VOL. 4, NO. 1 – FEBRUARY 2015

VMWARE TECHNICAL JOURNAL

Editors: Rita Tavilla and Curt Kolovson
Guest Editor: Kit Colbert

TABLE OF CONTENTS

1 **Introduction**
Curt Kolovson, Sr. Staff Research Scientist, VMware Academic Program

2 **FlashStream: A Multitiered Storage Architecture in Data Centers for Adaptive HTTP Streaming**
Moonkyung Ryu, Umakishore Ramachandran, Georgia Institute of Technology

23 **Reducing Cache-Associated Context-Switch Performance Penalty Using Elastic Time Slicing**
Nagakishore Jammula, Moinuddin Qureshi, Ada Gavrilovska, Jongman Kim, Georgia Institute of Technology

35 **Introduction – End User Computing**
Kit Colbert, VMware Principal Engineer

36 **The Role of Social Graph in Content Discovery Within Enterprise Social Networking**
Niloufar Sarraf

41 **NoETL: ETL Code Generation for a Dimensional-Data Warehouse**
Michael Andrews

46 **A Framework for Secure Offline Authentication and Key Exchange Between Mobile Devices**
Erich Stuntebeck, Kar-Fai Tse, Chaoting Xuan, Chen Lu, AirWatch

51 **Just-in-Time Desktops and the Evolution of VDI**
Daniel Beveridge

58 **Connectivity and Collaboration in VMware vCloud Suite**
Ravi Soundararajan, Shishir Kakaraddi

65 **Directions in Mobile Enterprise Connectivity**
Craig Newell

Welcome to the latest edition of the VMware Technical Journal (VMTJ), Volume 4, Number 1.

At VMware, we have a very clear and focused corporate strategy: Be the leader in the software-defined data center (SDDC), end-user computing (EUC), and hybrid cloud computing (our VMware vCloud® Air™ service).

This issue of VMTJ contains several papers from our EUC organization, and I am grateful to Kit Colbert for acting as the guest editor for this issue. To quote from his introduction in this issue about the work of our EUC teams:

The EUC team's mission is to enable a secure virtual workspace for work at the speed of life. The reality is that consumerization of IT is bringing more—and more diverse—devices onto company networks. The "one size fits all" one-desktop-per-employee model no longer works. IT now needs to manage a plethora of different devices, enabling rapid delivery of a user's applications and data to all those devices while at the same time ensuring security and compliance. Users, on the other hand, are demanding a seamless, integrated experience. They want information and apps at their fingertips and want to be able to set down one device, pick up another, and start right where they left off. These are some challenging requirements!

This is certainly true. We are in the midst of a major shift in how workers go about their computing tasks in the enterprise, where mobile applications and cloud computing are rapidly becoming the primary modes of computing. Kit's introduction describes the EUC papers in this issue, as well as a paper by Ravi Soundararajan and Shishir Kakaraddi on how social networking concepts can be applied to system performance management.

In addition to the papers from EUC and the Soundararajan/Kakaraddi paper, this issue contains two papers from professors and graduate students from Georgia Tech. The first, "Reducing Cache-Associated Context-Switch Performance Penalty Using Elastic Time Slicing" by Jammula et al., describes a novel hardware/software approach for implementing variable time slicing to minimize the context-switch overhead associated with cache-warmup slowdowns that can impact certain workloads, particularly in virtualized environments. The second, "FlashStream: A Multitiered Storage Architecture in Data Centers for Adaptive HTTP Streaming" by Moonkyung Ryu and Professor Umakishore Ramachandran, describes a design for a storage system that is optimized for video streaming. This paper is an expanded version of a paper that appeared in ACM Multimedia 2013.

We take great pride in the work of our talented engineers, and we appreciate the excellent work and significant contributions of our colleagues in academia. As always, we welcome your comments on this issue of the VMware Technical Journal.

Curt Kolovson
Sr. Staff Research Scientist
VMware Academic Program (VMAP)

Directions in Mobile Enterprise Connectivity

Craig Newell
VMware Inc.
craign@vmware.com

Abstract

One, if not the most important, attribute required for the enterprise use of mobile devices is access to network-accessible resources offered within the enterprise. Physical connectivity has been made widely accessible, with convenient and affordable Internet access available to mobile devices via technologies such as UMTS/LTE and WiFi. The predominant technology to enable this access to enterprise resources is the virtual private network (VPN) [17]. This paper describes some of the recent mobile-device VPN architectures and presents a proposal for future evolution: the "per-app" VPN with microsegmentation.

1. Motivations for VPN

To understand the currently available mobile VPN architectures, the driving factors behind the development should be examined. These include, in no particular order:

- Reachability
- Persistence
- Security
- Usability

1.1 Reachability

The architecture of the Internet is conceptually very simple, with every node directly addressable and accessible from all other nodes. This can be represented in the mobile device accessing enterprise services directly, as in Figure 1.

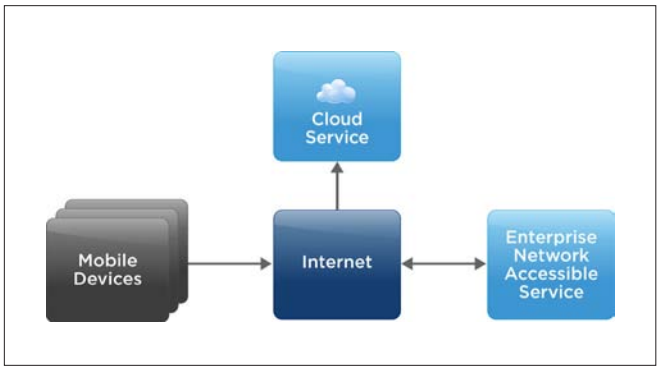

Figure 1. Simplistic Network Model

Due to the shortage of IPv4 address space and the easy availability of private IPv4 addresses [9], Network Address Translation (NAT [13]) has been extensively deployed, removing the directly addressable characteristic of the original Internet architecture. This has resulted in a logical network for mobile devices, as in Figure 2.

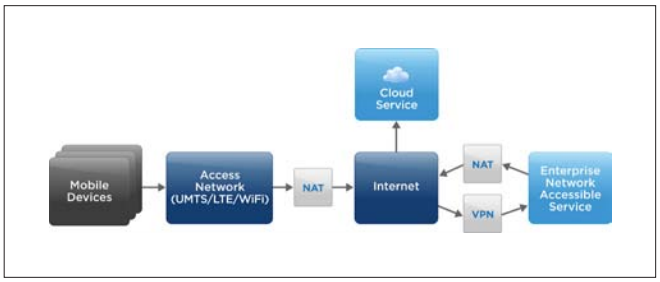

Figure 2. Actual Network Model

NAT generally only supports connections initiated from the "private" side to the "public" side using a subset of IP protocols (normally just a subset of ICMP, UDP, and TCP). This greatly limits the use of other IP protocols for VPN purposes, often forcing traffic to be tunneled via HTTP/HTTPS, because this is almost always transported for web browser use.

A VPN resolves both the reachability and protocol restrictions of NAT by connecting mobile devices logically to the inside of the "private" network.

1.2 Persistence

Most mobile devices can connect to the Internet via one or more wireless technologies (e.g., UMTS/LTE and WiFi) or wireless and wired in the case of laptop computers (see Figure 3). The exact configuration of these network-access methods dynamically changes with network conditions.

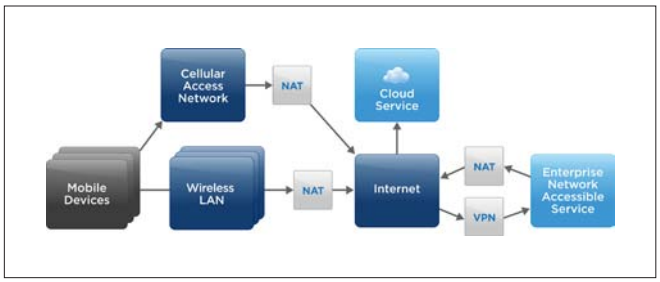

Figure 3. Commonly Available Access Network

The changing network connectivity from the mobile device results in a lack of connection persistence for applications, because the IP addressing differs for each connectivity method. This has been attempted to be addressed at Layer 3 [14] by standards such as Mobile IP [11] or at Layer 4/5 with efforts such as Multipath TCP [12]. These have not been deployed widely and/or end up with mobile applications having to manage the changing connectivity within the application itself.

The tunneling inherent in VPNs can provide persistence for applications, assuming suitable VPN session management has been used—significantly simplifying the required networking logic within applications.

1.3 Security

Information security is often defined as protecting three properties [8]:

- Confidentiality
- Integrity
- Availability

This information security with respect to mobile devices focuses on the confidentiality and integrity properties. The availability aspect is normally ignored because the most common availability issue is the battery going flat, which requires physical device usage protocols to address (e.g., always charging the device every night).

The confidentiality and integrity properties are normally addressed by adding a cryptographic protocol such as TLS [16] under the application protocol. However, practicalities such as enterprise security (e.g., data-loss prevention), availability, and scaling (e.g., TLS offload and load balancing) break the end-to-end security model suggested by the use of TLS (see Figure 4).

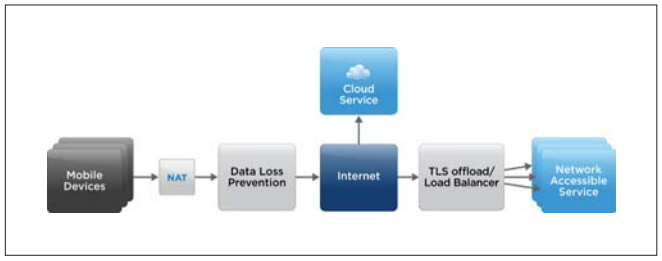

Figure 4. Actual End-to-End TLS Network

This is forcing applications to adopt an additional cryptographic protocol for use within TLS to achieve end-to-end confidentiality and integrity. The use of a VPN that provides an "outer layer" of security can provide hardening to applications using TLS that reduces some of the need for a separate inner encryption layer.

This extra encryption layer added by applications is removing the effectiveness of examining traffic and hence the effectiveness of traffic monitoring for security purposes. If the content of the traffic cannot be monitored, then the ability to segregate network traffic from individual applications to isolated networks is becoming the only practical solution (see Section 2.3).

1.4 Usability

Mobile devices are used for enterprise use for any number of reasons. Sometimes they are required to perform a certain task (e.g., retail stocktaking on a ruggedized portable terminal), but often the major uses are just personal productivity (e.g., checking email and calendar on a smartphone). When used for personal productivity, usability becomes really important. In particular:

- Seamless operation
- Authentication

1.4.1 Seamless Operation

The users of mobile devices have little interest in the complexities of networking beyond the simple monitoring of wireless signal strength. The consumer-driven expectation is that applications just "work" without any additional interdependencies such as launching and tracking the state of a VPN client—necessary so that applications can correctly access their network services.

Enterprise connectivity can be added within applications via software development kits (SDKs), but this is not scalable for third-party applications to maintain variants for each VPN vendor. This is pushing the model of independent VPN clients with "on-demand" initiation to make them transparent to the end user and the application.

1.4.2 Authentication

No one wants to enter a complex password and a two-factor authentication token just to view the next scheduled calendar event on a mobile device. When VPN has been deployed for mobile workers on laptops, cumbersome authentication methods such as external second-factor tokens have been commonly required. However, increasing acceptance by the security community that a certificate stored in a hardware-protected location on a mobile device that has a device lock by PIN code (or fingerprint) is a good balance between security and usability when coupled with simple access policies based on geo-location and time.

Enhanced security can be layered—without impacting the common user experience—via adaptive step-up authentication when simple time and geo-location policies are not flexible enough. (Executives accessing email from the other side of the world in the middle of the night does happen, even if rarely.)

2. Mobile VPN Architectures

VPNs are used in many different scenarios. In this section we review some of the possible architectures used for mobile-device access.

2.1 Client Platform Support

Mobile devices have constrained and restricted operating systems compared to desktop and server environments, to improve robustness and usability. In this section the VPN implementation models of the two most common mobile-device platforms, Apple iOS and Google Android, are detailed.

2.1.1 Apple iOS

Limited VPN support has been available on iOS since iOS 3, when Apple added a built-in VPN client. iOS 5 extended this support to third-party VPN clients with the addition of a VPN plug-in API. Two modes of operation are provided as of iOS 7:

- **Full-device** – In full-device mode, the VPN client can capture and inject Layer 3 frames (IP packets) from the iOS kernel and in doing so is able to tunnel all the traffic from all applications running on the device (see Figure 5) without any modification or configuration of the applications.

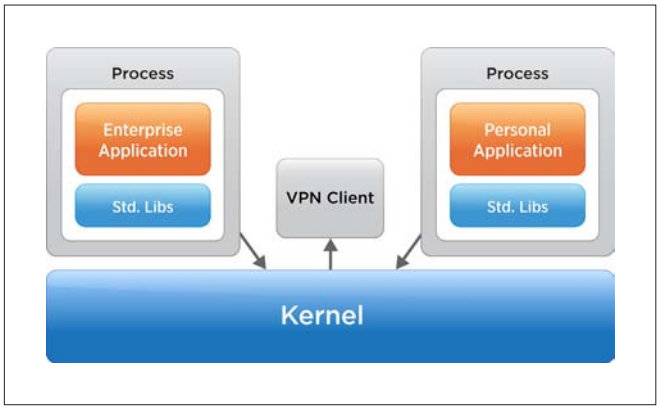

Figure 5. Full-Device VPN

- **Per-app** – iOS also provides a Layer 5 "per-app" VPN model. In this mode, individual applications "managed" via mobile device management (MDM) can have their networking traffic redirected to a third-party VPN client. This redirection is implemented within the standard iOS libraries (`CoreFoundation`) such that applications themselves do not need to be modified and without direct kernel involvement (see Figure 6).

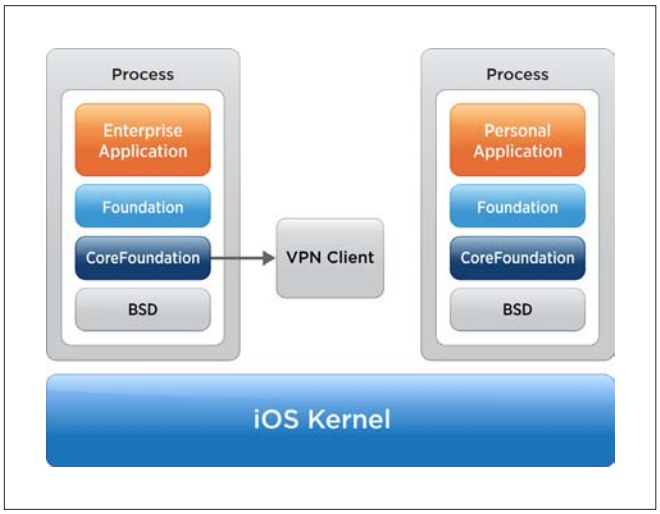

Figure 6. iOS Per-App Device VPN

An alternative implementation of "per-app" VPN on iOS is at the application level using SDKs such as those offered by AirWatch, F5, and NetScaler. However, this requires that applications be explicitly modified during development to support connectivity via the SDK and causes distribution issues with third-party applications. In addition, SDKs are becoming less compatible over time as iOS sandboxes libraries with a large security surface into separate processes (e.g. QuickLook, `WKWebView`).

2.1.2 Google Android

The Android platform has contained a standardized third-party VPN client model since Ice Cream Sandwich [2], with some limited enhancements available as of Android Lollypop [1]:

- **Full-device** – Limited VPN client support has been available on Android since Gingerbread on some devices, but it was not until Ice Cream Sandwich that a standardized API [3] was made available for third-party VPN clients. Since then, "full-device" VPN has been supported just as on iOS (see Figure 5) but with an Android user experience for control and status.
- **"Workspace"** – In Android Lollypop, the VpnService API was extended to support the whitelisting and blacklisting of applications accessing the VPN. With this support, network access via the VPN client can be restricted to a list of applications (a "workspace"). The VPN client still operates at Layer 3 (IP frames), unlike iOS's "per-app" VPN, and the VPN client is unable to distinguish traffic from individual applications (see Figure 7).

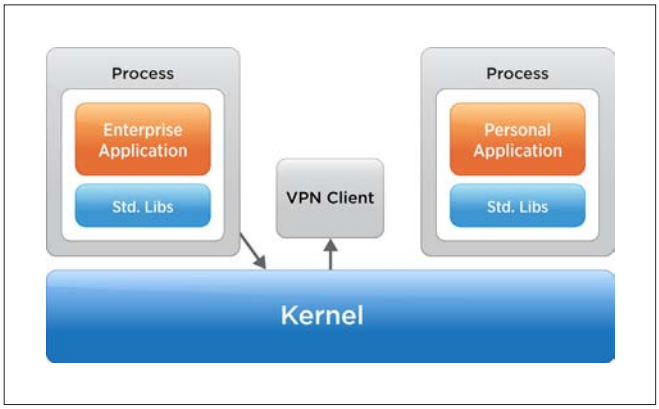

Figure 7. "Workspace" VPN

- **Per-app** – The current Android VPN model does not directly support a true "per-app" model whereby traffic can be identified and controlled on a per-individual-application basis, but it does have enough support to allow this to be reverse-engineered. This can be done by taking a "user space NAT" implementation used to provide "share with host" networking for virtual machines on PCs and adding a filtering layer based on the originating process and finally a Layer 5 VPN client (see Figure 8).

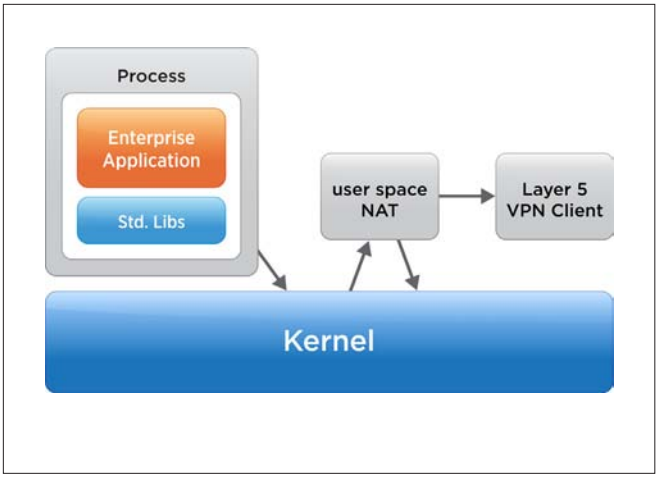

Figure 8. Android "Per-App" Device VPN

2.2 VPN Gateway Architectures

The concept of a VPN gateway is well understood, with the tunneling protocols operating at many different OSI layers (see Figure 9).

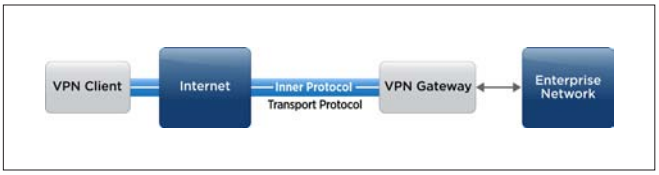

Figure 9. Abstract VPN Gateway Model

In recent times, with the widespread adoption of NAT, the transport protocol for VPN has been restricted in practice to UDP and/or TCP. It is also common to find simplistic firewalls that restrict options further to just TLS on TCP port 443 where UDP, such as encapsulated IPsec or DTLS, is not supported. However, the traffic transported within the inner protocol does not have these limitations and can still be of many different protocols often grouped according to the OSI level of the traffic that is being tunneled.

2.2.1 Layer 3 Gateway

A "classic" VPN tunnels traffic at OSI Layer 3, which is normally IP frames. This is the "full-device" VPN model supported by both iOS and Android. The IP configuration is normally an address from the enterprise network providing support for connections both from and to the mobile device. However, in the case that inbound connections are not required, the deployment configuration can be greatly simplified by integrating a NAT implementation and another private IPv4 address space for the clients.

2.2.2 Layer 5 Gateway

The iOS "per-app" (see Section 2.1.1) VPN model operates at OSI Layer 5, just like the common SOCKS proxy protocol [15]. Adding TLS and authentication to SOCKS, this can be used for VPN without requiring the complexity of configuration and deployment that a Layer 3 solution requires when used without NAT.

2.2.3 Layer 7 Proxy/Gateway

The dominant use of HTTP/HTTPS by browsers and applications has driven the use of the HTTP Proxy protocol as a "VPN." This can be used in a forward proxy configuration (using GET, POST, etc.) for HTTP traffic or in a "tunneling" mode with the CONNECT command. This is especially popular when the VPN client is a SDK that is used by the application.

2.3 Segmented Gateway

In the normal model of a VPN gateway (see Figure 9), all traffic from the VPN gateway is forwarded to a single "private" network. In practice, this is not always the case, and it is normally coarsely segmented into access groups (e.g., different user classes such as contractor and employee). However, as was discussed in Section 1.3, there is great value in segregating traffic according to the originating application. Because there are many different applications that perform the same basic task (e.g., iPhone, iPad, Android), grouping them into "service networks" can help with overall manageability (see Table 1).

SERVICE NETWORK	VLAN #	DESCRIPTION
Device Compromised	1	Captive portal for out-of-policy devices (e.g., jailbroken)
Internet Only	2	No intranet access, only filtered Internet access
Intranet Only	3	No internet access, only internal sites
Finance	4	Application access to Oracle iExpense, etc.
R&D	5	Application access to JIRA ticket tracking, source code
Sales & Support	6	Application access to Internal documentation and knowledge base

Table 1. Example Network Segmentation

In this case, the model of the VPN gateway ends up looking like Figure 10, with the VPN gateway having interfaces on multiple internal segments (VLANS).

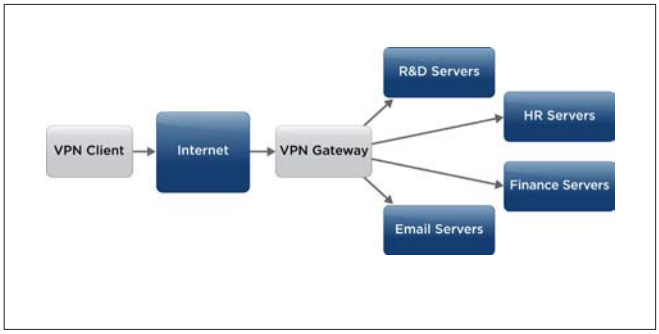

Figure 10. Segmented VPN

3. Summary

Mobile computing is replacing the desktop as the primary computing device for many [7], and the adoption of a mobile-friendly VPN has the potential to make enterprise use as convenient as the desktop. This use of VPN will result in

- Increased usage of mobile devices within the enterprise
- Improved user experience for enterprise employees
- Improved security from segmentation and reduced use of consumer applications for enterprise use
- Reduction of complexity for mobile-application developers, resulting in more enterprise applications

This paper has highlighted the key attributes of VPN for mobile-device use, including: clients with on-demand activation, certificate-based authentication, step-up authentication, and "per-app" segmentation; and gateways with NAT and multiple-network-segment support.

4. Future Work

This review of the current state of VPN architecture for mobile devices has revealed a large opportunity for the optimization of VPN in real-life deployment scenarios.

The inner VPN protocol is often very simple and can be enhanced in many areas, such as low-latency stream flow control, traffic prioritization, and reliability emulation (e.g., dropping UDP frames when traffic is congested over TCP transport). The transport protocol also needs to be examined, because mobile networks are both not-lossy (e.g., UMTS/LTE with reliable delivery) and lossy (e.g., WiFi), suggesting that some level of forward error correction [10] should be added to improve performance (e.g., [4] [5] [6]).

On the gateway side, closer integration with software-defined networking (SDN) in increasing the segmentation granularity (microsegmentation) and with the presentation of user, application, physical location, and so on attributes to the SDN policy layer should improve manageability and security.

Acknowledgments

Although VPN has been used on mobile devices for a long time, it is great to see Apple in iOS pushing VPN with a seamless end-user experience, making it practical for non-technical users. I would like to thank AirWatch for giving me the opportunity to shape a mobile VPN strategy; Tom Corn of VMware® NSX™ security for driving home the future importance of network segmentation; and VMware EUC CTO Kit Colbert for pushing for networking and mobility to work more closely within VMware.

References

1. Google. Android Lollipop – Android Developers, 2014. [Online; accessed 17-Nov-2014].
2. Google. Ice Cream Sandwich – Android Developers, 2014. [Online; accessed 17-Nov-2014].
3. Google. VpnService – Android Developers, 2014. [Online; accessed 17-Nov-2014].
4. Minji Kim, Muriel Médard, and João Barros. Modeling network coded TCP throughput: a simple model and its validation. In *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '11, pages 131–140, ICST, Brussels, Belgium, Belgium, 2011. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
5. Dzmitry Kliazovich, Magda Bendazzoli, and Fabrizio Granelli. TCP-aware forward error correction for wireless networks. In *Mobile Lightweight Wireless Systems*, pages 68–77. Springer, 2010.
6. Benyuan Liu, D.L. Goeckel, and D. Towsley. TCP-cognizant adaptive forward error correction in wireless networks. In *Global Telecommunications Conference, 2002. GLOBECOM '02*. IEEE, volume 3, pages 2128–2132 vol.3, Nov 2002.
7. Steven Norton. A Post-PC CEO: No Desk, No Desktop, 2014. [Online, accessed 11-Dec-2014].
8. Chad Perrin. The CIA Triad, 2014. [Online; accessed 24-Nov-2014].
9. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice), February 1996. Updated by RFC 6761.
10. Wikipedia. Forward error correction – Wikipedia, The Free Encyclopedia, 2014. [Online; accessed 11-Nov-2014].
11. Wikipedia. Mobile IP – Wikipedia, The Free Encyclopedia, 2014. [Online; accessed 13-Nov-2014].
12. Wikipedia. Multipath TCP – Wikipedia, The Free Encyclopedia, 2014. [Online; accessed 16-Nov-2014].
13. Wikipedia. Network address translation – Wikipedia, The Free Encyclopedia, 2014. [Online; accessed 16-Nov-2014].
14. Wikipedia. OSI model – Wikipedia, the free encyclopedia, 2014. [Online; accessed 11-Nov-2014].
15. Wikipedia. SOCKS – Wikipedia, The Free Encyclopedia, 2014. [Online; accessed 17-Nov-2014].
16. Wikipedia. Transport Layer Security – Wikipedia, The Free Encyclopedia, 2014. [Online; accessed 19-Nov-2014].
17. Wikipedia. Virtual private network – Wikipedia, The Free Encyclopedia, 2014. [Online; accessed 11-Nov-2014].

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

 Item No: VMW-TECH-JRNL-VOL4-NO1-2015-COVER-USLET-101 02/2015

http://labs.vmware.com/academic