and needs to get setup once again after arriving at a new location.	✗ SNR ✓ Security ✗ ASP ✓ Performance
MIPv4 + IPsec VPN	Network	HA, FA(optional), CN	Original IP packet is encapsulated with a Care-of-Address	• Enables session continuity at end points.	• Connectivity loss potential due to outgoing packets from the HA to the MN being dropped by filters and because FA may not be capable of reading a MN's request to its HA. • IPsec session will be torn down when roaming to a new network, and a new IPsec SA has to be established. • Triangle routing anomaly.	✓ SNR ✓ Security ✗ ASP ✗ Performance
MIPv4 with 2HAs + IPsec VPN	Network	2-HAs, FA(optional), CN, IPsec GW	Original IP packet is encapsulated with a Care-of-Address	• Enables session continuity at end points. • IPsec tunnel is not torn down. • No IPsec tunnel when MN is inside internal network.	• Extra layer of tunneling (x-MIP tunnel). • Triangle routing anomaly.	✓ SNR ✓ Security ✗ ASP ✓ Performance
MIPv6 + IPsec VPN	Network	Native support	Native support	• Mobility compatible with proven confidentiality and authentication capabilities.	• Protocol adds a lot of overhead. • Application persistence is not guaranteed.	✓ SNR ✓ Security ✗ ASP ✗ Performance
MOBIKE + IPsec VPN	Network	Native support	IPsec SA and IKE associated with multiple IPs	• Seamless vertical handover. • Horizontal handover dependent on time to acquire a new IP; and done without tearing down IKE and IPsec SA.	• Application persistence is not guaranteed.	✓ SNR ✓ Security ✗ ASP ✓ Performance
NEMO	Network	Mobile Router (MR), HA	Treats subnets as MN	• Entire subnets can be treated as mobile. • Reduces overhead and improves performance for a few niche applications.	• Does not meet many mobile VPN requirements. • HA is a single point of failure.	✗ SNR ✓ Security ✗ ASP ✗ Performance
MUSeS	Application	Peer-to-peer network running CLOAK	Peer-to-peer IP overlay network. Uses CLOAK to provide encryption and authentication.	• Supports mobility and traffic security.	• Communication between MUSeS middleware and the local applications on a machine are not secured.	✓ SNR ✗ Security ✓ ASP ✗ Performance
Fast VPN	Application	Modified OpenVPN client & server.	Tunnel binding table lookup is based on session ID instead of UDP address.	• Allows OpenVPN client to retain its virtual IP address. • SSL session is not destroyed if roaming happens quick enough.	• Packet loss prevention is based on best effort (not guaranteed). • True application persistancy is not guaranteed. • Fast SSL resumption is not supported.	✓ SNR ✗ Security ✗ ASP ✗ Performance
SIP-based	Application	SIP Proxy server, SIP Registrar, ALG, Diameter Server	IP/UDP/RTP headers replaced with an SRTP header by ALG.	• Excellent for real-time applications that can run on SIP. • Reduced overhead.	• Scalability issues. • SIP security vulnerability	✓ SNR ✗ Security ✗ ASP ✗ Performance
HIP-based	Network/ Transport	HIP enabled APs	New layer that contains cryptographic host identifiers introduced between Layer 3 and Layer 4.	• Native support for mobility.	• Requires use of all HIP enabled devices in network.	✓ SNR ✓ Security ✗ ASP ✗ Performance

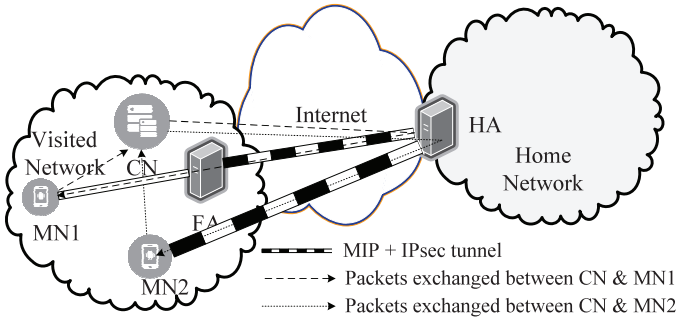

Fig. 5. MIPv4 based VPN: MN1 utilizes a FA, MN2 acts is its own FA

x-MIP	IPsec	i-MIP	Payload

Fig. 6. Mobile IPsec packet format

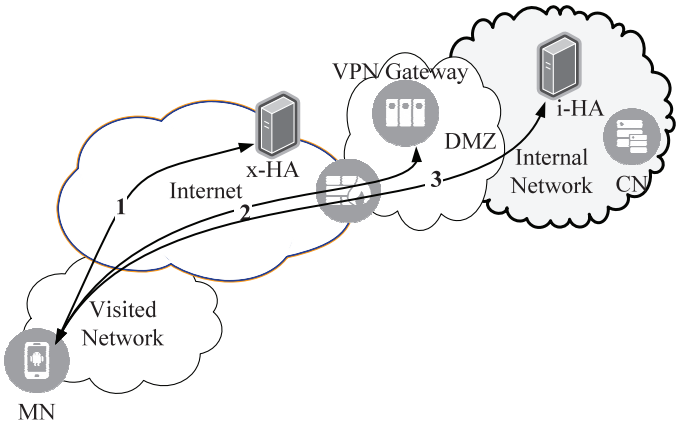

Fig. 7. Mobile IPsec registration

3) Mobile IPv6 Based VPNs: MIPv6 represents a logical combination of IPv6 and MIP, with knowledge gained from the development of MIP (or specifically MIPv4). MIPv6 shares a lot in common with MIP, but naturally offers many improvements over MIP. IPv6 in its native state has features that support mobility, such as the ability for a MN to use its CoA as the source address along with carrying a home address in the IPv6 header. Since every node in an IPv6 network has the ability to interpret this information, there is no longer any need to deploy FA as used in MIP deployments [46]. The functions satisfied by an FA in a MIP network, such as discovery and address configuration in foreign networks are not necessary since MNs can operate in any location without any special support required from its local router.

Figure 8 shows a sample header structure in a MIPv6 when two MNs need to communicate with one another while in visited networks. Note the capability provided in an IPv6 header to incorporate Extension Headers (EH) that can add multiple IP addresses for mobile situations.

4) BGP/MPLS Based Mobile VPN: In a BGP/MPLS based mobile VPN, the MN is registered and authenticated using a Diameter server. The MN generates a MIP registration request when it moves into a visited network [47]. In order to have

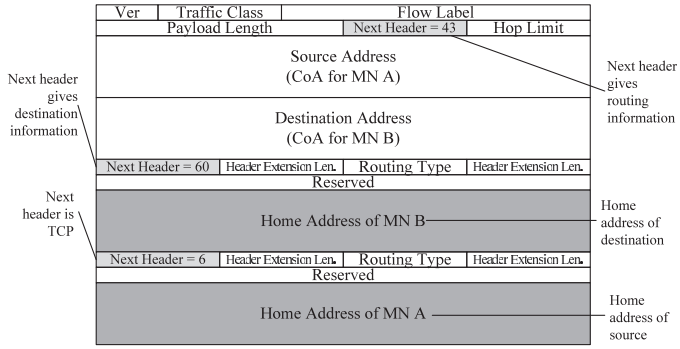

Fig. 8. MIPv6 header

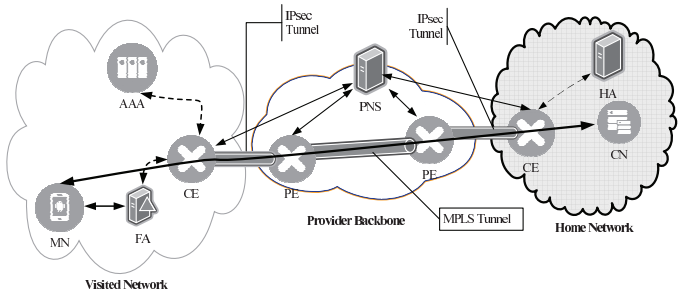

Fig. 9. MPLS based mobile VPN

the registration request to be delivered to the Provider Network server (PNS) in the home network, the address of VPN server replaces the address of HA in the HA field of the MIP registration request message. The authors of [47] assume this address to be pre-configured in the MN. A new field named Foreign Customer Equipment (FCE) address is added to specify the address of the MN's gateway in the visited network so that the PNS can determine the gateway serving the MN. Additionally, in the extension field of the MIP registration request message, the address of the visited network AAA is specified instead of the home network.

When the FA receives the MIP registration request message, it generates a message to the AAA in the visited network for authentication. Upon successful authentication and authorization, the AAA sends a message to the PNS to obtain the address of the HA for the MN. When the PNS receives this message, it prepares an IPsec VPN between the visited network and the provider. After establishing an IPsec tunnel between the PE and the visited network CE, the PE inserts the mapping between the address of the MN and the IPsec tunnel into the Virtual Routing and Forwarding (VRF) table of the corresponding MPLS VPN. The other PEs update their VRF table with the updated routing information, and forward the information as determined by the BGP/MPLS protocol. Figure 9 illustrates how a mobile VPN user obtains access to a VPN from a visited network.

5) MOBIKE-Based VPNs: The IKEv2 Mobility and Multihoming Protocol (MOBIKE) solves an inherent problem with IKEv2 and IPsec when the IP address of a MN changes [48]. MOBIKE provides mechanisms to enable MNs with VPN connectivity using an IPsec tunnel mode to preserve the Security Associations (SA) during a Layer 3 handoff [49].

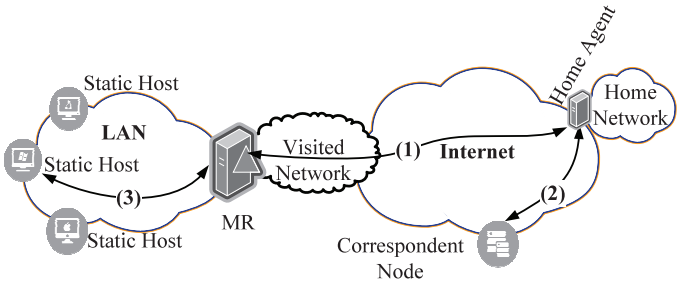

Fig. 10. NEMO network setup

With IKEv1 and IKEv2, the IPsec SAs are created implicitly with the initial IP address of the MN. If the IP address changes, the IPsec tunnel will be torn down and a new SA has to be fully reestablished. MOBIKE enhances this by providing the ability to create SAs (IKE SA and IPsec SA) that are associated with multiple IP addresses. It also provides the ability to update such addresses without having to reestablish the SAs. Such features are very suitable for MNs with multiple network interfaces like cellular and WiFi. The initiator of the connection (usually the MN) and the responder (VPN Gateway) may include one or more ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS notification messages in the IKE_AUTH exchange. During vertical handover, the MN simply notifies the server to use another IP address already agreed upon through the ADDITIONAL_*_ADDRESS notification. For horizontal handover, the MN will simply send an UPDATE_SA_ADDRESSES notification to update the IP address. The server would then perform a "return routability" check before accepting the new address [48].

MOBIKE helps in giving the application sessions persistence only if a handover happens fast enough before the application session or the underlying transport layer session times out. Therefore, applications may not survive long coverage gap where both cellular and WiFi are not available.

6) Network Mobility (NEMO): Devarapalli *et. al.* [50] propose a network mobility (NEMO) protocol that treats entire networks, and not hosts as mobile. A real-world scenario would be a corporate bus. It is conceivable that every person on the bus would want to VPN into the corporate network. Instead of having several individual VPNs, it would make practical sense to have the network on the corporate bus be an extension of the corporate intranet. The hosts in the bus are static with respect to each other, as the network on the bus moves through different access networks. The protocol is essentially an extension of MIPv6 and is illustrated in Figure 10.

A new network device, called a Mobile Router (MR) is introduced in NEMO. The MR registers at the HA as a MN does in a MIPv6 network. But instead of registering one IP, the MR registers one or many subnets. Packets with destination to the network(s) behind the MR are intercepted by the HA forwarded through a tunnel to the network behind the MR.

While NEMO makes minimal extensions to MIPv6, it has the HA as a single point of failure. However, it reduces overhead and improves performance for a few niche applications.

7) Cellular Networks - CDMA2000 Mobile VPN: CDMA2000 is a 3G technology for cellular systems. It is widely deployed in the Americas and in some regions in Asia and East Europe [18]. The main components in CDMA2000 as shown in Figure 11 are:

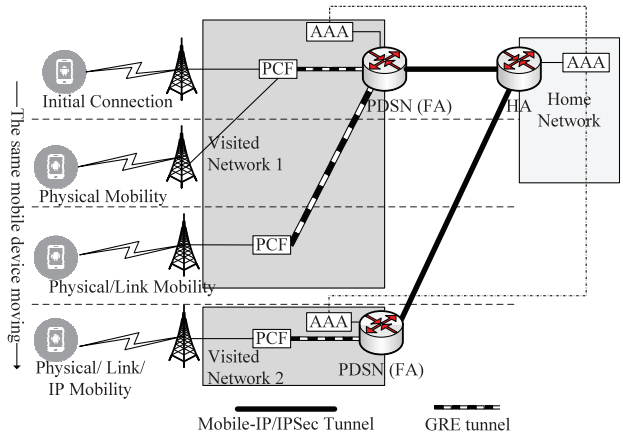

Fig. 11. Mobile VPN in CDMA2000

- CDMA2000 Radio Access Network (RAN). An MN connects to RAN through radio access.
- Packet Control Function (PCF). RAN and PCF communicate through a Radio-Packet (R-P) interface.
- Home and foreign AAA servers.
- Packet Data Serving Node (PDSN) acting as a Foreign Agent (FA). PDSN and PCF communicate through a GRE tunnel.
- Home Agent (HA) which communicates with the FA through a MIP/IPsec tunnel.

When a MN visits a CDMA2000 network, it establishes a PPP session with the PDSN (FA). The PPP traffic is actually encapsulated inside R-P traffic. When it reaches the PCF it decapsulates the R-P traffic to obtain the PPP frames and further encapsulates them inside a GRE packet that gets transfered to the PSDN. The PPP session is terminated at the PSDN. The payload of the PPP frames can then be transfered from the PSDN to the HA via a MIP/IPsec tunnel.

When a MN register with a PDSN, the PDSN delegates the IP assignment to the HA. The HA assigns a dynamic or static IP to that MN. When a MN roams, there are three different levels of mobility:

- The MN leaves the range of one RAN to another. Here a physical layer soft hand-off occurs transparent to the above layers.
- The MN may move far enough to join a range of a new PCF. Here, link layer mobility takes place transparent from layer 3.
- The MN roams to another network. At this point, IP mobility takes place. The MN will register with a new PDSN and the HA will update the mobility binding table resulting in all subsequent traffic being routed via the new PDSN.

8) Cellular Networks - UMTS Mobile VPN: In cellular networks, VPN mobility is provided through the cellular access network consisting of towers and base stations, and mobile VPNs in cellular networks use a combination of GPRS tunnelling protocols (GTP) and IPsec [51] as shown in Figure 12. GTP encapsulates packets over IP/UDP transport paths and provides control messages to setup and modify tunnels.

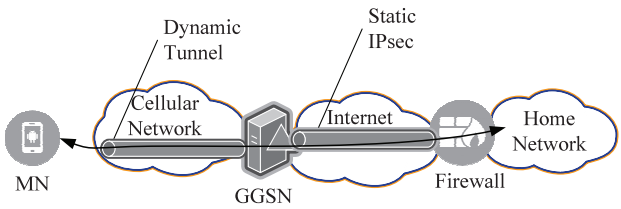

Fig. 12. Mobile VPN in UMTS cellular networks

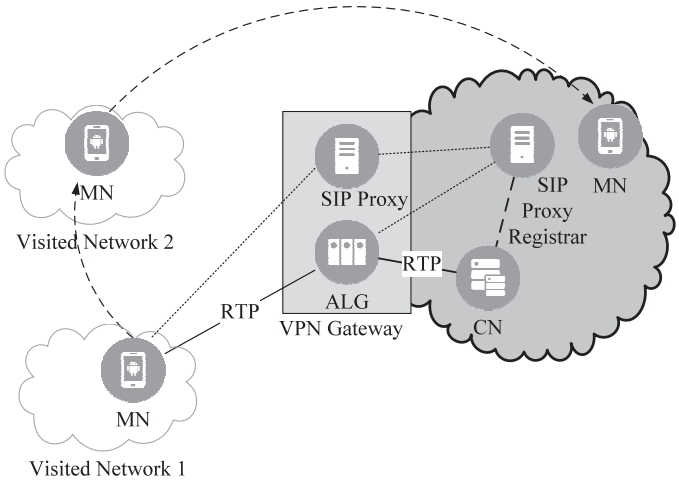

Fig. 13. SIP-based mobile VPN

A MN in such a setup obtains dynamically allocated IPs and are authenticated by the cellular network providers by the Gateway GPRS support node (GGSN) [51]. In non GPRS networks, a node with a different name, but similar functionalities would replace the GGSN. IPsec tunnels are setup between the GGSN and ISPs to transmit traffic to the final destination.

B. Mobile VPN Through Application Mobility

This section discuss mobile VPN solutions that support mobility at the application layer of the TCP/IP protocol stack.

1) SIP-Based Mobile VPN: Huang *et. al.* propose a SIP-based mobile VPN solution for real time applications, tailored to delivering security and mobility to real-time applications [42]. Figure 13 illustrates the proposed SIP-based mobile VPN architecture.

When a MN roams from a home network, a SIP proxy server located within the VPN gateway authenticates the incoming SIP messages, and routes the messages through to another SIP proxy server which is designated as the SIP registrar. An Application Layer Gateway (ALG) interacts solely with a SIP Proxy server, and oversees all the traffic. When the ALG receives an incoming RTP stream from the home network to a host in the Internet, it replaces the IP/UDP/RTP headers with a SRTP header, and deliveries the stream to the destination. Communication in the reverse direction is handled by verifying the validity of the SRTP packet, and by replacing the SRTP headers with a new RTP header. The payload remains unchanged in both directions. Every such bi-directional communication is represented as a session in the ALG.

As and when a MN enters and leaves its home network, it registers its new location with the SIP registrar during initial session setup. Huang *et. al.* use a Diameter service for the registration process. After the MN registers with the SIP registrar, it checks whether there are active sessions in the ALG [52]. If an active session is found, the MN needs to RE−INVITE the CN, where a RE−INVITE is essentially an INVITE message with the same call-ID as the initial INVITE message, with the new contact address of the MN. The RE−INVITE is sent to SIP Proxy in the VPN gateway, which in turn routes the message to the SIP Registrar. If authentication is needed, then the SIP registrar leverages the Diameter server. If the MN is allowed access to the home network, the SIP Registrar uses the ALG to allocate enough resources to guarantee session protection. At this point, the RE−INVITE message is routed to the CN [52].

When a MN returns back to its home network, the messages do not need to go through the SIP proxy in the VPN gateway. So, upon registering its new address with the SIP Registrar and sending the RE−INVITE message, the SIP Registrar will free all the resources previously allocated. The MN can then communicate directly with the CN without going through the ALG [52].

Since the proposed architecture is based on SIP, there is no need to tunnel a packet three times, as is required in the IETF mobile VPN (Section V-A2), thereby significantly reducing overhead. Additionally, the proposed architecture is particularly useful for real-time application as most SIP-based applications are [52]. Performance of the SIP-based mobile VPN seems to indicate that it is especially suitable for real-time applications given the small payload in real-time applications [52].

The SUM framework we discussed in Section V-A2 also utilizes SIP along with MOBIKE as an alternative approach in their mobile VPN framework [45]. The main objective is to achieve a dynamic VPN tunnel establishment in order to use a secure VPN tunnel on demand. For example, a secure tunnel is not needed when the mobile client is inside the internal home network or when it is roaming externally but is not sending sensitive data.

2) WTLS-Based Mobile VPN: One of the most popular commercial mobile VPN products is Columbitech [53], which uses the idea of an application layer solution to add mobility to VPN. By addressing mobility concerns at the application layer, the product liberates the network and transport level connections from having to address mobility, and have those layers working as they were originally designed. The solution relies on recovery mechanisms at the transport layer.

Columbitech [53] splits the client-server connection into three connections as shown in Figure 14. The first connection is a TCP/UDP connection inside the MN between the application client and the mobile VPN client. The VPN client then establishes a session with the VPN server using reliable UDP. Similar to the VPN client, the VPN server establishes a TCP/UDP connection with the application server. This split is used to fool the applications in the MN into believing they are connecting directly to the application server, when in reality, the application client connection ends at the VPN client.

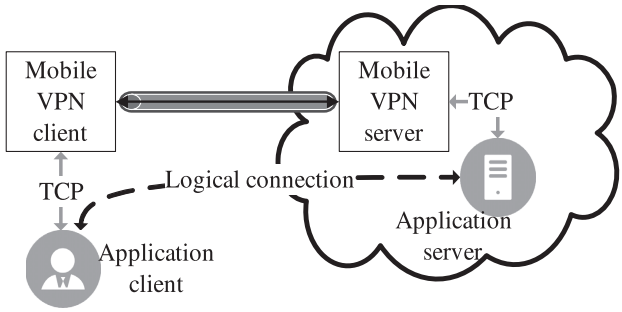

Fig. 14. Columbitech mobile VPN setup.

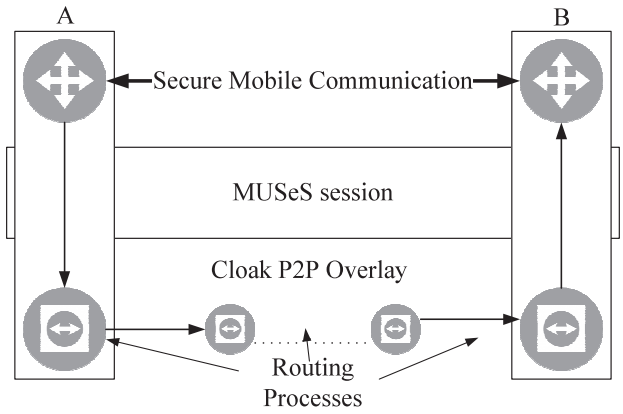

Fig. 15. MUSeS Setup.

When the VPN client receives an application request to connect to an application server, mobile VPN will intercept that request and ask the VPN server to connect to the application server. After learning that the VPN server has completed setup with the application server, the mobile VPN client will inform the application that the end-to-end connection to the server is completed successfully. The mobile VPN server and client setup the VPN session using WTLS. In addition, the system supports multiple VPN servers with a multiplexer that can distribute the load to the VPN servers. When a VPN server experiences failure, all connected clients will lose their sessions and will have to initiate a new connection with a different VPN server since the system does not provide a transparent way to hand-over the sessions of a failed VPN server to an active one.

3) MUSeS: Ahmat and Magoni [54] suggest a similar application control mobile VPN solution. Their solution, called MUSeS supports both the mobility and traffic security. MUSeS allows user connections to survive disruptions caused due to mobility. Similar to Columbitech [53], MUSeS hides the network disruptions due to mobility from the user by creating a secure session using an application layer abstraction. MUSeS uses a peer-to-peer overlay network called CLOAK [55] above any IP network. Instead of using IPsec or TLS for VPN, MUSeS relies on device identifiers called provided and managed by CLOAK to provide encryption and authentication.

When a MUSeS node generates a packet to send to a remote MUSeS node, the packet makes its way through the underlying CLOAK node via a loop back TCP connection. The underlying CLOAK node routes the packet to the destination through the P2P overlay network. The CLOAK node associated with the destination MUSeS node intercepts the packet and locally forwards it to B. The P2P overlay network ensures therefore the proper routing of MUSeS secured packets over the network. The authors do not however, provide explicit details of the security assurances of this mechanism, instead, stating that MUSeS protects user communications from common traffic attacks because it uses standard cryptographic algorithms. Since the communication between the MUSeS middleware and the local applications on a machine are not secured, the security of this system appears suspect compared to more traditional VPN solutions. Figure 15 shows how a packet is forwarded between the source and the destination.

4) FastVPN: Zúquete and Frade [56] suggest a solution for fast VPN mobility of OpenVPN clients across WiFi hotspots, called *FastVPN*. The goal of FastVPN is to reconfigure an OpenVPN tunnel after a VPN client gets a new IP address post handover to a new network without having to terminate and reestablish the OpenVPN tunnel. This is achieved by updating the VPN tunnel context at the VPN server once the client receives a new IP address. Normally, an OpenVPN server looks up a tunnel context by the VPN client's physical IP address and UDP port. When the client obtains a new physical address due to joining a new network, OpenVPN server will not be able to associate this client with its original tunnel context. This leads to two major side effects: 1) the client will have to reestablish a new tunnel causing unnecessary overhead stemming from tunnel setup and new TLS handshake; and 2) the private IP address obtained by the VPN client in the previous session will less likely be maintained as it will not be released until the previous tunnel context is eliminated by OpenVPN's garbage collection, which only occurs after certain period of inactivity. Reusing the original tunnel context allows for maintaining the same private IP address, and allows for faster tunnel resumption by avoiding the reestablishment of the tunnel from scratch.

Fast VPN reconfigures the original tunnel context by having the client send the original session ID to the VPN server whenever it obtains a new physical IP. Sending the session ID is done in two ways: a lazy approach and an aggressive approach. In the lazy approach the session ID (64 bits) is sent in Initialization Vector (IV) field in all data messages all the time. This works well for CBC cipher-mode as randomness of IV does not improve CBC security [56]. For Cipher Feedback mode (CFB) and Output Feedback mode (OBF), 128-bit IV has to be used since randomness of the IV is a requirement. With 128-bit IV, only the first 64 bits will be constant (occupied by the session ID) while the other 64 bits are random.

In the aggressive approach, the client sends a keep-alive message to the server padded with a clear-text session ID at the end of the message payload. When the VPN server receives such a message, it will not be able to find an entry for the client with the new IP address in the tunnel context table. Thus, it checks the size of this keep-alive message and if it is longer that what it normally is, it detects that this is a reconfiguration message that contains a session ID. The session ID is then used to look up the tunnel context, and if found, the physical IP address associated with this context is updated with the new IP address. Figure 16 shows the format of the reconfiguration

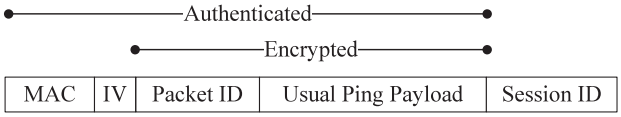

Fig. 16. Reconfiguration message in Fast VPN [56].

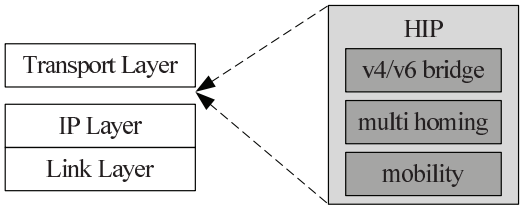

Fig. 17. HIP protocol

ping message. This approach is considered aggressive because the client will keep sending the reconfiguration ping message until a confirmation from the OpenVPN server is received.

Fast VPN minimizes the packet loss but does not avoid it. In addition, there is no mechanism to maintain the application sessions while the MN is experiencing a gap in WiFi coverage. Allowing the VPN client to maintain the same private IP address is quite helpful but such a solution would work only if the client was to move from one WiFi network to another immediately, without experiencing a long gap in coverage that could trigger TCP sessions to timeout.

C. Host Identity Protocol (HIP) Based Mobile VPNs

HIP seeks to change the TCP/IP protocol stack to enhance security, mobility and multi-homing capabilities of today's network. A new layer is introduced between Layer 3 and Layer 4 of the protocol stack that contains cryptographic host identifiers as shown in Figure 17. HIP provides IPsec encryption and enables authentication to a visiting network and to an intranet firewall.

The use of HIP enables Single Sign-on (SSO) functionality in a visited network, where the operator only has to obtain a list of hosts authorized to use the network. During the HIP handshake, the visited network can verify the identity of the MN [57]. As long as the MN can authenticate with a network that has a HIP enabled access point, a VPN can continue to operate seamlessly (except for delay caused by the HIP handshake). Similar to the solutions using the IETF RFC 5265, TLS and IPsec VPN solutions can be configured to run on top of an HIP stack, thereby ensuring VPN functionality [11]. Figure 18 shows a sample HIP based mobile VPN tunnel with the minimum required components.

VI. Comparative Analysis

Mobile VPN based on MIPv4 and IPsec as proposed by IETF meets the main criteria associated with a mobile VPN: it can handle mobility, and is proven to keep data confidential and authenticate the identity of the systems participating in the VPN. However, it adds a lot of protocol overhead. This could potentially result in throughput degradation and adds to

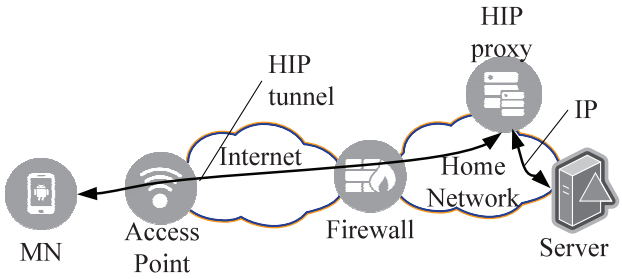

Fig. 18. HIP mobile VPN

configuration complexity. Throughput degradation is especially critical in low-speed wireless networks. An additional concern, depending on the application in question is that this type of mobile VPN does not offer application persistence through network connection drops. Application persistence is only guaranteed if the underlying transport protocol like TCP remains idle [58]. It also suffer from the problem of triangle routing or two-crossing problem in which traffic sent to the MN has to always go to the home agent first even if the MN and the corresponding node are in the same network [58]. Finally, this type of mobile VPN suffers from a performance problem which stems from having to reestablish the security association of IPsec. This problem is addressed in a similar mobile VPN that uses two HA. The IPsec tunnel between the external HA and the FA (can be the MN itself) is persistent since the external CoA does not change during mobility. This method however increase the tunneling overhead by adding an extra MIP layer.

MOBIKE-based VPNs offer native support for multiple network interfaces where switching from one interface to another cause no delays if both interfaces are active. If the interface switched to was not active, the delay incurred is only the delay required to obtain a layer 3 IP address. It also support updating IP addresses during horizontal handover without tearing down IKE and IPsec SAs. Application persistence is guaranteed when there is at least one network available. However, there are no guarantees that applications will survive long coverage gaps.

NEMO is an excellent mobile VPN solution for a niche application. It does not meet many of the requirements of a true mobile VPN solution, but can be used in association with another mobile VPN solution to reduce overhead and enhance efficiency.

While BGP/MPLS based mobile VPN technically makes provisions for mobility, and has obvious VPN capabilities, it falls short of the other solutions, since it requires specialized equipment and configuration on part of the ISP. More than a mobile VPN, it should be considered a stationary VPN for nodes with limited mobility. Whenever a node moves from one location to another, the VPN drops, and with it the application sessions. After arriving at a new location, the VPN needs to be setup once again, thereby causing service interruption.

The encrypted radio communication between the user and the cellular access points in mobile VPN configurations in cellular networks are based on encryption between the user and the mobile network provider [51]. Additionally, there is need for specialized network devices like the GGSN which are owned by entities other than the one the MN is seeking a tunnel to. The

TABLE IV
COMPARISON OF RADIO IP'S MULT-IP VS. COLUMBITECH MOBILE VPN

Mobile VPN requirements (Section IV)		Radio IP's Mult-IP	Columbitech's Mobile VPN
SNR	**Mobility technology**	FVIP + Proprietary mobility protocol	FVIP + TCP-Split
	Seamless Roaming	✓	✓
	Vertical Handover Policy	Pre-set policy defined by the admin.	The user can set limits of maximum amount of data transmitted per medium.
	Automatic Reconnection	No re-registration required within a configurable time period.	WTLS session is resumed upon reconnection.
Security	**Security protocol**	Proprietary	WTLS, DTLS
	Authenticatios Methods	AD, Radius, RSA, EAP, passwords, smart cards, biometrics	AD, Radius, X.509 certificates, WTLS certificates, SecurID, smartcard, biometrics.
	Cryptographic Algorithims	DES, 3DES, AES-256	DES, 3DES, AES-256, RSA, MD5, SHA-1.
ASP	***Applications Sessions Persistence***	FVIP + proprietary algorithms to keep applications sessions alive by buffering application data.	FVIP + 3-way TCP-split + Zero Window messages.
Performance	**Load Balancing**	✓(the master is pre-configured but can be re-elected after failure)	✓, however, the master is fixed. (single point of failure)
	Server Failover	✓, however, application sessions are reset.	✗
	Split Tunnel	✓(decided by the server)	✓(decided by the client)
	Adaptive Encryption	✗, however, encryption is disabled inside the home network	✗, however, encryption is turned off inside trusted zones.
	Adaptive Compression	✗, however, different compression methods can be configured for each pipe	✗, however, compression can be turned off for some communication mediums like dial-up
Supported Clients		Windows XP, 7, 8.1	Android 4.x,Android 2.2-3 rooted, Windows XP/Vista/7/8/Mobile/CE, iOS 5.x

setup of multiple tunnels adds overhead, which could impact performance in low bandwidth networks.

SIP-based mobile VPN [52], [42] has a centralized client/server architecture owing to the nature of the SIP protocol. This inherently brings with it scalability issues. In addition, the solutions are adapted for real-time applications, and may not suitably convert over for other applications. Moreover, it suffers from the security vulnerabilities of SIP, which have been widely studied [59].

TLS-based mobile VPNs and its variants (WTLS, DTLS) are more mobility-friendly than the Mobile IP based solutions. This stems from the fact that TLS is an application protocol and therefore a TLS session is independent to any changes to the network layer i.e. IP changes. In order to support application persistence, TLS-based mobile VPNs rely on establishing a virtual interface that remains active even during network disruption. The virtual interface maintains a fixed virtual IP (FVIP) which an application in the MN can use as a source address. True application persistence (TAP) is not natively supported by TLS-based mobile VPNs. If the MN experiences a long coverage gap, the underlying transport protocol may time out.

Mobile VPNs based on HIP has the potential to evolve into a universal mobile VPN solution, since HIP supports mobility in its native form. However, it requires the use of HIP enabled devices in all visited networks which may not always be feasible, especially in legacy systems. However, HIP VPN solutions appear to lack maturity of other solutions discussed in this paper.

MIPv6 or other MIP type approach which keeps the VPN tunnels active while a MN is visiting other networks only partially solves the issues at hand. Depending on the application, communication disruptions while a MN switches networks might crash the application. For this reason, application session persistence during network disruptions is very important and several of the more accepted mobile VPN solutions like [53] [54] offer the capability to mask network disruptions from the application.

Mobile VPNs that have provisions for application session persistence seem to be the most promising of all mobile VPN options, and appear to be well established in the market. But how these solutions will adapt to IPv6 remains to be seen.

VII. COMMERCIAL MOBILE VPN SOLUTIONS

We surveyed the existing commercial mobile VPN solutions and found seven mobile VPNs available in the market namely: Birdstep's SafeMove, Radio IP's IpUnplugged, Radio IP's Mult-IP, NetMotion's Mobility XE, Columbitech's Mobile VPN, Motorola's MultiNet Mobility and Cisco's AnyConnect Mobile. We reached out to the vendors of these products in order to test and verify their features. We were able to obtain a trial version of Radio IP's Mult-IP and Columbitech's Mobile VPN. In the remainder of this section we briefly present the mobility technology used in these products, and then, we provide a case study for both Radio IP's Mult-IP and Columbitech's Mobile VPN. Table IV summarizes the main features we verified of these two products.

A. Unevaluated Commercial Mobile VPN Solutions

Birdstep provides two mobile VPNs: SafeMove Mobile VPN which is designed for Windows [60]; and SafeMove for Android [61]. SafeMove for Windows implements various

Mobile IP and IPsec RFCs. It uses IPsec to provide security, while providing mobility through Mobile IP [60]. SafeMove provides a module to a predictive vertical handover [62]. SafeMove for Android leverages IPsec to provide security features and MOBIKE to support mobility [61]. Application persistence is only achievable if a change in IP address and the MOBIKE address update occurs before the application or the underlying transport protocol times out.

Radio IP's IpUnplugged utilizes IPsec for security and MIP for mobility [63]. Packet loss during roaming is inevitable in this solution and application sessions are not guaranteed to survive a lengthy period of network unavailability.

Mobility XE is a proprietary mobile VPN. NetMotion does not reveal how the mobility or security protocols are implemented. However, their product description [64] states that Mobility XE is designed to provide true application persistence which means applications can be suspended on network disruption and resumed upon reconnection (without any upper bound on time limit). It also provides seamless roaming through a proprietary protocol. However, [64] indicates that a MN always keeps its virtual IP in order to maintain application sessions. Mobility XE employs link optimization to reduce packet retransmission on the wireless links [65].

Cisco's AnyConnect Mobile is implemented based on TLS and DTLS [66]. When a the VPN client establishes a connection with a VPN server, three sessions are established: a parent tunnel, a TLS tunnel for control traffic and a DTLS tunnel for data traffic. During the VPN establishment phase, the server generates a VPN session token which is then transfered securely to the client via the TLS tunnel. The parent tunnel is inserted in the server's mapping table as the token. The token includes a session ID and is mapped to the assigned private IP address. During roaming, both the TLS and DTLS sessions are torn down, while keeping the parent session alive. When the client regains network connectivity, the VPN session is resumed, by reestablishing both TLS and DTLS tunnels and then presenting the VPN session token to the server, which will reassign the same private IP to the client. TLS and DTLS abbreviated session resumption is not supported. AnyConnect Mobile does not support true application persistence; only if the reconnection occurs before the application TCP session times out.

MultiNet Mobility (MultiNet) is a mobile VPN based on mobile IP and IPsec. It's designed as a client-server model where the MN acts is its own agent. It supports automatic multi-network roaming. Unlike the previously discussed commercial MIP-based mobile VPNs, this mobile VPN provides true application persistence functionality. It allows the VPN to suspend applications during network interruptions and resume them once network connectivity is recovered. MultiNet implements a vertical handover policy allowing administrators to prioritize network selection based on speed or low-cost . It also provide the ability to prioritize application data which can be important for mission-critical applications [67], [68].

B. CASE STUDY I: Radio IP's Mult-IP

We tested and verified the features on Mult-IP in a testbed illustrated in Figure 19. Our test was conducted by observing

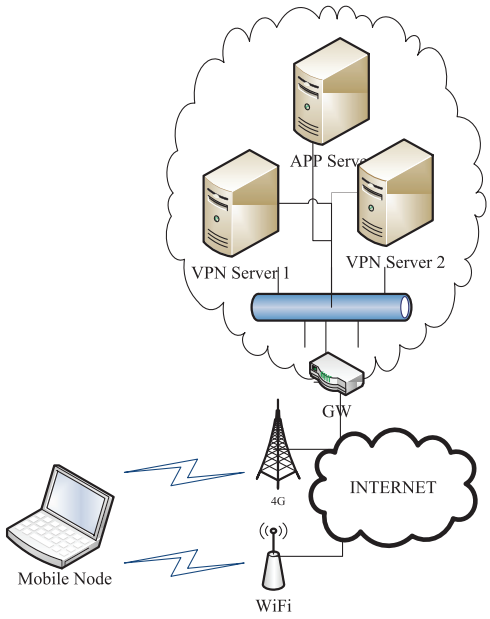

Fig. 19. Mult-IP testbed network diagram

the survivability of the VPN session and the application sessions. We used *netcat* [69] to send messages between the MN and the application server. We also used *iperf* [70] for a more stressful testing by send large volume of data. We began our testing with both WiFi and 4G interfaces enabled, then we performed network interruption events. We disabled WiFi, after that we disabled 4G before we eventually enabled WiFi. We installed the Mult-IP servers version 3.11.0 on Windows 2008, and used Windows 7 for the client. Below, we discuss our observations according the mobile VPN requirements stated in Section IV.

1) Seamless Network Roaming (SNR): Mult-IP was able to sustain the VPN session through all of the network disruption events. The VPN tunnel seamlessly moved from WiFi to 4G, and survived the duration in which both WiFi and 4G were disabled. It automatically reconnected after enabling WiFi.

Mult-IP uses a proprietary mobility protocol that allows the MN to keep the same virtual IP during network interruptions. It also allows for concurrent networking in which the VPN can utilize both WiFi and 4G at the same time. This is done to allow for implementation of a policy by which a VPN server administrator can choose which interface a client application can use. The concurrent networking concept allows the administrator to define up to eight pipes. Each pipe has a pre-set roaming profile prioritizing the network interfaces to be used, as well as configurable timeouts. This allow the mapping of different application to different pipes based on the nature and criticality of the application.

The automatic reconnection does not require the VPN client to re-register with the VPN server if the reconnection occurs within a configurable time period.

2) Security: Mult-IP implements its own proprietary security module which provides authentication, encryption and decryption of all data transmitted through the VPN tunnel. It provides authentication via Windows AD, RADIUS, 801.x

EAP or extended authentication via hard tokens, soft tokens, smart cards etc. The encryption algorithms supported are DES, 3DES and AES.

3) Application Sessions Persistence (ASP): Mult-IP provides true application persistence by 1) allowing the MN to keep its virtual IP despite network interruption events; and 2) buffering application data during network unavailability, while informing the applications that their data has been received by the intended recipient.

When we disabled both WiFi and 4G from the MN, we sent messages from the mobile node to the application server using *netcat*. Using Wireshark [71], we observed that packets sent from the MN node were acknowledged by the application server. Mult-IP client generates these ACK packets on behalf of the application server. Similar behavior happens on the other side when sending the a message from the application server to the MN. Upon reconnection, Mult-IP sends the buffered packets to the remote end. Upon receiving them, Mult-IP on the receiving end changes the ACK number to the last ACK number reported to the application.

4) Performance: Mult-IP provides load balancing. A pool of VPN servers will have one master while the rest are slaves. When a MN tries to establish a VPN session, the request goes to the master server which will redirect it to the server in the pool that has the least number of connected MNs. If each VPN server has the same number of connected MNs, the master will serve that connection. If the master server fails, one of the slave servers in the pool will be elected as a master.

Mult-IP also provides a failover feature by which MNs will automatically connect to a new VPN server in case the original VPN server fails. While this happens seamlessly, active TCP sessions will be reset.

Mult-IP also allows for split-tunneling which allows defining which traffic can go through the VPN tunnel and which can be routed to the Internet directly. The split-tunneling however is controlled by the VPN server and the policy is pushed to the client. Encryption is not adaptive but is eliminated when the MN is inside the home network. Compression is not adaptive either but the administrator can select between LZ4 or arithmetic compression methods for each pipe. Arithmetic compression yields better compression rate and thus recommended for slow networks. LZ4 is recommended for fast networks since it has better computation efficiency.

C. CASE STUDY II: Columbitech's Mobile VPN

We tested and verified the features of Columbitech Mobile VPN in a testbed illustrated in Figure 20. The testbed is similar to Mult-IP except for the introduction of a GateKeeper server which is responsible for load balancing. Our testing was done using Columbitech server version 6.5.0.300 and version 6.5.0.265 for the client. We used the same testing scenarios that we used for Mult-IP as described in Section VII-B. We discuss our observations in relation to the mobile VPN requirements stated in Section IV.

1) Seamless Network Roaming (SNR): In our test, we observed that the VPN client can switch between two 4G and WiFi seamlessly. Unlike Mult-IP, only two physical interfaces

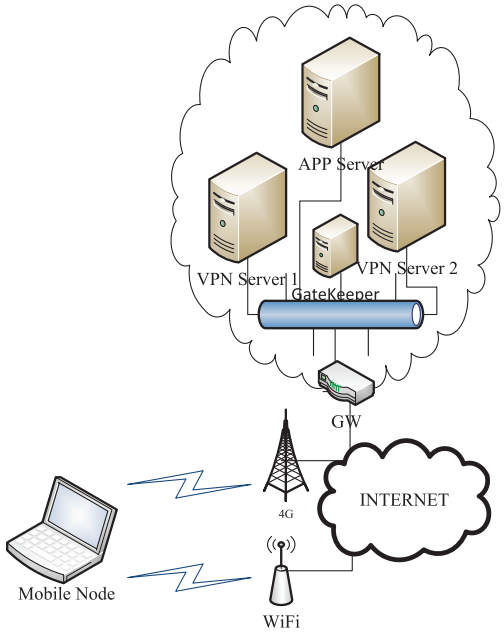

Fig. 20. Columbitech testbed network diagram

are supported by this VPN. The VPN session survived the network interuption events. We disabled both WiFi and 4G for 20 minutes before re-enabling WiFi, the VPN session reconnected seamlessly with an abbreviated WTLS handshake. This mobile VPN does not implement a vertical handover policy. The MN can define a quota for the maximum amount of data transmitted through a certain interface either using daily or monthly rates.

2) Security: Columbitech's mobile VPN uses WTLS and DTLS as its security protocol. It implements several authentication methods like Windows AD, Radius, X.509 certificates, WTLS certificates, RSA SecurID, smartcard and biometrics. For encryption, it uses DES (56 bit), 3DES (112 bit) or AES (up to 256 bit). Key exchange is done using RSA (512-15000 bit), while hashing and signing is done via MD5 (40-128 bit), SHA-1 (40-512 bit) [53].

3) Application Sessions Persistence (ASP): Our testing confirms what we described in Section V-B2 that Columbitech provides true application session persistence by splitting an application's TCP connection into three connections. The merit behind this split is that the connections between the application client and the mobile VPN client; the connection between the VPN server and the application server can be easily maintained to keep both the application client and application server believing they are always connected. This can be achieved regardless of the condition of the TCP connection between the mobile VPN client and the mobile VPN server, thereby maintaining an application session intact. During network disruption events, the VPN client sends a message to the application client indicating the TCP buffer is full and can not temporarily accept any more data. This is achieved by sending an ACK to to the application client with the window set to zero. The VPN server does the same thing to the application server when it tries to send out data. This pauses both ends of the application session until the VPN tunnel is restored. Upon resumption, we observed that a new TCP session was opened to replace the

TABLE V
A SUMMARY OF MOBILE VPN OPEN ISSUES

Issue	Proposed solution	Current status
Inflexibility due to static configuration of VPN clients and servers	Utilizing SDN to allow for dynamic configuration and routing.	No known VPN solutions utilizes SDN.
Applications sessions loss during network interruption	Application persitency using possibly caching and freeze-TCP.	Four commercial VPNs were found to address this problem namely Columbitech, Mult-IP, Multinet, and Mobility XE. No open source VPN is found to address this issue including OpenVPN and MIP/IPsec VPNs.
VPN tunnels often resumed using a full-handshake.	Utilize a lightweight session resumption.	RFC 5723 [72] proposes an extension to IKEv2 that allows lightweight IPsec session resumption. SSL already provide such feature but is not utilized in VPNs like OpenVPN due to security concerns. Columbitech mobile VPN utilizes the session resumption feature in WTLS.
VPN tunnel handover is not a possibility.	Develop a session-hijack-resistant mechanism to hand over a VPN session from one mobile device to another.	No mobile VPNs provide such feature.
Dead peer detection mechanism used in mobile VPNs lead to either delayed detection or excessive network traffic.	Predictive functionality based on profiling the network status of the mobile client.	All mobile VPNs use dead peer detection mechanism.
Battery consumption due to keep-alive messages to prevent NAT servers from dropping a connection, and to detect dead peers,	The use of IPv6 to relieve the need for NAT servers, along with using alternatives to frequent dead peer detection messages.	No mobile VPN is immune to this problem.

middle TCP connection (between VPN client and VPN server). Just like Mult-IP, this mobile VPN allows the client to maintain its virtual IP until the VPN tunnel is no longer needed by the user.

4) Performance: This system supports multiple VPN servers with the use of a GateKeeper (multiplexer) that can distribute the load to the VPN servers. The GateKeeper is a single point of failure; and is not present in Mult-IP.

When a VPN server experiences failure, all connected clients will lose their sessions and will have to initiate a new connection with a different VPN server since the system does not provide a transparent way to hand-over the sessions of a failed VPN server to an active one.

Columbitech allows the client to decide whether to use split tunneling or not. Adaptive encryption is not truly implemented, however, encryption is disabled when the MN is trusted zones. A user can configure the MN to use compression for selected profiles. True adaptive compression is not implemented. A mechanism to dynamically adjust TCP buffer parameters is implemented by observing the RTT in order to optimize TCP performance.

VIII. FUTURE WORK

In this section, we identify six open issues that are worthy of investigation by the research community in order to develop a mobile VPN that meets the requirements and needs of mobile-heavy IT environments. These issues are summarized in Table V.

A. Software Defined Networking-Enabled Mobile VPN

Software-Defined Networking (SDN) has been used recently to allow effective seamless operation of network infrastructure in cloud environments and data centers. Mobile VPNs allow mobile devices to access cloud resources and private networks. With new trends such as BYOD, all applications in a mobile device can access private networks and cloud resources after establishing the VPN tunnel. This may not be desirable to an organization that requires only certain applications to have access to the private resources. SDN-enabled mobile VPN is envisioned to allow the SDN controller to provide VPN on demand by dynamically configuring the routing options a VPN server pushes to the mobile client based on a fine-grained policy that satisfy the MN needs and conform to the organization's objectives. As SDN is still an emerging paradigm, SDN-enabled VPNs are in their infancy. The authors of [73] present a novel approach to use SDN to enhance the flexibility of MPLS VPNs. We believe utilizing SDN in mobile VPNs is an interesting direction for future work in mobile VPNs.

B. Application Persistence

Currently, there is a lack of research in mobile VPNs that add persistence to applications when a mobile user experiences relatively long gap in network coverage. Application sessions most likely will not survive long network disruption. There are a few commercial solutions that address this problem as described in section VII, however, to the best of our knowledge, there is no research-based work published in this area. Open source and free mobile VPNs that provide such feature are non-existent. An open source mobile VPN that allows applications sessions to survive long network coverage gaps, preferably with ability to cache and buffer application data during network disruption will be an interesting direction for future work.

C. Lightweight VPN Tunnel Resumption

TLS-based mobile VPNs will benefit from resuming TLS sessions with abbreviated handshake instead of reestablishing a new TLS session with full handshake upon network reconnection. Once again, there are some commercial solutions that take advantage of such feature but no open source solution has taken advantage of such feature. TLS session resumption is specified in RFC 5077 [74] and RFC 5746 [75]. However, TLS-based VPNs like OpenVPN do not utilize this feature due to triple

handshake attacks [76]. We envision that developing a TLS-based mobile VPN that utilizes TLS session resumption and is resistant to triple handshake attacks will be a good direction for future work on TLS-based mobile VPNs.

D. VPN Tunnel Handover

The evolution of mobile VPN will give rise to new design issues, some of which will include session handover between clients to fully capture the confluence of mobility and cloud computing environments. Since most of the computing power will be in the cloud, a mobile session that could be handed off between devices will greatly enhance features that can be provided to a mobile user. For example, a VPN session could be handed over from a cell phone to a tablet, when the user requires the use of a larger screen. The challenge here is to hand over a VPN tunnel from one device to another in a way that the new device maintains the same virtual IP of the old device so that migrated applications sessions can be resumed on the new device. Another challenge is how to perform the handover in a secure fashion that protects the VPN session from being compromised. Such feature need to be resistant to a well-defined attack threat model.

E. Detection of Network Disruption

All mobile VPN solutions available essentially rely on dead peer detection mechanism to detect the of unavailability of the remote end. This is accomplished by sending control messages, such as a ping message, periodically according to a preset timer. This approach has three drawbacks: 1) The detection of network disruption may be delayed, resulting in avoidable TCP retransmission timeouts which maybe interpreted as network congestion that may lead to dropping the congestion window, 2) when the detection timer is reduced to overcome the problem above, it leads to excessive unnecessary network traffic.

Instead of relaying on dead peer detection, adding predictive functionalities to preempt connection drops to gracefully halt sessions and reduce retransmissions is another potential research area. Using alternatives to dead peer detection in mobile VPNs has been introduced in [77] [78] [79] by using adaptive fuzzy logic and particle filter. These studies introduce mathematical models and simulation but lack empirical evaluation.

F. Battery Consumption and NATing Proxies

Mobile VPN clients sitting behind a NAT server , even when the VPN tunnel is idle, requires continuous sending and receiving of control messages to keep the VPN tunnel alive. This makes the radio module of the mobile device remain active, preventing it from hibernating which results in excessive battery consumption. For example, the radio state machine of android devices transitions between three states: full power, low power and standby [80]. It transition from full power to low power after 5 seconds of the radio being idle. 12 seconds later of idle time, it transitions to the standby state. Most VPNs sends keep-alive messages within 10-second periods by default such as OpenVPN [81]. This prevents the radio module from entering the standby state as it will always toggle between low power and full power states. This issue is not problematic for stationary computers connected to a power source. However, this is a major concern for battery-operated mobile devices. Until today, this remains an open issue for mobile devices that uses IPv4. Mobile VPNs that use IPv6 suffers less from this problem as NATing will not be required, however, dead peer detection mechanism discussed in section VIII-E cause the same phenomenon to exist in IPv6.

IX. Summary

Mobile VPN technology is a powerful information security tool for today's computing environment. Due to complexity of the issues involved, and the numerous possible options available, a mobile VPN solution that custom fits the problem at hand can be devised using a structured methodology [10].

While modifications of IPsec and TLS based client VPNs have their place, they are not optimized for a mobile environment and fail to address the needs for application performance, usability, and productivity.

Finding a mobile VPN solution based on current protocols is not trivial. MIP, and its successor MIPv6, add mobility support to the IP networks. With MIPv6 an integral part of the IPv6 functionality, the narrow waist of the OSI stack is being made friendly to mobility. However, deployments of IPv6 are still lagging behind the IPv4 Internet backbone that we have all become accustomed to. With the inevitable movement of networking to an IPv6 based core, incorporating session mobility based solutions to MIPv6 appears to have the greatest scope of success. However, research on this front is lacking as of this survey. A mobile VPN solution built on session transfer that is IPv6 compatible, we believe will usher the mobility constrained VPN into the mobile age.

In this survey, we presented to the reader a technical background of building-block protocols used in mobile VPNs solutions. We then provided a taxonomy for classifying mobile VPN, and stated the requirements for a true mobile VPN. We surveyed and discussed the state-of art mobile VPN technologies with analytical comparison. We then presented two case studies of commercial mobile VPNs before we concluded our survey with a section that discusses the open issues of mobile VPNs.

References

[1] D. Huang, X. Zhang, M. Kang, and J. Luo, "MobiCloud: Building secure cloud framework for mobile computing and communication," in *Proc. IEEE 5th Int. Symp. Serv. Orient. Syst. Eng. (SOSE'10)*, 2010, pp. 27–34.

[2] K. Heyman, "A new virtual private network for today's mobile world," *Computer*, vol. 40, no. 12, pp. 17–19, 2007.

[3] V. D. Tzvetkov, "Virtual private networks for mobile environments. Development of protocol for mobile security and algorithms for location update," Ph.D. dissertation, Tech. Dept. Comput. Sci., Tech. Univ. Darmstadt, Darmstadt, Germany, 2010.

[4] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. IEEE 19th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM'00)*, 2000, vol. 3, pp. 1537–1545.

[5] Dimension Data. (2014). *Secure Mobility Survey Report* [Online]. Available: http://www.dimensiondata.com/Global/Downloadable Documents/Secure Mobility Survey Findings Report.pdf

[6] D. Saha, A. Mukherjee, I. S. Misra, and M. Chakraborty, "Mobility support in IP: A survey of related protocols," *IEEE Netw.*, vol. 18, no. 6, pp. 34–40, Nov./Dec. 2004.

[7] I. F. Akyildiz, J. Xie, and S. Mohanty, "A survey of mobility management in next-generation all-IP-based wireless systems," *IEEE Wireless Commun.*, vol. 11, no. 4, pp. 16–28, Aug. 2004.

[8] Z. Zhu, L. Zhang, and R. Wakikawa, "A survey of mobility support in the internet," RFC 6301 (Informational), 2011 [Online]. Available: https://tools.ietf.org/html/rfc6301

[9] A. Liotta, D. H. Tyrode-Goilo, and A. Oredope, "Open source mobile VPNs over converged all-IP networks," *J. Netw. Syst. Manage.*, vol. 16, no. 2, pp. 163–181, 2008.

[10] A. V. Uskov, "Information security of mobile VPN: Conceptual models and design methodology," in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT'12)*, 2012, pp. 1–6.

[11] G. Pulkkis, K. Grahn, M. Mårtens, and J. Mattsson, "Mobile virtual private networking," in *Future Internet-FIS 2009*. New York, NY, USA: Springer, 2010, pp. 57–69.

[12] F. Barceló, J. Paradells, F. Setaki, and M. Gibeaux, "Design and modelling of internode: A mobile provider provisioned VPN," *Mobile Netw. Appl.*, vol. 8, no. 1, pp. 51–60, 2003.

[13] B. Lloyd and W. Simpson, "PPP authentication protocols," *RFC 1334(Proposed Standard), Internet Engineering Task Force, obsoleted by RFC 1994*, Oct. 1992 [Online]. Available: http://www.ietf.org/rfc/rfc1334.txt

[14] W. Simpson, "PPP challenge handshake authentication protocol (CHAP)," *RFC 1994 (Draft Standard), Internet Engineering Task Force, updated by RFC 2484*, Aug. 1996 [Online]. Available: http://www.ietf.org/rfc/rfc1994.txt

[15] G. Meyer, "The PPP encryption control protocol (ECP)," *RFC 1968 (Proposed Standard), Internet Engineering Task Force*, Jun. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1968.txt

[16] J. Lau, M. Townsley, and I. Goyret, "Layer two tunneling protocol–Version 3 (L2TPv3)," *RFC 3931 (Proposed Standard), Internet Engineering Task Force, updated by RFC 5641*, Mar. 2005 [Online]. Available: http://www.ietf.org/rfc/rfc3931.txt

[17] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter, "Layer two tunneling protocol "L2TP"," *RFC 2661 (Proposed Standard), Internet Engineering Task Force*, Aug. 1999 [Online]. Available: http://www.ietf.org/rfc/rfc2661.txt

[18] A. Shneyderman and A. Casati, *Mobile VPN: Delivering Advanced Services in Next Generation Wireless Systems*. Hoboken, NJ, USA:Wiley, 2003.

[19] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic routing encapsulation (GRE)," *RFC 2784 (Proposed Standard), Internet Engineering Task Force, updated by RFC 2890*, Mar. 2000 [Online]. Available: http://www.ietf.org/rfc/rfc2784.txt

[20] G. Dommety, "Key and sequence number extensions to GRE," *RFC 2890(Proposed Standard), Internet Engineering Task Force*, Sep. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2890.txt

[21] C. Perkins, "IP encapsulation within IP," *RFC 2003 (Proposed Standard), Internet Engineering Task Force, updated by RFCs 3168, 6864*, Oct. 1996 [Online]. Available: http://www.ietf.org/rfc/rfc2003.txt

[22] C. Perkins, "Minimal encapsulation within IP," *RFC 2004 (Proposed Standard), Internet Engineering Task Force*, Oct. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2004.txt

[23] K. Scarfone, W. Jansen, and M. Tracy, *Guide to General Server Security: Recommendations of the National Institute of Standards and Technology*. Darby, PA, USA: DIANE Publishing, 2009.

[24] S. Kent and R. Atkinson, "IP encapsulating security payload (ESP)," Internet Engineering Task Force, RFC 2406 (Proposed Standard), Nov. 1998 [Online]. Available: https://tools.ietf.org/html/rfc2406.

[25] S. Kent, "IP encapsulating security payload (ESP)," *RFC 4303 (Proposed Standard), Internet Engineering Task Force*, Dec. 2005 [Online]. Available: http://tools.ietf.org/html/rfc4303

[26] S. Frankel and S. Krishnan, "IP security (IPsec) and internet key exchange (IKE) document roadmap," *RFC 6071 (Proposed Standard), Internet Engineering Task Force*, Feb. 2011 [Online]. Available: http://tools.ietf.org/html/rfc6071

[27] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.2," *RFC 5246 (Proposed Standard), Internet Engineering Task Force, updated by RFCs 5746, 5878, 6176*, Aug. 2008 [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt

[28] Open Mobile Alliance. (2001). *Wireless Transport Layer Security. Wireless Application Protocol WAP-261-WTLS-20010406-a* [Online]. Available: http://www.wapforum.org

[29] C. Perkins, "IP mobility support for IPv4, revised," *RFC 5944 (Proposed Standard), Internet Engineering Task Force*, Jun. 2010 [Online]. Available: https://tools.ietf.org/html/rfc5944

[30] J. Rosenberg *et al.*, "SIP: Session initiation protocol," *RFC 3261 (Proposed Standard), Internet Engineering Task Force, updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141, 6665, 6878*, Jun. 2002 [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[31] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," *RFC 3550 (INTERNET STANDARD), Internet Engineering Task Force, updated by RFCs 5506, 5761, 6051, 6222, 7022, 7160, 7164*, Jul. 2003 [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[32] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The secure real-time transport protocol (SRTP)," *RFC 3711 (Proposed Standard), Internet Engineering Task Force, updated by RFCs 5506, 6904*, Mar. 2004 [Online]. Available: http://www.ietf.org/rfc/rfc3711.txt

[33] D. I. Wolinsky *et al.*, "On the design and implementation of structured P2P VPNS." Preprint, 2010. arXiv:1001.2575.

[34] M. Lewis, *Comparing, Designing, and Deploying VPNs*. Indianapolis, IN, USA: Cisco Press, 2006.

[35] T. Takeda, I. Inoue, R. Aubin, and M. Carugi, "Layer 1 virtual private networks: Service concepts, architecture requirements, and related advances in standardization," *IEEE Commun. Mag.*, vol. 42, no. 6, pp. 132–138, Jun. 2004.

[36] P. Knight and C. Lewis, "Layer 2 and 3 virtual private networks: Taxonomy, technology, and standardization efforts," *IEEE Commun. Mag.*, vol. 42, no. 6, pp. 124–131, Jun. 2004.

[37] R. Stanton, "Securing VPNS: Comparing SSL and IPsec," *Comput. Fraud Secur.*, vol. 2005, no. 9, pp. 17–19, 2005.

[38] J. J. A. Rosado, "Mobile virtual private networks," U.S. Patent 8,544,080, Sep. 24, 2013.

[39] A. V. Uskov, "Information security of IPsec-based mobile VPN: Authentication and encryption algorithms performance," in *Proc. IEEE 11th Int. Conf. Trust Secur. Privacy Comput. Commun. (TrustCom'12)*, 2012, pp. 1042–1048.

[40] F. Adrangi and H. Levkowetz, "Problem statement: Mobile IPv4 traversal of virtual private network (VPN) gateways," Internet Engineering Task Force, Tech. Rep. RFC 4093, Aug. 2005 [Online]. Available: https://tools.ietf.org/html/rfc4093.

[41] S. Vaarala and E. Klovning, "Mobile IPv4 traversal across IPsec-based VPN gateways," Internet Engineering Task Force, RFC 5265 (Proposed Standard), Ju. 2008 [Online]. Available: https://tools.ietf.org/html/rfc5265

[42] S.-C. Huang, Z.-H. Liu, and J.-C. Chen, "SIP-based mobile VPN for real-time applications," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2005, vol. 4, pp. 2318–2323.

[43] D. Benenati, P. M. Feder, N. Y. Lee, S. Martin-Leon, and R. Shapira, "A seamless mobile VPN data solution for CDMA2000,* UMTS, and WLAN users," *Bell Labs Tech. J.*, vol. 7, no. 2, pp. 143–165, 2002.

[44] P. Feder, N. Lee, and S. Martin-Leon, "A seamless mobile VPN data solution for UMTS and WLAN users," in *Proc. 4th Int. Conf. 3G Mobile Commun. Technol. (Conf. Publ. No. 494)*. 2003, pp. 210–216.

[45] A. Dutta *et al.*, "Secure universal mobility for wireless internet," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 9, no. 3, pp. 45–57, 2005.

[46] T. Braun and M. Danzeisen, "Secure mobile IP communication," in *Proc. 26th Annu. Conf. Local Comput. Netw. (LCN'01)*, 2001, pp. 586–593.

[47] H. Byun and M. Lee, "Network architecture and protocols for BGP/MPLS based mobile VPN," in *Information Networking. Towards Ubiquitous Networking and Services*, T. Vazão, M. M. Freire, and I. Chong, Eds. New York, NY, USA: Springer, 2008, pp. 244–254.

[48] P. Eronen. (2006). *IKEv2 Mobility and Multihoming Protocol (MOBIKE)* [Online]. Available: http://www.ietf.org/rfc/rfc4555.txt

[49] A. Dutta and H. Schulzrinne, *Mobility Protocols and Handover Optimization: Design, Evaluation and Application*. Hoboken, NJ, USA: Wiley, 2014.

[50] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network mobility (NEMO) basic support protocol," Internet Engineering Task Force, Tech. Rep. RFC 3963 (Proposed Standard), Jan. 2005 [Online]. Available: https://tools.ietf.org/html/rfc3963.

[51] A. Shneyderman, A. Bagasrawala, and A. Casati. (2000). *Mobile VPNs for Next Generation GPRS and UMTS Networks* [Online]. Available: http://esoumoy.free.fr/telecom/tutorial/3G-VPN.pdf
[52] Z.-H. Liu, J.-C. Chen, and T.-C. Chen, "Design and analysis of SIP-based mobile VPN for real-time applications," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5650–5661, Nov. 2009.
[53] "Columbitech wireless VPN technical description," Columbitech, White Paper, Oct. 2007 [Online]. Available: http://www.columbitech.com/img/2008/3/5/16245.pdf
[54] D. Ahmat and D. Magoni, "MUSeS: Mobile user secured session," in *Proc. Wireless Days (WD'12)*, 2012, pp. 1–6.
[55] T. Tiendrebeogo, D. Magoni, and O. Sié, "Virtual internet connections over dynamic peer-to-peer overlay networks," in *Proc. 3rd Int. Conf. Evolv. Internet (INTERNET'11)*, 2011, pp. 58–65.
[56] A. Zúquete and C. Frade, "Fast VPN mobility across wi-fi hotspots," in *Proc. 2nd Int Workshop Secur. Commun. Netw. (IWSCN'10)*, 2010, pp. 1–7.
[57] A. Gurtov, *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Hoboken, NJ, USA:Wiley, 2008, vol. 21.
[58] D. Comer and D. L. Stevens, *Intenetworking With Tcp/Ip*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.
[59] D. Geneiatakis *et al.*, "Survey of security vulnerabilities in session initiation protocol," *IEEE Commun. Surv. Tuts.*, vol. 8, nos. 1–4, pp. 68–81, 3rd Quart. 2006.
[60] Birdstep Technologies. (2013). "Safemove mobile VPN product sheet," Oslo, Norway [Online]. Available: http://www.birdstep.com/media/177925/product-sheet-safemove-mobile-vpn-7.0_web.pdf, accessed on Jun. 15 2014.
[61] Birdstep Technologies. (2014). "Safemove for android," Oslo, Norway [Online]. Available: http://www.birdstep.com/media/197107/product-sheet-safemove-for-android_web.pdf
[62] "Safemove toolkit for android," Birdstep Technol., Tech. Rep., 2012 [Online]. Available: http://www.birdstep.com/media/174709/product-sheet-safemove-toolkit-for-android-web.pdf, accessed on Jun. 15 2014
[63] Radio IP. "Datasheet ipunplugged," Montreal, Canada, 2012 [Online]. Available: http://www.radioip.com/downloadfile/?file=/imports/medias/download/ipu-data-sheet-final-november-2012.pdf, accessed on Jun. 15 2014
[64] NetMotion Wireless. (2014). "Core functionality of netmotion mobility," Seattle, WA, USA [Online]. Available: http://www.netmotionwireless.com/mobile-vpn.aspx, accessed on: Jun. 15, 2014.
[65] Sierra Communications. (2008). "Enabling seamless and secure mobility for the enterprise, an overview of the market drivers, alternatives and birdsteps safemove mobile VPN solution," Mariposa, CA, USA Available: http://www.marcomconsultant.com/samples/b-wpgsm.doc, accessed on Jun. 06, 2014
[66] Cisco. (2015). *Anyconnect FAQ: Tunnels, Reconnect Behavior, and the Inactivity Timer* [Online]. Available: http://www.cisco.com/c/en/us/support/docs/security/anyconnect-secure-mobility-client/116312-qanda-anyconnect-00.html, accessed on Jul. 05, 2015.
[67] Motorola, "Mobile VPN secure connectivity on the move," Motorola Inc., Tech. Rep. RO-99-2157, 2008 [Online]. Available: http://content.motorolasolutions.com/web/Business/Products/Software%20and%20Applications/Mobility%20Software/Mobile%20Application%20Utilities/Multi-net%20Mobility/_docs/_staticfiles/Mobile%20VPN%20white%20paper.pdf
[68] Motorola, "Multi-net mobility mobile VPN solution," Motorola Inc., Tech. Rep. R3-14-2046A, 2008 [Online]. Available: http://content.motorolasolutions.com/web/Business/Products/Software%20and%20Applications/Mobility%20Software/Mobile%20Application%20Utilities/Multi-net%20Mobility/_docs/_staticfiles/Multi-net%20Mobility_SS.pdf
[69] Hobbit. *Netcat* [Online]. Available: http://sectools.org/tool/netcat/
[70] J. Dugan, S. Elliott, B. Mah, J. Poskanzer, and K. Prabhu. *Iperf* [Online]. Available: https://iperf.fr
[71] G. Combs. *Wireshark* [Online]. Available: https://www.wireshark.org
[72] Y. Sheffer and H. Tschofenig, "Internet key exchange protocol version 2 (IKEv2) session resumption," *RFC 5723(Proposed Standard), Internet Engineering Task Force*, Jan. 2010 [Online]. Available: https://tools.ietf.org/html/rfc5723
[73] G. Lospoto, M. Rimondini, B. G. Vignoli, and G. Di Battista, "Rethinking virtual private networks in the software-defined era," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM'15)*, 2015, pp. 379–387.
[74] J. Salowey, H. Zhou, P. Eronen, and H. Tschofenig, "Transport layer security (TLS) session resumption without server-side state," *RFC 5077(Proposed Standard), Internet Engineering Task Force*, Jan. 2008 [Online]. Available: http://www.ietf.org/rfc/rfc5077.txt
[75] E. Rescorla, M. Ray, S. Dispensa, and N. Oskov, "Transport layer security (TLS) renegotiation indication extension," *RFC 5746(Proposed Standard), Internet Engineering Task Force*, Feb. 2010 [Online]. Available: http://www.ietf.org/rfc/rfc5746.txt
[76] K. Bhargavan, A. D. Lavaud, C. Fournet, A. Pironti, and P. Y. Strub, "Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS," in *Proc. IEEE Symp. Secur. Privacy*, 2014, pp. 98–113.
[77] V. Tzvetkov, "Fast detection of disconnection using adaptive fuzzy logic," in *Proc. IEEE Int. Conf. Netw. Sens. Control*, 2007, pp. 828–833.
[78] V. Tzvetkov, "Optimization of update intervals in dead-peer-detection using adaptive fuzzy logic," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. (AINA'07)*, 2007, pp. 266–273.
[79] V. Tzvetkov, "Optimization of mobile updates using particle filter," in *Proc. 3rd Int. Conf. Commun. Netw. China (ChinaCom'08)*, 2008, pp. 915–920.
[80] Android Open Source Project. *Android Developers Training: Optimizing Downloads for Efficient Network Access* [Online]. Available: http://developer.android.com/training/efficient-downloads/efficient-network-access.html, accessed on Nov. 14, 2015.
[81] OpenVPN Technologies. *OpenVPN* [Online]. Available: http://www.openvpn.net, accessed on Nov. 14, 2015.

Abdullah Alshalan (S'14) received the B.S. degree (with Hons.) in computer science from King Saud University, Riyadh, Saudi Arabia, and the M.S. degree in computer science from Indiana University, Bloomington, IN, USA, in 2003 and 2009, respectively. While on leave from the College of Computer and Information Sciences, King Saud University, he is pursuing the Ph.D. degree in computer science at Arizona State University, Tempe, AZ, USA. He has 9 years of combined work experience in information security engineering, programming, web development, and teaching. His research interests include computer networks, mobility, information security, and cloud computing.

Sandeep Pisharody (S'13) received the B.S. degree (distinction) in electrical engineering and the B.S. degree (distinction) in computer engineering from the University of Nebraska, Lincoln, NE, USA, in 2004, and the M.S. degree in electrical engineering from the University of Nebraska, in 2006. He is currently pursing the Ph.D. degree in computer science (information assurance) at Arizona State University, Tempe, AZ, USA. He has over 8 years experience in designing, building, and maintaining enterprise and carrier class networks while working in various capacities for Sprint, Iveda, University of Phoenix, and Insight. His research interests include secure cloud computing and software defined networking.

Dijiang Huang (M'00–SM'11) received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, and the M.S. and Ph.D. degrees from the University of Missouri-Kansas City, Kansas City, MO, USA, 1995, 2001, and 2004, respectively. He is an Associate Professor with the School of Computing Informatics and Decision System Engineering, Arizona State University, Tempe, AZ, USA. His research interests include computer networking, security, and privacy. He is an Associate Editor of the *Journal of Network and System Management* (JNSM) and an Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He has served as an organizer for many international conferences and workshops. His research was supported by the NSF, ONR, ARO, NATO, and Consortium of Embedded System (CES). He was the recipient of the ONR Young Investigator Program (YIP) Award.