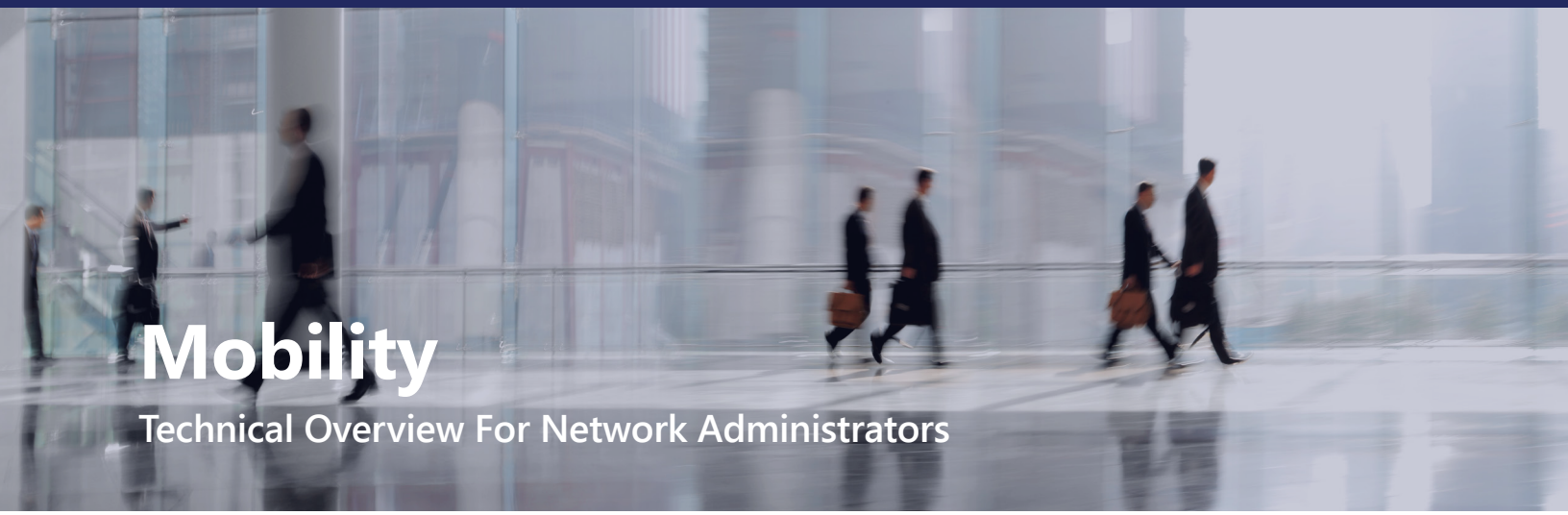


Mobility

Technical Overview For Network Administrators

About This Document

This document discusses operational and technical aspects of the NetMotion Mobility® mobile VPN. It is particularly useful to network administrators who seek a deeper understanding of how Mobility functions before deploying it in their environments. NetMotion Mobility and its technology are protected by copyrights and patents both issued and pending, in the U.S. and in other countries.

Contents

An Overview Of The NetMotion Mobility Architecture

NetMotion Mobility is a highly scalable, software-based mobile VPN. It works with standard network infrastructure and offers high availability and active / active failover. There are two main components of the Mobility VPN: The Mobility server and the Mobility client. They communicate using a proprietary, secure, guaranteed delivery protocol called IMP (Internet Mobility Protocol) and RT-IMP, a version of IMP optimized for real-time traffic such as voice and video. Both IMP and RT-IMP run over UDP. On wireless networks, they provide TCP-like reliability with the performance advantages of UDP.

The Mobility client and server use a transparent, transport level, proxy architecture to isolate all tunneled IP flows from changes in the underlying physical wireless network. This ensures that the TCP connections for tunneled applications remain connected across network roams and other disruptions in network connectivity.
Mobility uses industry-standard encryption and authentication protocols as well as FIPS 140-2 validated and NSA Suite B compliant cryptographic libraries.

Mobility Server

The Mobility server provides secure access to protected network resources for mobile devices running the Mobility VPN client. The server can be run on a virtual machine or dedicated hardware running the Windows Server operating system. The Mobility server supports single or multi-homed configurations. In multi-homed configurations one network adapter must be designated as the internal interface. Unencrypted VPN client network traffic is only sent and received over the internal interface. The server also performs proxy ARP for all active client virtual IP addresses on the internal interface.

Mobility deployments integrate seamlessly with other elements of the data center:

- VPN client IP address assignment is done using proxy DHCP or a static address pool. Clients use the same virtual address for the lifetime of the VPN session, regardless of changes to the underlying physical network.
- Multiple servers form a server pool with failover and load balancing capabilities. Failover and load balancing are automatically configured whenever a server attaches to the pool. With the pool architecture, up to 10 Mobility servers can be managed as a single unit.
- The server provides tools and metrics that a system administrator uses to configure, manage, and troubleshoot remote connections.
- The web-based management console features granular Role Based Access Controls (RBAC): system administrators can grant or deny access to any management or reporting function on a per-user or group basis using either locally defined or Active Directory groups.
- The Mobility server keeps a record of configuration changes, who made them, and when. IT organizations can easily undo configuration errors and demonstrate compliance with security regulations and industry best practices.
- System and connection status information is programmatically available to enterprise dashboards via a RESTful API.
- Mobility user configuration settings and policies can be managed within Active Directory.
- Remote management of Mobility is secure and easily configured to comply with a wide range of corporate security policies. Administrators can allow broad access to the management console, restrict access by source IP address, or restrict access to the machine hosting the Mobility server.
- Mobility supports full server virtualization using VMware ESX or Microsoft Hyper-V.
- IPv6 is supported for client-to-server and server-to-server traffic.

Mobility Client

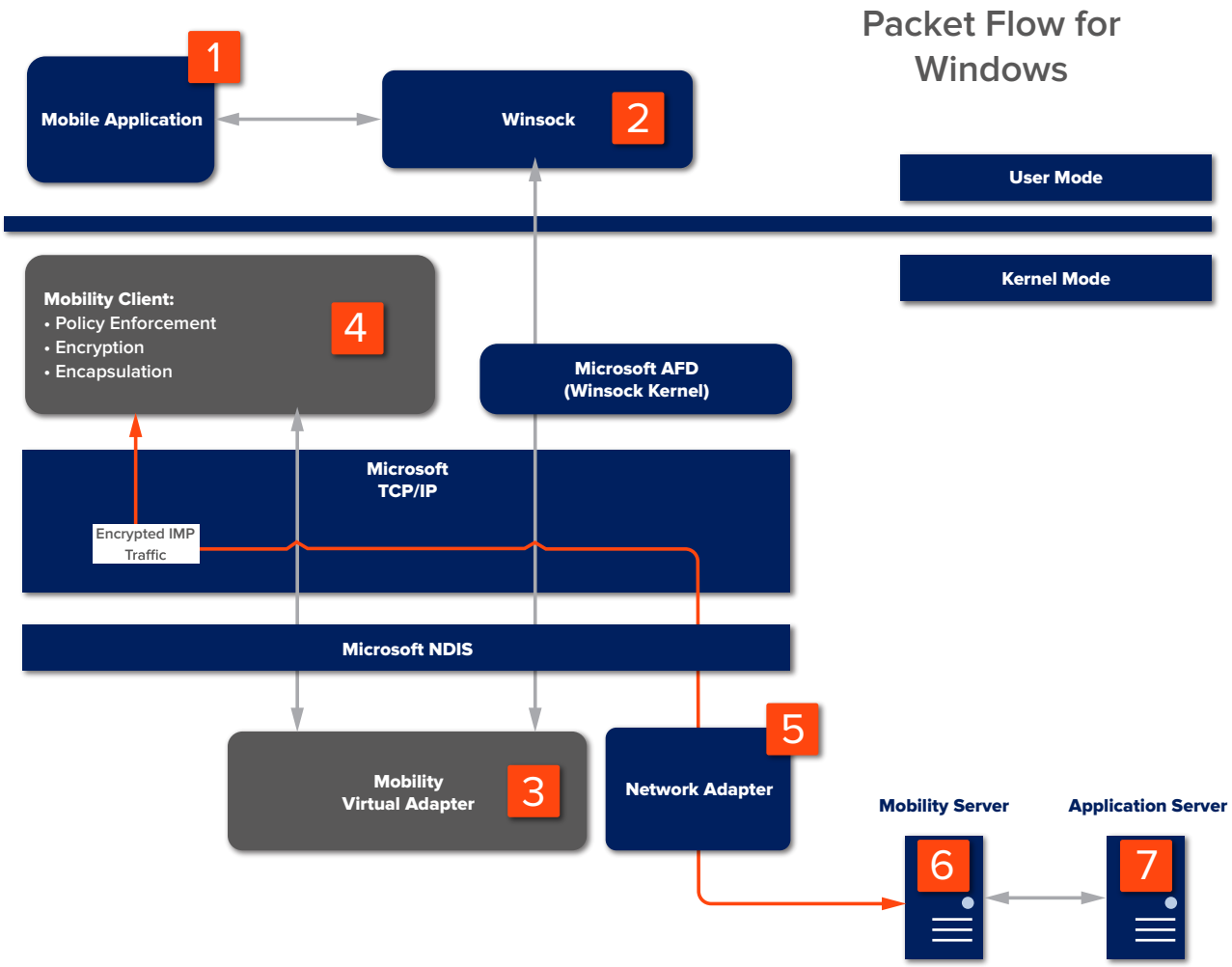
Mobility supports iPhones, iPads, Macs, Android devices, Windows Pro tablets, and devices running Windows.

The Mobility client installs a virtual network adapter on the host operating system. Using this adapter, the client controls which network flows are sent to the network by translating Mobility policy rules into routing rules that are supported by the host operating system. At the virtual adapter, network packets are transformed into remote procedure call (RPC)/IMP messages, encrypted and sent over the Mobility tunnel to the Mobility server.

A more detailed view of the Mobility client architecture is shown in the two diagrams below. Inbound and outbound packets follow the same path through the clients.

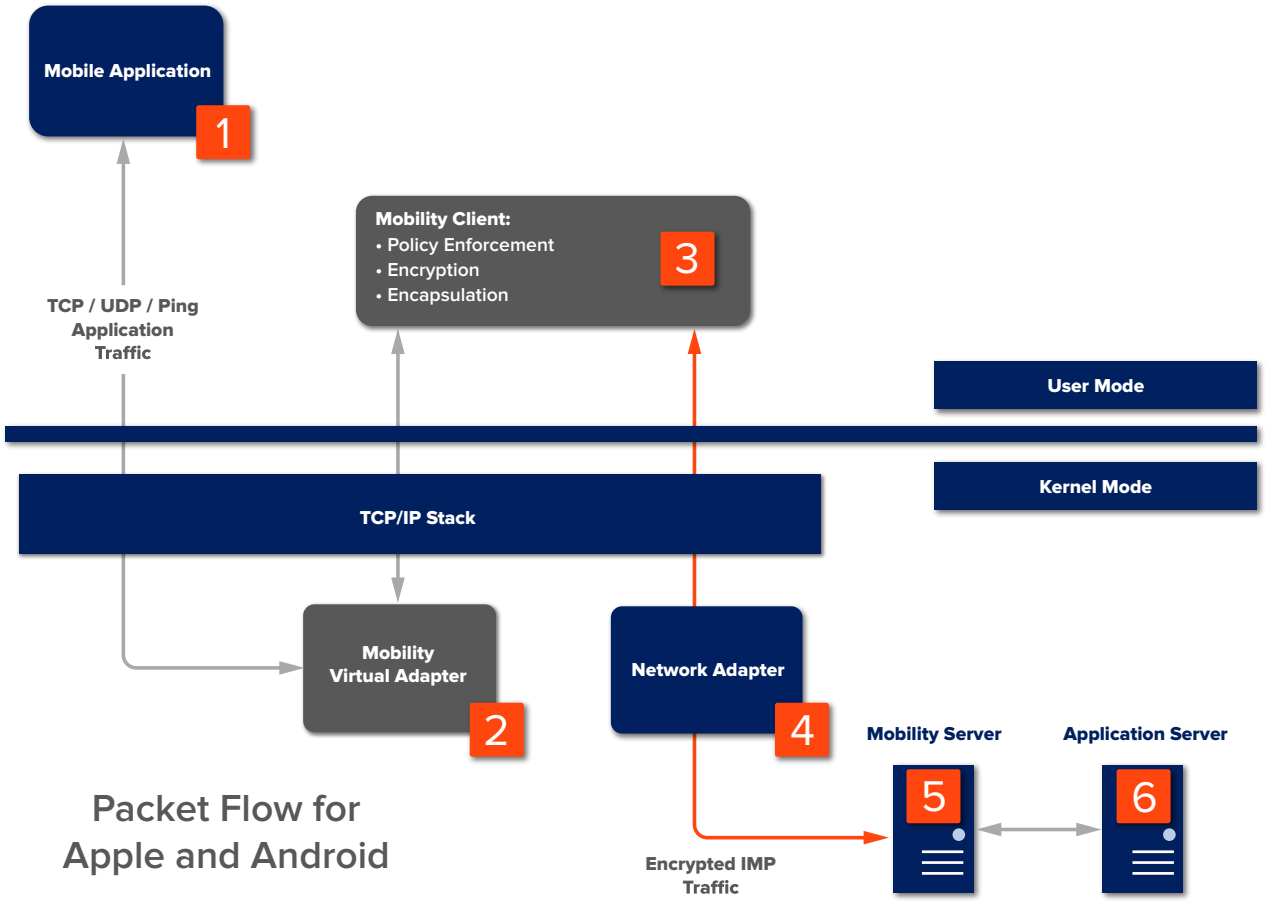
Windows Client Packet Flow

A network application (1) makes a call to Winsock (2), which is translated into a call to the Microsoft TCP/ IP layer. The traffic is then passed to Mobility virtual adapter (3), which then routes the traffic to the Mobility client (4), for policy enforcement, encryption, and encapsulation as an IMP packet. The Mobility client then passes the encrypted, encapsulated traffic to the network adapter (5) where it is sent to the Mobility server. The packet is received by the Mobility server (6), decrypted, and passed on to the application server (7).



Apple And Android Client Packet Flow

A network application (1) makes a socket call, which results in a call to the TCP/IP stack. The TCP/IP stack creates an IP datagram and sends it to the Mobility virtual adapter (2). The virtual adapter passes the datagram to the Mobility client (3). The client make policy enforcement decisions, encrypts the traffic, encapsulates it as IMP traffic, sends it to the network adapter (4), and from there on to the network. The traffic is received by the Mobility server (5) where it decrypted and passed on to the application server (6).



Packet Flow for Apple and Android

Operation

Device Registration

When a Mobility client connects to the server for the first time, the server registers the mobile device's unique permanent identification (PID) number, which the client uses for all subsequent connections. This registration occurs only on the first connection and does not require any action by the user or administrator. The PID is stored in the client system registry (Windows clients) or other secure storage (Android, Mac, iPhone, and iPad clients) and in the Mobility warehouse.

In the warehouse, the PID is associated with the name of the device that generated it. As long as the device name does not change, the server can restore the PID to the client device should it be deleted. This may happen, for example, if the client device's hard drive is re-formatted or a mobile platform is reset to factory defaults. When the Mobility client is re-installed and connects, the server searches for a matching device name. If it finds one, the server will use it rather than create a new one.

Mobility automates assigning policies and settings to new devices based on operating system. Administrators can pre-configure and automatically assign the following:

- Default authentication type based on operating system. Windows, Apple, and Android devices can be assigned different defaults.
- Access policies for networks and applications.
- Device settings such as encryption, network timeouts, and addressing.
- Custom logon notices to remind users of corporate security policies.

In addition to the features used to manage the onboarding of new devices, administrators can specify a list of people authorized to add new devices to the deployment, providing an extra layer of accountability and control.

Device Connection

Mobility can be configured to require certificate based device authentication with PEAP-TLS or EAP-TLS before allowing a user to authenticate. After the Mobility client establishes a connection to the Mobility server using device authentication, it prompts the user to authenticate. If user authentication fails, the device is disconnected.

The device authentication confirms that the device is trusted and guarantees that a secure tunnel has been created to the server prior to user authentication. Enterprises that use unencrypted user authentication methods such as EAP-GTC can enable device-level authentication to ensure that user credentials pass through a secure tunnel. In addition, device authentication can maintain a secure tunnel when no user is logged on to the device (unattended mode). This capability can be used to support remote management of devices and is described further in ***Device Authentication and Unattended Mode***.

The mobile worker can use standard Windows logon credentials to authenticate to the network. The Mobility server authenticates the user against the enterprise's domain using NTLMv2, RADIUS, or RSA SecurID. When configured for RADIUS authentication, Mobility can use the PEAP-MSCHAPv2, PEAP-TLS, EAP-TLS, or PEAP-GTC protocols. PEAP-TLS and EAP-TLS enable support for strong user authentication to public-key infrastructure (PKI) using smart cards and/or user certificates as described in ***Advanced Authentication***.

After authentication is complete, the server and client derive symmetric encryption keys via an authenticated ECDH (elliptical-curve Diffie-Hellman) key exchange and create the secure VPN tunnel. If the deployment supports unattended mode, the administrator has the choice of configuring Mobility to maintain the VPN tunnel after a user logs off, or to tear it down and create a new one.

Persistence And Roaming

The Mobility VPN can roam across multiple networks with different IP address without affecting the state of the VPN tunnel.

The tunnel remains available and application sessions persist during all of the following events, assuming the application itself does not terminate its own connections, or either the client or server sends a disconnect.

- Suspending operation on the mobile device and later resuming it
- Moving to a different network
- Connecting a mobile device over congested, low bandwidth, or high-latency networks
- Encountering interference from microwaves, stairwells, elevator shafts — anything that interferes with radio signals
- Changing networks (for example, from a Wi-Fi to a public carrier network)
- Moving across gaps in coverage such as a tunnel.

In test scenarios, devices have been suspended in the middle of an application transaction, awakened hours later on another network, and the applications resumed exactly where they left off. An inactivity timeout can be configured on the Mobility server to ensure that the server resources allocated for use by inactive sessions are released when they are no longer needed.

User Re-Authentication

Periodic re-authentication confirms that the authenticated user is still in possession of the device. Using the Mobility console, the administrator controls the interval between re-authentication challenges, the grace period during which the user is expected to respond, and whether the user must re-authenticate when the device resumes after suspending or hibernating. Mobility maintains the secure tunnel and application sessions throughout the challenge process and grace period. If the grace period expires and the user fails to re-authenticate, Mobility blocks all further network activity.

Managing The Connection State

The Mobility client sends periodic keep-alive packets to the server during times of network inactivity to let the server know it is still running. If the Mobility server does not receive any packets during a configurable timeframe, it marks the client device as unreachable in the Mobility console. The frequency of these keep-alive packets is configurable and can be decreased to reduce traffic on bandwidth-constrained networks. If a device is unreachable for longer than is allowed by the inactivity timeout, the server closes the session and a new one is created the next time the client connects.

When an application server on the protected network is transmitting data to the Mobility client and the client becomes unreachable, the Mobility server buffers the data until the client is available again. When that happens, the buffered data is forwarded to the client and traffic flows normally. Similarly, when an application on the Mobility client is transmitting data to an application server and the Mobility server becomes unreachable, the client buffers the data. Once the server is reachable, the buffered data is forwarded to the Mobility server and traffic flows normally again.

Mobility preserves the VPN tunnel, applications and data even when the client is unreachable.

Hotspot Detection

Mobility automatically detects when hotspots require logging on to a web portal for access to that network and displays the portal logon page without disabling the VPN. Organizations save on data costs and employees gain access more quickly without compromising security because Mobility preserves the VPN tunnel throughout the portal logon process.

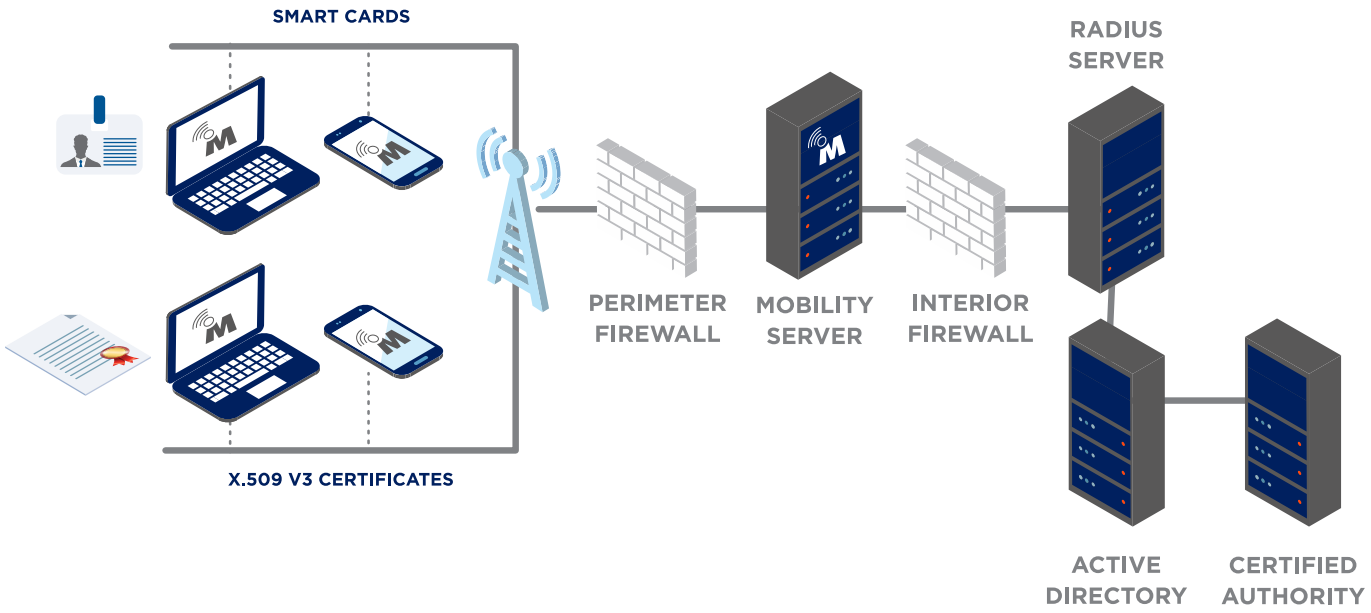
Terms Of Use

Corporate security policies and government regulations often require that users be reminded of and accept terms of use that govern access to the organization's private network before being granted access. Mobility can display a customized notice and require that users accept it before connecting. Notice text is customizable, and different notices can be displayed for different users, devices, or groups.

Advanced Authentication

Mobility is interoperable with NTLM and RADIUS based authentication solutions. In addition to a user name/ password combination, Mobility also supports two-factor authentication methods. Supported methods include:

- Security tokens such as an RSA® SecurID key fob
- Proximity, RFID or contract smart cards readers
- Biometric scanners, such as fingerprint readers
- User and device X.509 v3 certificates in a local certificate store
- Other multi-factor technologies that are compatible with PEAP-TLS, EAP-TLS or PEAP-GTC
- Direct integration is supported with NPS, RSA, SafeNet, Gemalto, , 2FA, Imprivata, Verisign, Vasco, Comodo, Entrust, YubiKey, Digital Persona, Duo-Security, mi-Token and others.



RSA SecurID

Mobility holds a RSA Secured Partner Program Certification for RSA's SecurID signifying that the products are interoperable and that a technical partnership has been established to increase security for joint customers using both products.

In an RSA SecurID installation, the RSA Authentication Agent software runs on the Mobility server. It encrypts the supplied credentials with a one-way hash and then passes them on to the RSA Authentication Manager running on a separate server. The RSA Authentication Manager handles the task of verifying the credentials and passing the authentication status back to the Mobility server. Mobility also supports periodically re-establishing a new SecurID PIN (New PIN Mode) for provisioning new tokens as well as entering successive token codes (Next Token Mode).

Public Key Infrastructure/Radius

Mobility supports standards-based, two-factor authentication using:

- Digital user and device certificates (X.509 v3)
- PKI (for certificate validation)
- RADIUS Extensible Authentication Protocol (EAP) or protected EAP (PEAP) authentication

To implement PKI-based authentication, the Mobility clients and RADIUS server(s) must have certificates installed for mutual authentication. The RADIUS server has the following installed and configured:

- X.509 v3 digital certificate and private key
- Certificate for the trusted Certificate Authority (CA) that signed the certificates on the client devices

The Mobility clients must also have the following installed and properly configured:

- Certificate for the trusted Certificate Authority that signed the RADIUS server's certificate
- X.509 v3 certificate installed and configured or a smart card provisioned with a valid certificate deployed through the enterprise's Certificate Authority
- For smart cards, a reader that supports the Microsoft cryptographic service provider (CSP)

The Mobility server acts as a Network Access Server (NAS) in the RADIUS architecture. The EAP-TLS or PEAP-TLS is used to create a secure tunnel from the client to the RADIUS server. If the client uses password-protected digital certificates stored on the device, the user must enter the certificate password to unlock them. If the certificate is stored on a smart card, the user must enter the associated PIN. The Mobility server and Mobility client then create a mutually-authenticated secure tunnel using the unlocked X.509 v3 certificates.

The Mobility client passes the user credentials to the RADIUS server. The RADIUS server completes the authentication sequence by validating the certificate with the certificate authority and the user's credentials with the authentication server. If the validation succeeds, the RADIUS server notifies the Mobility server, authorizing user access to Mobility services. If the RADIUS server is unable to validate the certificate or the wrong user credentials are entered, it rejects the authentication request and the Mobility server terminates the connection attempt.

Device Authentication And Unattended Mode

Device authentication enables a Windows device and a Mobility server to establish an encrypted VPN tunnel based on device certificates prior to authenticating the user. Device authentication uses the RADIUS EAP-TLS protocol and requires signed X.509 v3 certificates installed on each device as well as on the RADIUS server. If either the device authentication or a subsequent user authentication fails, the device is disconnected.

Device-based authentication can be loosely or tightly tied to a user's authentication. Each user may be assigned one or more specific listed devices, and only logons from those devices are allowed. Mobility offers four authentication modes that define how device and user authentication work together. These can be assigned globally, for groups of devices, or for individual devices.

Device-Authentication Mode	Description
User Authentication Only	The system performs user authentication only. Device authentication is not attempted even if there is a valid device certificate installed.
Multi-Factor	The system performs both device authentication and user authentication before a VPN connection is established. When user authentication is configured to require a username and password or other single-factor authentication method, Mobility uses the device certificate as the second factor in a two-factor authentication solution. Mobility can combine this authentication mode with other two factor user authentication solutions such as smart cards or RSA's SecurID for added security.
Unattended (Windows Only)	The system establishes a VPN tunnel after successfully authenticating the device, thereby maintaining a secure tunnel for management while a user is not logged on. If a user subsequently attempts to log into the desktop and fails, the device is disconnected. After the user logs off, Mobility can either preserve the existing tunnel or force the device to re-authenticate, based on the client settings.
User Required / Device Optional	If the device is configured for device authentication, the system attempts to authenticate the device, but will still allow user authentication to proceed and the VPN to connect even if the device authentication fails. This is used as a test mode, allowing the administrator to transition to device authentication without adversely impacting users..

Because device authentication is independent of the user logon, administrators can be assured that a device is authorized and connected via a secure tunnel, even when a user is not actively logged on. This allows after-hours device management for applying security patches and software updates.

Policy Management settings associated with the device and Network Access Control (described below) remain operational in unattended mode. When unattended mode is enabled, it should always be used in conjunction with policy enforcement to restrict the applications that can access the network.

Traffic Optimizations

TCP/IP packet sizes are not always optimum for wireless transmission. In a wireless environment, error rates rise as transmission power drops, decreasing effective throughput and making errors much more likely in marginal coverage areas. Sending large packets in these situations increases the probability that an entire packet has to be thrown away and re-sent. Smaller packets may increase overall efficiency by decreasing the number of re-sends. Many network administrators are unaware that even WLANs can have substantial numbers of dropped packets and packet errors.

The UDP protocol is much more appropriate for use over wireless networks. It avoids the overhead and inefficiencies of TCP. Instead, Mobility's Internet Mobility Protocol (IMP) and Real-Time IMP (RT-IMP) ride on top of UDP, implementing their own methods for dynamically adjusting both packet sizes and timing parameters to suit the performance characteristics of the network.

Mobility's link optimization techniques include:

- Fragmentation optimizations
- Data compression
- Web image acceleration

Working in concert with each other, these advanced features provide for the efficient movement of data.

Below are descriptions of how these features help in real-world wireless environments.

Fragmentation Optimizations

The fragmentation of IP packets is regarded by the networking community as a necessary evil that should be avoided. It uses resources in a number of ways, including:

- Intervening systems (for example, routers) must do further processing on fragmented frames instead of just forwarding them to their ultimate destination.
- It consumes significant resources on the receiving system to reassemble a frame.
- If any part of the fragmented frame is lost, the entire frame must be retransmitted again.

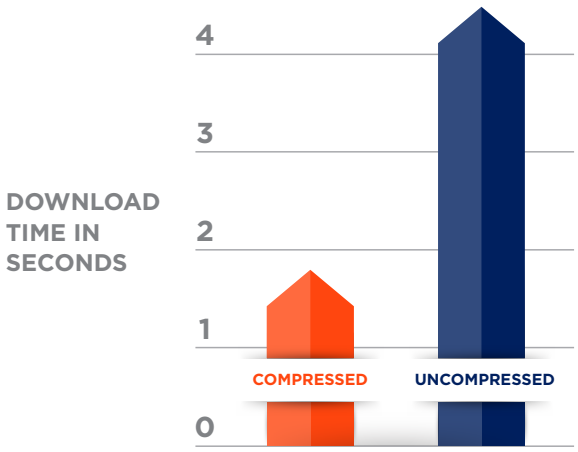
But when roaming from one network to another (with a possible change in the maximum transmission unit, or MTU), fragmentation might be unavoidable. Mobility periodically probes the network to measure its MTU. When the application submits a request to send data, Mobility adjusts the packet size to accommodate the network's measured MTU it before passing the data to the underlying network layer. The data traverses the network as "normal" (unfragmented) frames and does not cause any extra overhead on intermediary systems.

The Mobility message fragmentation algorithm has been optimized to ensure that a minimal amount of resources, both computational and memory-related, are consumed to both fragment and reassemble the message. In the event of a retransmission, the fragmentation is reassessed. If the MTU increased, the frame may be retransmitted in its entirety, again conserving network overhead.

Data Compression

Data compression can improve effective throughput over slow connections or congested network environments, saving customers a significant amount of time and money . With Mobility, administrators can determine when compression should be enabled and whether it should be enabled globally (for all users and devices), for a mobile device group, a specific device, a group of users, or an individual user. Mobility can also be configured to automatically switch compression on or off based on the current interface type, speed, or whether the interface has a monetary cost associated with it so that users can roam between high-bandwidth 802.11 WLANs and lower-bandwidth cellular networks and automatically get the best performance possible.

Mobility compresses data transmitted between the Mobility server and client. It employs the standard algorithms outlined in RFC 1951 (LZ77 Deflate/Inflate). Only the application payload of each frame is compressed; the transport headers are not modified. This allows Mobility to operate through any policy enforcement equipment, such as firewalls and network address translators (NATs). It also operates over any IP-based network.



Effect of data compression on file transfer time (341k text file, WLAN link)

Unlike other compression technologies that are associated with specific applications, Mobility compresses all data that traverses the Mobility tunnel. No modification or reconfiguration of the application is necessary to take advantage of this functionality. While all environments differ, English text usually compresses by a factor of 2.5 to 3; executable files usually compress somewhat less. Mobility also abides by the "no expansion policy" as defined in RFC 1951: transmitted data will not increase in size.

Since the compression process is computationally intensive, a trade-off must be made to provide the maximum benefit to the user. Mobility takes this into account and uses other advanced algorithms that detect link speed, network type and compressibility of recent frames. Based on these factors and the percentage of savings gained by

transmitting the compressed frame instead of the original, Mobility may elect to transmit a frame without compressing it. This reduces the CPU cycles required to decompress the frame upon receipt and provides the maximum benefit to the user. As always, Mobility consumes network and computational resources as efficiently as possible.

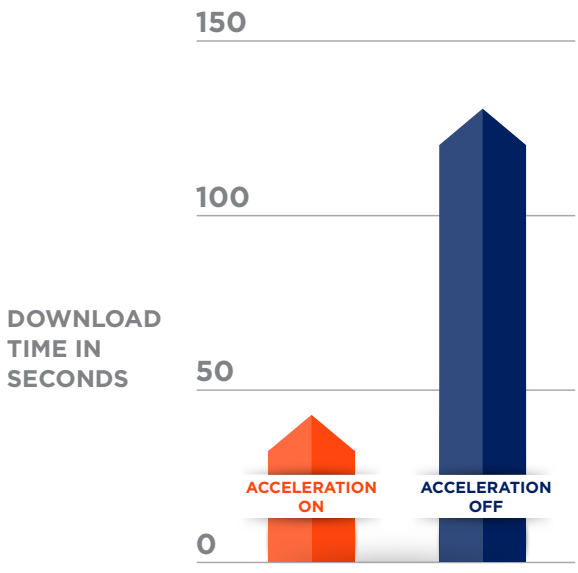
Web Image Acceleration

To speed up web browsing on slow networks, Mobility has the option of compressing images. The level of compression is configurable and co-exists with Mobility's mobile VPN security (in contrast, the acceleration solutions provided by most wireless carriers cannot be used in conjunction with a VPN):

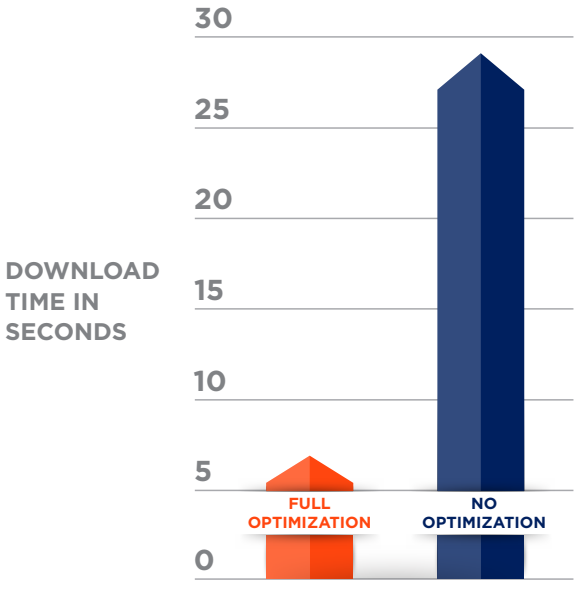
- JPEG images are compressed using the JPEG quality metric. The fastest setting results in a file that is about 28 percent of its original size.
- The number of bits per pixel in GIF images is reduced, which reduces the number of colors. The image is also "flattened": animated GIFs are reduced to one image and textual comments are removed.

Web acceleration is managed in two different ways:

- In the Policy Management module, administrators can selectively turn web acceleration off and on or change the level of compression based on current network characteristics, on a specific application, or on any other available conditions.
- Under Client Settings, web acceleration is available in the core product. The level of compression is configurable; when on, all HTTP traffic on the designated ports will have the images compressed at the configured level.



Effect of web acceleration on file transfer of a JPEG file (4.29MB image, WLAN link with weak 1Mbps signal)



Combined effect of compression and Web Acceleration on a text/graphic transfer (WLAN link with weak 1 Mbps signal)

Traffic Shaping

Quality Of Service

Mobility implements sophisticated Quality of Service (QoS) mechanisms using the Policy Management module to prioritize and shape network traffic. QoS is crucial to maintaining productivity as workers move from high-speed, high-bandwidth networks to lower-capacity, higher-latency networks. For example, while connected to the LAN via Ethernet, performance may be fine for mission-critical enterprise applications or a voice-over-IP call (VoIP), running alongside e-mail, web browsing, and other applications. But on a WWAN, administrators want to prioritize use of the narrower bandwidth and make sure that a web browser and e-mail client do not use capacity needed by enterprise or VoIP applications.

Without QoS, there is undifferentiated access to the tunnel and network.

QoS plays a vital role because voice and video traffic demand a greater share of network resources. These are high-bandwidth applications that depend on timely, reliable packet delivery. Using QoS policies, administrators are able to give these applications the priority they need to function properly.

With QoS, tunnel and network priorities are set by the administrator. Tunnel bandwith and resources prioritized.

Mobility pre-defines five broad classifications for traffic priority: High Priority, Voice, Video, Best Effort and Background. Each classification has a preset configuration for the various QoS-related settings, which provide fine-grained control over traffic shaping, packet queuing behavior, timing, and other related mechanisms. For instance, critical VoIP traffic can be assigned the Voice priority for optimum traffic-shaping, or else assigned Best-Effort or Background priority if it is not an essential use of the network.

Mobility allows for different settings based on specific applications or IP addresses. This degree of control over network use is a hallmark of the Mobility VPN. Conventional VPNs may only allow administrators to shut off non-essential applications. In contrast, with Mobility, applications can continue to run but be throttled back. And when the application(s) with highest priority finish transmitting data, full access to the network tunnel is automatically restored to other applications.

Packet-Loss Recovery (PLR)

Packet-Loss Recovery is part of the Mobility QoS feature set and uses RT-IMP. PLR is beneficial for real-time data streams such as voice or video traffic because they rely on continuous, time-sensitive, sequential packet delivery. Since retransmitting a packet takes a relatively long time, especially on lower-bandwidth wireless networks with higher latency and jitter, lost or dropped packets often result in momentary picture loss or a break in the conversation.

The PLR technique applied by Mobility QoS policies uses a sophisticated mathematical model that adds a small amount of overhead to each packet. When packets are lost, PLR reconstructs them using information from the packets that were received without retransmitting the lost packet. The Packet Loss Recovery level allows administrators to balance the need for recovery against the network conditions and the amount of additional payload added to each packet. A low PLR setting adds less data, and is generally sufficient in situations with minor packet loss. A high PLR setting makes recovery more effective, but also increases the amount of bandwidth used. PLR is enabled through the QoS Policy Management settings and is applied by default for traffic classified as Voice or Video. The default Packet Loss Recovery setting is medium.

Management

Policy Management

The Mobility server maintains user policies and pushes them out to the Mobility clients when it connects and each time there is a change. Policies are enforced on the client. These policies can be specific to the device, user, or group. In addition, separate policies may be defined for enforcement when a device connects in unattended mode; this is a recommended practice for allowing only designated device-management applications to run unattended. Policy updates or modifications are applied in real-time.

Policy rules allow for an extremely flexible and fine-grained control over user and device access to network resources. While the rules are enforced at the device level, the human-readable rule sets are maintained at the Mobility server.

Policy Enforcement

Mobility's Policy Module allows administrators to enforce access policies based on individual users or devices, groups of users or devices defined on the Mobility server, user groups defined in Active Directory, or globally for all users and devices in a deployment. Policy controls determine access to specific application resources and/or networks and under what conditions. These controls extend not only to resources behind the corporate firewall but can be used to specify which applications running on the mobile device are allowed to access the network, preventing blocked application traffic from ever leaving the mobile device. Controls can be based on many different attributes: application name, user name, device name, time of day, network name, SSID, BSSID, Protocol, IP address, port number, device ID, interface name, interface speed, interface type, whether or not the interface is tariffed, interface plug and play ID, Network access control status, operating system version, client version, battery status, an arbitrary registry key value, or an externally defined condition.

The basic Policy Management actions that have an impact on connections are: allow, block, disconnect, passthrough, and bypass traffic. Mobility enforces these actions on the client. When an application attempts to send data over the network, the Mobility client checks the policy list for the application, port, destination address, NAC status, and other parameters to see if action should be taken.

In addition to control over the traffic flow, Mobility supports other actions such as the ability to launch applications, execute command-line statements, set various system parameters, display information to the user, and others. Policy Management makes granular management of wireless bandwidth, security, and mobile productivity straightforward and achievable over networks the administrator does not own or control.

NetMotion Diagnostics

Users who are running both NetMotion Mobility and NetMotion Diagnostics have a powerful tool for visualizing and troubleshooting mobile network errors. The administrator can use policies to have the Mobility client automatically detect impairments in the connection to the server and other network faults. When faults are detected, a diagnostic routine does an end-to-end test of the network adapter status (cellular, WLAN, and Ethernet), the local network routing table and gateway, the GPS state, connection speed, DNS services, the presence of a captive portal, and the availability of other servers in the server pool. In addition, administrators can customize a battery of tests to assess connectivity with other corporate resources using HTTP/HTTPS, Ping, DNS, TCP connect to other servers and services, and traceroute. The results of the tests along with probable root-cause analysis are displayed on the device and are also uploaded to the Diagnostics server. Administrators are then notified that the test has been run and given a link to the results. The number of tests run, the conditions under which they are run, and the frequency are all customizable, and users also have the option to launch a diagnostic routine on their own. This capability is available on iOS, Android, and Windows platforms.

Automatic Client Updates

Mobility provides tools for installing client updates without interrupting users. Administrators can push an update to individual Windows devices, groups of devices, or an entire deployment in real time, or schedule updates during non-peak periods like evenings or weekends. Administrators can even allow users to interrupt an update, or defer it to a more convenient time in the work day.

Administrators have many controls for managing automatic client updates:

- Allow users to defer installing an update and to defer rebooting after installation completes
- Allow users to choose when to install previously deferred updates
- Specify which network types (Ethernet, WiFi, and Cellular) are allowed for downloading updates
- Specify when to start an update and the time frame during which it can occur
- Decide whether to allow the update to be downloaded outside the tunnel
- For users who have opted to defer an update, set the content for reminders and the amount of time between them
- Set the maximum amount of time an upgrade can be deferred and what happens when that time runs out

Client updates for Apple and Android clients happen via their respective stores.

Network Access Control (NAC)

Administrators use the Mobility Network Access Control (NAC) module to create policies that check the security posture of a client device. The NAC module detects whether security products on the client system are enabled and current, and whether the device is configured in accordance with defined policies. If a device is not in compliance with NAC module policies, a variety of actions can be taken, beginning with simple warnings, to enforcing remediation steps, quarantining the device, or disconnecting it. NAC is only available for Windows clients.

The Mobility NAC module differs from a conventional NAC technologies in its ability to maintain worker productivity by customizing the response to a non-compliant client. A worker in the field should not be forced to interrupt their workday because of a minor security concern. Operating system updates and antivirus signature downloads that can take many minutes over a cellular network can be delayed until the end of the working day, or put off until the worker is within reach of a higher-speed link.

As with the Policy module, the Mobility server stores NAC rules and pushes them out to each device when it connects. The Mobility client on each device checks for NAC compliance at connection time and periodically at intervals (the default is five minutes). NAC module policies can specify the following parameters:

Category	Parameters Checked
Antivirus and Antispyware	Specified product installed, real-time protection enabled, signatures current, date and result of the last scan.
File	Specified file either present or not present on the client.
Firewall	Specified product installed and running
Process	Specified application or service running or not running
Registry Key	Keys in the HKEY_LOCAL_MACHINE\section of the registry present or not present, and have the expected values
Windows Update	Auto-update enabled, and specific patches present
Mobility Version	Version of the Mobility client
Operating System	OS version, service pack, processor and other platform information

NAC Enforcement

If a client device fails a NAC policy check, it can be assigned one of four states:

Status	Description/ Action
Warn	The client does not comply with one or more checks in a rule. The Mobility client device is allowed to connect, but the Mobility client displays a warning.
Remediate	The client does not comply with one or more checks in a rule that requires remediation. The action required to bring the device into compliance is determined by Policy Management rules that apply to the remediation level.
Disconnect	The client does not comply with one or more checks in a rule that results in a client disconnect.
Quarantine	The client does not comply with one or more checks in a rule that causes the device to be quarantined. The system administrator must clear the quarantine before the device can connect.

When the status is Remediate the administrator has a number of options, including taking specific action based on the speed of the current connection, time of day, etc. A typical example is as follows: if anti-virus signatures are more than 7 days old, send a reminder message telling the user to update; if they are more than 14 days old and on a WWAN connection, send a reminder to update as soon as possible; if they are more than 14 days old and on a WiFi or LAN connection, download new signatures immediately; if they are more than 21 days old, quarantine the device. The same capability can be used to automatically download and install updates and operating system patches, at a time and in a manner that does not have an effect on worker productivity while protecting the device and the network.

Client Activity / Mobility Console

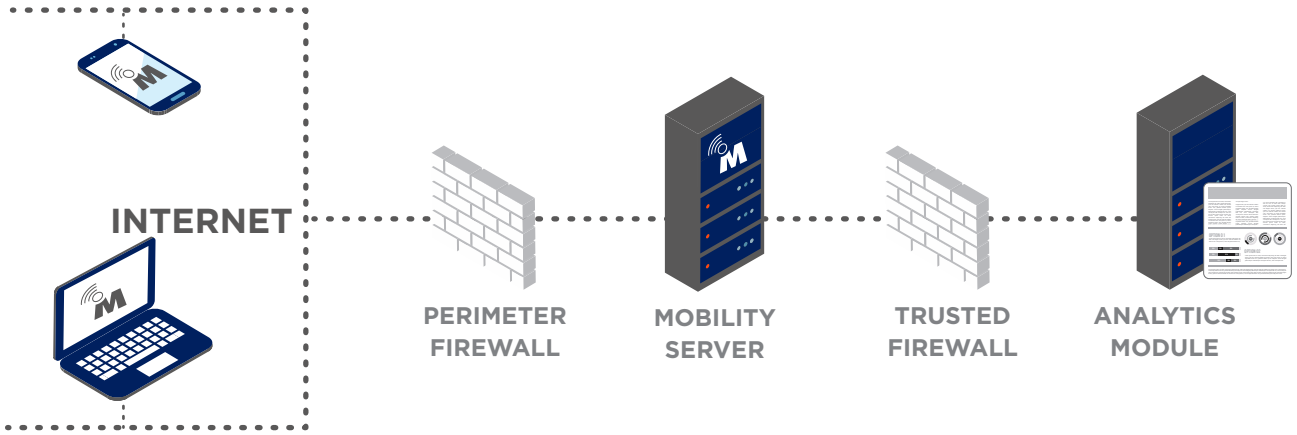
During the initial client connection and periodically thereafter, the Mobility server collects the following data on each active client session. This information gives a detailed view into the health of a mobile deployment:

- Device name
- User name
- Client status
- Authentication mode
- Device description
- Device class & device ID
- Server name
- Virtual address
- Point of Presence address (POP)
- Bytes sent
- Bytes received
- Client registered (date/time)
- Connection established (date/time)
- Last re-authentication
- Total connect time
- Client version and OS
- Client docked/ undocked
- Power source (battery/external)
- Battery (power percent)
- Wireless AP SSID
- Wireless AP BSSID
- Apps using the network (bytes sent/received)

This information is readily accessible either through the Mobility console or programmatically using the server RESTful API. Client session details are updated when the information changes or on a periodic basis. For example, POP address details only changes when the client roams to another network. When the mobile device acquires a new POP address, the Mobility client informs the Mobility server of the new session detail — no other details are sent if they have not changed.

Analytics Module

The Mobility Analytics Module adds reporting and notification capabilities to Mobility deployments. Data is collected from all of the Mobility servers in a pool and forwarded to the database for storage and analysis. The Analytics Module also monitors various system conditions and sends notifications when it encounters them.



The Analytics Module captures intelligence on user, device, and network behavior and usage.

Because connection data is continuously gathered from both the client and server side of the connection, the Analytics Module is able to capture an extraordinary amount of detailed information regarding mobile resource usage. This goes far beyond monitoring connections and logon/logoff events. It includes the applications used, amount of bytes transferred per application and in aggregate, and the name of the wireless interface (and therefore, the network used.) The data collected delivers insight into individual devices, application and VPN usage; bandwidth consumption, connectivity patterns, and battery life. Because much of this information is already available by virtue of the Mobility's unique architecture, Analytics adds very little additional overhead to wireless networks.

The Analytics Module includes a set of pre-defined reports, accessed through the Mobility console, with a straightforward interface for selectively filtering the data. This allows the administrator to group and isolate users, networks, applications, time periods and more. It can also be configured to check incoming data against a set of administrator-defined conditions and issue notifications when a condition is met.

Related Information

Additional information can be found on the NetMotion web site, www.netmotionsoftware.com and in the System Administrator Guide.