

# Intel<sup>®</sup> Technology Journal

Wireless Technologies

**Seamless Connectivity to  
Wireless Local Area Networks**

# Seamless Connectivity to Wireless Local Area Networks

Allan Chin, Intel Communications Group, Intel Corporation  
Ajay Gupta, Intel Communications Group, Intel Corporation  
Ranjit Narjala, Corporate Technology Group, Intel Corporation  
Venkata Vallabhu, Intel Communications Group, Intel Corporation

Index words: Wireless LAN, Adapter Switching, Automatic VPN Launch, Seamless Roaming

## ABSTRACT

The Internet is clearly becoming more diverse in its types of constituent interconnected networks. Wireless Local Area Networks (WLANs) are becoming ubiquitous in today's technology-driven world. Quick and easy connectivity to these networks is a big step in achieving the objectives of continuous mobility, constant reachability, and consistent connectivity.

This paper describes the challenges associated with a client solution and discusses how the Intel® Wireless PROSet software could support quick and seamless connectivity to WLANs, including additional features that enhance the user experience. Some of these challenges and solutions are covered in this paper as follows:

1. Configuration and manageability of the WLAN adapter and the available networks, and Intel's Wireless Configuration feature.
2. Multi-homed systems with their associated challenges, and a proposed solution using the Adapter Switching feature.
3. Security implications of wireless networks, and the proposed automatic Virtual Private Network (VPN) launch feature.
4. Co-existence with other "smart clients" on the client machine.

We also provide a walk-through of a common usage scenario, and depict how this software could enable a mobile user to truly enjoy the advantages of being wireless. Finally, we highlight what lies ahead in the world of constant connectivity and continuous roaming.

## INTRODUCTION

The Internet is clearly becoming more diverse in its constituent interconnected networks. Wireless 2.5G/3G licensed spectrum radio networks, unlicensed spectrum-

based Wireless Local Area Networks (WLANs) and Personal Area Networks (PANs), and wired IP-based networks will soon converge and interconnect to provide a rich medium to drive voice-data convergence to IP. Internet-attached devices are becoming mobile and multi-modal (i.e., capable of connecting to more than one physical network type). Due to higher bandwidth availability and ubiquitous reachability, peer-to-peer communication will become more prevalent.

In this paper, we describe the challenges that are peculiar to WLANs, and go on to illustrate how the Intel Wireless PROSet solution could meet these challenges.

Unlike wired networks, one cannot physically connect a wire to a wireless network. The client needs to listen for signals coming from the access point (AP) in order to establish a connection. By automatically sensing 802.11 signals and intelligently auto-connecting the user to the "best" wireless network, based on a user-defined configuration in the form of profiles, the Wireless Configuration feature essentially eliminates the complexity associated with configuring and managing WLANs on the client.

Many laptop computers in use today contain both wired and wireless network interface cards, thus making them multi-homed. This means that users cannot deterministically select one interface to be used for all traffic when multiple interfaces are available. In this paper, we describe a solution that allows the user the ability to prioritize a list of media types; the software then intelligently and automatically selects, and presents for use, one particular adapter when multiple adapters may be present. We then touch on the possibility of using this platform approach to intelligently conserve battery power.

Intel's "Intelligent Roaming" concept defines two different roaming models:

1. Nomadic Computing: This is a usage model where a user easily connects to the best available network

without expectation of session continuity (for example, a persistent TCP connection or a persistent Virtual Private Network (VPN) session) across networks. Nomadic computing can be further broken down as follows:

**Quick Connect:** In this situation the user's mobile device (i.e., laptop) has only one physical network adapter available for use. As soon as the adapter (otherwise called the Network Interface Card, or NIC) becomes active (i.e., enters a link-up state), a connection to the target network will be automatically initiated if permitted by policy. A VPN session may be activated if necessary as determined by a profile associated with the physical NIC type and network identifier, such as the WLAN Service Set Identifier (SSID).

**Switched Roaming:** This is a refinement of the Quick Connect scenario and is applicable only when the mobile device has two or more physical network adapters that are available for use. In this situation, if connectivity is possible on more than one interface, the software will select and activate the interface that is preferred by the user as determined by a local policy.

2. **Continuous Roaming:** This is a further refinement of the connectivity types listed above, adding session continuity across IP subnets and across physical network adapters. This solution can also provide for maintaining session continuity across VPN sessions. Session continuity is obtained by allowing the mobile client to retain a persistent IP address while moving across different IP subnets, using the Mobile IP standard (RFC 3344).

This paper describes a solution that supports only nomadic computing. The long-term goal is to provide technology building blocks to enable heterogeneous roaming across global wired and wireless IP-based networks while accomplishing the objectives of continuous mobility, constant reachability, and consistent connectivity.

Due to the broadcast nature of the wireless medium, security is required at the link layer. However, Wired Equivalent Privacy, or WEP-based security for wireless LANs has proven to be weak; an accepted solution is to deploy security at the network layer using IPSec. This paper describes how secure layer-3 connections are established as part of connecting to a wireless network.

With WLANs being increasingly deployed in hotspots, there are several smart clients from various clearing-house vendors and hotspot operators. We describe how Wireless PROSet automatically detects other smart clients in the system and co-exists with them.

We provide the reader with a walk-through of a common usage scenario and depict how this software could enable a mobile user to truly enjoy the advantages of being wireless. We end by highlighting what lies ahead in the world of constant connectivity and seamless roaming.

**Note:** The architectural views expressed in this paper are solely that of the authors. The authors make no express claims or comment on either existing or future Intel products, including the PROSet framework, based on ideas conveyed on this paper.

The next four sections describe some of the challenges that are unique to the configuration and use of Wireless Local Area Networks (WLANs), from a client perspective.

## CONFIGURATION AND MANAGEABILITY

### The Problem

Wireless networks are fundamentally about mobility. Users move between a number of locations that offer connectivity to different networks by using varied security models, and sometimes by requiring different TCP/IP settings. To switch between these networks, a user is required to make various network configuration changes. A lot of these changes require an in-depth understanding of network configuration, TCP/IP settings, and security models. Some networks also require additional software, such as Virtual Private Network (VPN) clients, to be run in order to obtain complete connectivity.

*Ad hoc* networks, where clients connect as peers directly to each other, are also becoming more popular. They offer the benefits of being able to easily and quickly exchange data between peers without the need to set up any infrastructure. Setting up the client to act in this mode can be very challenging; switching between ad hoc mode and normal (or infrastructure) mode can also be intimidating.

Configuring and managing these wireless networks is thus a time-consuming, tedious, and error-prone process that only very sophisticated users are able to successfully accomplish. Since one cannot get around the need to make these changes, the need for a tool that makes these changes automatically is highly desirable, and in fact is necessary if a normal user is expected to be able to benefit from wireless connectivity.

### Our Solution

Intel's Wireless PROSet software suite enables the user to configure and manage all supported Intel wireless (802.11a/b) network adapters present in the system.

Users can create “profiles” for the different networks they would like to connect to; each of these profiles contains all the information necessary to connect to a particular network. This suite also provides a wizard for quick and easy configuration of these profiles, and allows the user to prioritize the list of profiles. Another valuable feature in the software suite is the ability to import and auto-import profiles. This allows an IT administrator to create profiles for the company WLAN and then provide it to the employees for import, or push it onto the employee machines for auto-import.

The Intel Wireless Configuration service, a component included in this software suite, will use the information specified in these profiles to automatically and seamlessly connect to the best available network, including ad hoc networks. This saves users from having to determine what network they need to connect to, and it alleviates the need for users to constantly remember all the network configuration parameters and to change the settings every time they move to a different network.

### The Details

Quite unlike wired 802.3 networks, a user who would like to connect to an 802.11 WLAN does not need to physically connect a network cable to use the network. Instead, a number of WLANs may be available and connected to by merely “listening” to beacons that are broadcast by WLAN access points (APs) and/or WLAN client peers operating in ad hoc mode. Connecting to one of these WLANs requires programatically providing the “name” (SSID) of the network that the client desires to be connected to, and also optionally providing the authentication and encryption parameters that may be required for the particular network.

The ability to connect to various WLANs without requiring any user intervention facilitates the concept of automatic and seamless network connection services.

Firstly, the Intel Wireless PROSet software enables the user to create a WLAN profile or set of profiles, each profile consisting of information necessary for automatically connecting to a network, i.e., a set of network identification parameters as well as authentication and encryption parameters. There are essentially two types of profiles, one for infrastructure mode and one for ad hoc mode.

When the wireless adapter is configured in infrastructure mode, it will connect to a wireless AP. In ad hoc mode, however, the wireless adapter will look for and connect to other wireless adapters that are within mutual communication range of each other. An ad hoc network is typically created spontaneously, and its most distinguishing feature is its limited temporal and spatial extent. This mode allows users to quickly and easily

exchange data without requiring any infrastructure (such as APs) to be set up.

Once a set of profiles have been configured, the user may also optionally prioritize this list. The Intel Wireless Configuration service, which is a component of the Wireless PROSet software, will attempt to first connect to a network using settings in the user’s “most preferred” profile. This is accomplished by populating a “scan-list” that comprises the currently available networks, and then running through the profile list to determine which one can be used. The process of mapping the user’s WLAN profiles to available networks uses the information available from the scan-list.

The Adhoc Wizard, another component of the Wireless PROSet software, enables the use of the wireless adapter in ad hoc mode. With the Adhoc Wizard, a user can select, from a list of peers, a particular peer he wishes to communicate with. The Adhoc Wizard will set up communication with that peer. Once the necessary data transfer with that peer is complete, the user can close the Adhoc Wizard, which will result in a connection to an infrastructure network being re-established.

Intel’s Wireless Configuration service goes beyond Microsoft’s ZeroConfig by allowing the user to specify, as part of an infrastructure profile, TCP/IP settings and VPN parameters. The VPN settings specified in the profile allow a VPN tunnel to be automatically established, once network connectivity has been obtained, therefore obviating the need for the user to remember that a VPN tunnel needs to be established before using the wireless network.

Another difference between Intel’s Wireless Configuration service and Microsoft’s ZeroConfig service is the way in which “stealth” networks are handled. A stealth network is one where the network’s primary identifier (its SSID) is not broadcast via 802.11 beacons. While Microsoft’s ZeroConfig service gives a higher priority to non-stealth networks, Intel’s Wireless Configuration service establishes priority solely based on the order of the user’s profiles.

In addition, Intel’s Wireless Configuration service provides these automatic and seamless connectivity features not only on Windows XP and Windows 2000,\* but also on older versions of Microsoft’s operating systems, such as Windows NT\* and Windows 98,\* thus allowing a variety of users the ability to enjoy the benefits of wireless networks.

### Summary

Intel’s Wireless PROSet software suite thus offers the following benefits:

1. It enables the user to create and prioritize a set of profiles for different WLANs. These profiles are used by the Wireless Configuration service to automatically connect to the best available network. Users therefore do not need to constantly configure the wireless adapter each time they move into a different network.
2. This software will work on almost all versions of the Windows operating system in use today, thus providing a variety of users with the ability to easily configure and manage their WLAN connections.
3. The Adhoc Wizard enables seamless connectivity to ad hoc networks. It automatically provides a list of available peers within the ad hoc network specified by the user, which helps users see the other peers they can connect to. It also connects back to an infrastructure network when the ad hoc connection is terminated, thus simplifying the entire connection process.

## ADAPTER SWITCHING

### Problems Associated with Multi-Homed Hosts

The standard for today's laptops is to come with at least two different types of network adapters: a wired adapter and a WLAN adapter. Most enterprises support both wired and WLAN connectivity. Users are also starting to install and use both wired and WLANs at home, along with their broadband Internet connections. As a result, the user may be simultaneously connected using both a wired and WLAN adapter, and is able to communicate using both these adapters.

To understand the problems that arise when the user is connected using more than one network interface, we need to take a closer look at how the Microsoft Windows operating system functions when multiple adapters are present. In general, Windows maintains a route-table for all installed network adapters (also known as interfaces), and this route-table is consulted to determine which interface an outbound packet should be sent out on. Whenever an adapter gets an IP address, Windows inserts a set of entries into this route-table. If more than one adapter is in a link-up state and acquires an IP address, each of these adapters will have a set of entries in the route-table that point to itself. When an application sends a packet out, the packet will first encounter the TCP/IP stack in the kernel. The TCP/IP stack will consult the route-table to determine which interface this packet should be sent out on. If there are multiple choices because of the multiple adapters that are available, TCP/IP will select one particular adapter, based on a set of internal algorithms, and the packet will be sent out

over that interface. Subsequent packets from the same application that belong to the same stream will be sent out over that interface, as long as the interface remains in a link-up state and maintains a valid IP address. The user thus cannot deterministically ensure that all traffic will go out a particular "best" interface when more than one interface is available—the choice of the interface is up to the operating system.

### Our Proposed Solution

Adapter Switching could step into the picture at this point, and allow the user the option to deterministically select one interface to be used for all traffic when multiple interfaces are available. The user can prioritize the adapters present in the system based on the adapter type. The Adapter Switching software will select the best adapter out of those available, depending on the user's preferences and present that to the operating system to use for all network connections.

### The Details

As mentioned in the introduction, Quick Connect is a feature that will enable the user to quickly and easily connect to an available network when the user's mobile device has only one physical adapter available for use. When the mobile device has more than one physical adapter, the Switched Roaming feature would provide the user with the best interface to use.

The proposed Adapter Switching component consists of a Network Device Interface Specification (NDIS) 5.0-compliant intermediate driver, a mobility services client (also called the Adapter Switching service, which contains the Policy Manager), and a plug-in user interface (UI) component for the PROSet GUI Manager. The intermediate driver is responsible for Dynamic Host Configuration Protocol (DHCP) blocking (explained below), and has a mini DHCP client implementation. It also serves as a link monitor and informs the Policy Manager (PM) up in user space about link change events. The plug-in UI component for the PROSet UI allows the user to select a preferred adapter type, and it also displays the current status of the various network adapters present in the system.

The PM enforces a set of user-defined policies. It contains all the logic necessary to determine the most preferred state of the mobile device, given a dynamically changing network environment and a set of user-defined policies. The user will be able to set a preference level for the different network adapter types that are available. The PM will attempt to select an adapter that has the highest preference, as determined by the policy, and make that the primary adapter for all network connections on that mobile device. In the event that the preferred adapter

type is unavailable, the PM will attempt to select another one from the list that it maintains.

The Adapter Switching component will be enabled only when all supported adapters are DHCP-enabled—if any adapter is assigned a static IP address, the Adapter Switching component will disable itself and appropriately notify the user. Normally, all adapters in a connected state on the mobile device will be able to obtain an IP address; the route-table on the Windows machine will contain routes using all the available adapters on that machine. If there is more than one adapter in the system that is available, and if the route-table has the same metric for all routes, then the particular adapter that will be chosen for data transfer will be indeterminate (an adapter for data transfer is chosen based on a combination of longest prefix match and route-table metrics). The Adapter Switching component's job is to deterministically present a single best interface for use, given that multiple interfaces are available.

This is achieved by allowing the TCP/IP stack to obtain an IP address for only one interface at any given time—and this will be the “active” interface. This means that the route-table will contain entries for only one interface, and there will no longer be any ambiguity when it comes to choosing an interface for data communication. The intermediate driver thus blocks DHCP traffic on all but the active interface, and prevents the TCP/IP stack from obtaining an address on an “inactive” interface. The interface to be made active is chosen by the PM, depending on network conditions and the user's policy, and this decision is communicated to the intermediate driver when state transitions occur.

The proposed Adapter Switching component will also support *ad hoc* 802.11 connections (in either static or dynamic addressing mode) concurrently with other adapters in infrastructure mode. This means that if a WLAN adapter is set to be in *ad hoc* mode, it can be used for communication at the same time another adapter (in non-*ad hoc* mode) is used, without leading to a loss in the deterministic behavior that Adapter Switching provides.

### Power Conservation

Another very useful feature of the Adapter Switching software is its ability to help save battery life by working in conjunction with Wireless PROSet.

Consider a scenario where a user Joe has prioritized wired adapters over WLAN adapters. When Joe is working at his desk, he is connected using his wired adapter; since WLAN deployment in his company is ubiquitous, he is also connected to the WLAN. When the Adapter Switching software is enabled, it detects the availability of both types of connections; however, since Joe has indicated his preference for the wired adapter, the

Adapter Switching software will only allow his wired adapter to obtain an IP address and be used for communication. Since the WLAN card is essentially not used at this point, the Adapter Switching software informs Wireless PROSet that this adapter can be turned off. When Wireless PROSet receives this indication, it can potentially turn off the radio on the WLAN card and therefore save precious battery life.

When Joe unplugs his laptop and walks to a conference room, the Adapter Switching software will detect this change; since the preferred (wired) connection is no longer available, the software will request Wireless PROSet to activate the WLAN card. Once that happens, Adapter Switching will ensure that this adapter can acquire an IP address; traffic will now start flowing over this adapter.

This intelligent interaction between Adapter Switching and Wireless PROSet can thus help save power.

### Summary

The Adapter Switching software thus offers the following benefits:

1. It offers a single best interface to use, depending on the user's preferences and the current state of the network. All traffic will be deterministically directed over this one interface, even when other less-preferred interfaces are available.
2. It offers one way to help conserve valuable battery power by intelligently turning off the WLAN adapter when it is not needed.
3. It increases the security of a system when a VPN is enabled. If a VPN client is enabled when multiple adapters are available, the VPN tunnel will be established over one particular interface, and all packets flowing through that particular interface will be encrypted. However, there is a possibility for unencrypted packets (which might contain sensitive information) to flow over another interface that is not protected by the VPN tunnel. The Adapter Switching software prohibits this from happening by ensuring that there is only one interface that is active at any given time.
4. It supports “split-tunneling” within an interface when a VPN tunnel is active, if this feature is also supported by the VPN client. This will enable access to local resources (such as a local printer) while in a VPN session.
5. It supports communication over a WLAN adapter in *ad hoc* mode while simultaneously supporting communication over the preferred interface (which is in non-*ad hoc* mode).

## AUTOMATIC VPN INVOCATION

### The Problem

WLAN deployment is quickly gaining traction, especially in enterprise environments, where employees expect to be able to move between buildings and conference rooms and be constantly connected to the network. The first question that will be asked by an IT administrator before she starts to deploy a WLAN in her enterprise will be about its security features. It's a well-known fact that the best currently available mechanism, Wired Equivalent Privacy (WEP), is broken. Other more secure mechanisms (such as 802.11i) are not yet standardized, and interim solutions (such as Wi-Fi Protected Access (WPA)) are not yet widely deployed. As a result, most enterprises require their users to protect all WLAN traffic with a VPN connection. This ensures that a person with malicious intent sitting in the parking lot of the company campus cannot gain access to the enterprise Intranet via the WLAN. It also protects (by encrypting) all user traffic that flows over the air, thus ensuring that the same person in the parking lot cannot just snoop the air for tasty tidbits of information.

While this is great from a security perspective, it is a virtual nightmare from a user's standpoint. When a user wants to use his wireless connection, for example at a conference room at work, he first has to figure out if his wireless adapter is connected to the network, and then remember to launch his VPN client and connect back into the enterprise before he can be "connected." When he then walks back to his desk and plugs in his wired card (because he likes a higher speed connection when one is available), he needs to remember that his VPN is currently running over the wireless card and that the wired card will be inaccessible to applications until the VPN client is turned off. And then when he needs to run to a meeting a few minutes later, he has to remember to launch the VPN client yet again to connect using the wireless network, and the whole process starts again.

### Our Proposed Solution

In order to spare the user from this tedious and error-prone process, we propose an auto-launch feature that enables a specific VPN client to be launched at the right time; and not only that, the VPN client will also be torn down at an appropriate time.

The user will be allowed to specify all the information necessary to configure and enable the auto-launch feature; this information will be tied to a WLAN profile, as described in the following section.

The auto-launch feature can be supported both when Adapter Switching is enabled and when it is disabled. When a wireless profile with VPN configured is applied

and the wireless card is currently "active," Wireless PROSet will automatically launch the specified VPN client (with a particular VPN tunnel, if configured). Subsequently, when a higher-preferred adapter becomes available, or the WLAN adapter loses connectivity, the VPN tunnel will be proactively torn down.

This feature thus enables the user to enjoy the freedom that a wireless connection provides, without having to be bogged down with all the details required to set up and configure that connection.

### VPN Invocation Using Profiles

With this proposed solution, Wireless PROSet can support the concept of automated VPN connectivity via profiles. While setting up a WLAN profile, the user will have the option of selecting a VPN client (and a VPN profile, if supported by the VPN client) to launch when the profile is applied. For example, the user can create a profile called "Office," and select a VPN client to be launched when the profile is applied. Each time this profile is subsequently applied, the software will automatically launch the VPN client and connect to the VPN gateway using a VPN profile, if one was specified. The software therefore behaves in a proactive manner, and attempts to do as much for the user as possible, in terms of enabling the user to quickly and seamlessly connect to a preferred network.

The Adapter Switching component can automatically detect the presence of supported VPN clients, using a combination of two mechanisms: searching for the VPN clients and having the clients register themselves with our component using "VPN adapters" (described below). Once the Adapter Switching component detects a client(s), it will present that to the user via the Profile Wizard when the user creates a profile. The Profile Wizard then stores this information as part of its WLAN profiles; as soon as a profile is applied, the Wireless Configuration service will query its database to determine if the profile applied contained VPN-related information. If it detects that a VPN client needs to be launched, it will pass all necessary information to the Adapter Switching component that will then attempt to actually launch the client.

### An Extensible Solution

Due to the wide variety of VPN client and gateway implementations, the architecture cannot be designed to work with all VPN solutions; this proposal for Quick Connect and Switched Roaming has been investigated with a select set of third-party VPN implementations. However, we propose an extensible mechanism that will enable other VPN implementations, if they meet certain requirements, to work with this architecture.

VPN clients from different vendors have their own implementations and expose different interfaces. VPN clients also have their own unique profile formats that are used to store all tunnel setup and configuration information. Furthermore, legacy versions of certain VPN clients allow only the client to be launched, while more recent versions allow the client to be launched along with a specific tunnel. In addition, these clients may also allow querying of their internal state to determine the status of a tunnel in place, and so on. In order to provide a common framework across these varied implementations, we have designed what is termed a “VPN adapter.” This adapter implements a uniform interface that we have defined, and it essentially acts as an abstraction layer between our software and the actual VPN client. Using this adapter, we will be able to instruct the VPN client as needed, without having to worry about proprietary interfaces for particular VPN clients. We have designed built-in adapters for a select set of VPN clients; if a new client wants to co-exist with our software, and assuming it has passed the other requirements for co-existence (such as co-existing with our intermediate driver), all that it needs to do is implement this VPN adapter and register itself on the mobile device. Our software will then dynamically detect its presence and be able to utilize that VPN client.

## CO-EXISTENCE WITH “SMART CLIENTS”

With the proliferation of WLANs come hotspots. These hotspots are public places where access to the Internet is provided through WLANs. Several wireless ISP’s (WISPs) provide internet access at these hotspots. In order to gain access to these hotspot networks, the user first needs to be authenticated; the particular authentication mechanisms employed are sometimes proprietary methods. As a consequence, hotspot operators usually require users to install and use their own specialized client software for Authentication, Authorization, and Accounting (AAA). These specialized clients also help the user to look for and connect to the wireless networks at the hotspots.

A user with subscriptions to several WISPs could therefore potentially have a number of “smart clients” installed and active on her laptop. All of these clients, if active simultaneously, will be trying to search for and connect to networks that they each know about. This can lead to software co-existence issues and in-deterministic behavior, eventually resulting in a good amount of confusion on the user’s part. One possible solution is to have Intel’s Wireless PROSet software communicate with these various smart clients using a pre-defined mechanism. This solution is obviously not very extensible, and it would not be interoperable with clients

that are already proliferating the market. Furthermore, this solution will work only with clients from vendors or operators that have chosen to embrace and implement this pre-defined mechanism. To circumvent these problems, Wireless PROSet employs a generic mechanism that is independent of smart clients from other vendors.

In the Windows\* Operating System, applications communicate with the WLAN driver using either a standard OS-defined interface or some proprietary interface. While the Wireless PROSet software can use a combination of both standard and proprietary mechanisms, all hotspot and other AAA smart clients can only communicate with the driver using the OS-specified standard interface.

As mentioned above, Wireless PROSet communicates with the driver using a proprietary interface. When the OS boots up and Wireless PROSet “registers” with the driver, the driver monitors for a subset of “set-able” OS-defined wireless commands as part of the standard interface exposed by the driver. When a AAA smart client is active, the driver receives these standard commands from the smart client and deduces that these commands are from a smart client, and not from Wireless PROSet. It then notifies Wireless PROSet which in turn notifies the user that another smart client is also active. Depending on the action taken by the user, certain network connection-related features will get disabled in Wireless PROSet if the AAA client continues to remain active. At the end of this operation, only one client—either the Intel Wireless PROSet software or the smart client—will be guaranteed to be active and in control. Thus, this solution does not involve any direct client-client communication, but relies on information deduced from the client platform.

## OVERALL SYSTEM ARCHITECTURE

[Figure 1](#) below provides a conceptual overview of the system architecture as described in this paper.

---

\*Other brands and names are the property of their respective owners.

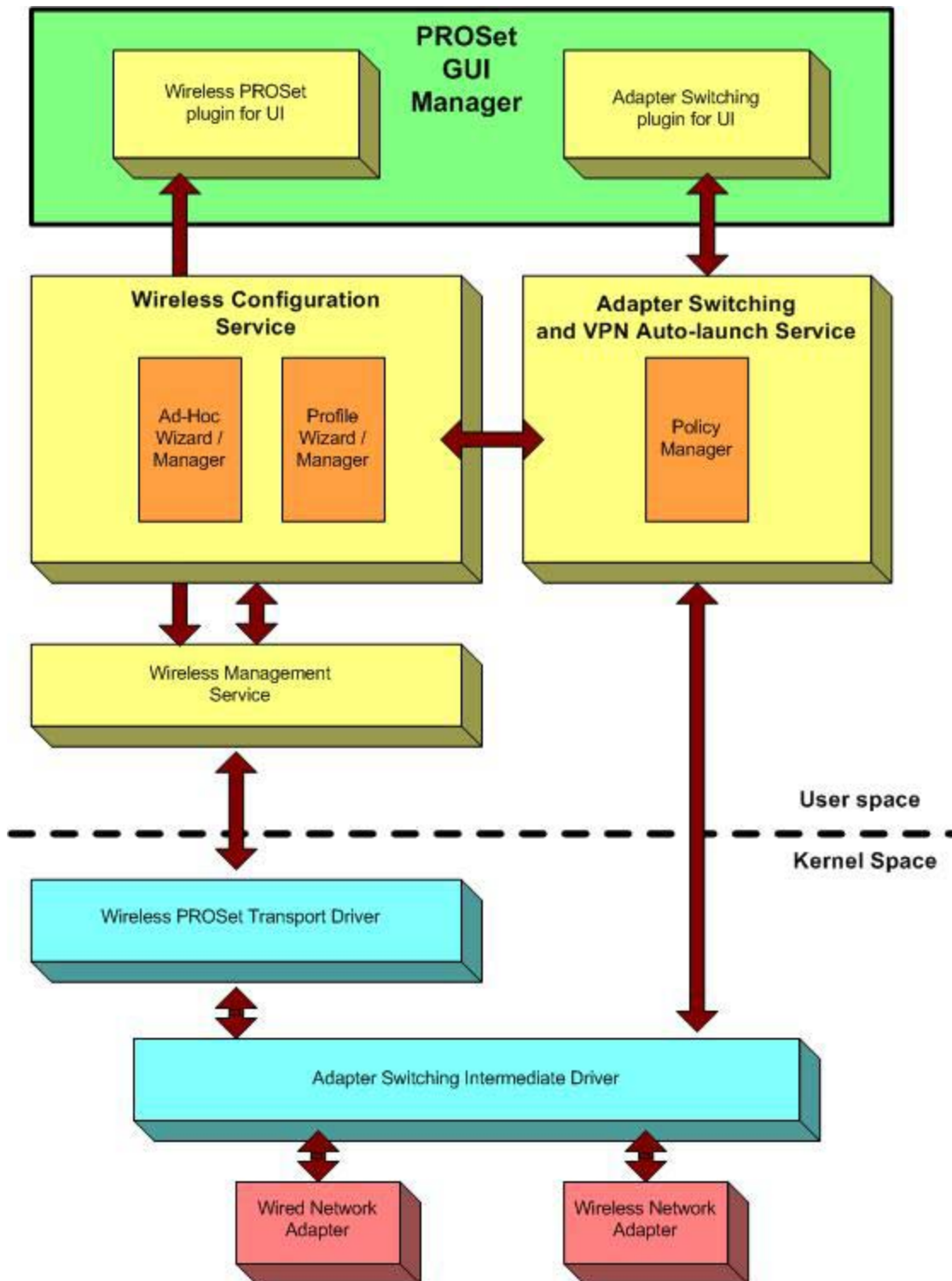


Figure 1: Overall system architecture

## WALK-THROUGH OF A COMMON USAGE SCENARIO

Let's consider a common scenario, and see how all these components fit together. To make this a little more descriptive, let us picture a mobile user, Jane, who has a laptop with two network adapters. One network adapter is wired, and the other one is a built-in dual band wireless adapter, that can function as either an 802.11a or 802.11b adapter. The Intel PROSet software with the proposed features described in this paper is also running on Jane's laptop, to manage her network adapters.

Jane has used PROSet to configure two wireless profiles for her laptop—one profile (ProfileA) is for the 802.11a access points (APs) at her office, and the other (ProfileB) for the 802.11b APs (profile descriptions are illustrative only). Both these profiles have a Virtual Private Network (VPN) auto-launch setting associated with them, as the APs are deployed outside the company's Intranet. Using the Adapter Switching plug-in, Jane has given the wired adapter a higher preference than the wireless adapter.

When Jane walks in to work early Monday morning and powers up her laptop, the PROSet mechanisms are put into motion.

1. The Adapter Switching service detects both the wired and wireless adapters. It first verifies that all adapters are supported by the software. Once it detects that the wired adapter is plugged into an available link, it attempts to obtain a Dynamic Host Configuration Protocol (DHCP) address for the adapter. Simultaneously, the Wireless PROSet component also tries to associate with an available AP.
2. Once the Adapter Switching service verifies that the wired adapter can obtain a DHCP address, and since the wired adapter is the preferred adapter, it makes that the "active" adapter. What this means is that it will ask the operating system (OS) to renew the IP address on the active adapter, and release the IP address on all other adapters. The Adapter Switching component also notifies the Wireless PROSet component of this decision.
3. In the meantime, Wireless PROSet associated the wireless adapter with an AP matching ProfileA. The Adapter Switching service was able to internally verify that the adapter was able to acquire a DHCP address; however, since the wired adapter had a higher preference, it was the one that was made active.

At this point, Jane will be using her wired adapter for all network connections. After a while, she leaves her desk to attend a meeting in a conference room. She unplugs

the wired adapter and takes her laptop to the conference room.

1. The Adapter Switching service detects that the wired adapter was unplugged, and it updates its state to register that there is currently no active adapter.
2. It then remembers that the wireless adapter is available for use. It makes the wireless adapter active and notifies the Wireless PROSet component of this.
3. The Wireless PROSet component then looks up its profile table and detects that there is a VPN client associated with this profile. It instructs the Adapter Switching service to establish a VPN tunnel and passes it the necessary information.
4. As soon as the Adapter Switching service successfully obtains an IP address for the wireless adapter, it attempts to create a VPN tunnel based on information passed by Wireless PROSet in Step 3. It uses the VPN auto-launch component to invoke the VPN client with pre-configured parameters. Jane is required to enter her credentials as part of the VPN connection setup process.

Jane will now be using her wireless adapter, associated using ProfileA, for her network connections. Once the meeting is over, Jane plans on going down to the cafeteria to continue discussions with a colleague. As she takes the elevator down and walks into the cafeteria, she walks out of range of the first AP and into range of a different one.

1. The Profile Switching service within the Wireless PROSet component detects the change and uses the scan-list to determine the APs that are currently in range. It then applies the best profile it can, which happens to be ProfileB.
2. The Adapter Switching service detects the changes in the link state of the wireless adapter. As soon as the adapter disconnects from the first AP, it notifies the Wireless PROSet component, which in turn asks the VPN Manager component within the Adapter Switching service to shut down the VPN tunnel.
3. Once the wireless adapter associates with the new AP, the Adapter Switching service detects a new link, and once again it attempts to make this adapter active. Once that process is completed, it notifies Wireless PROSet. This, in turn, causes another VPN tunnel to be initiated, similar to the process described above.

Jane can now use her wireless link once again without having to determine whether she needs a VPN tunnel, or whether she has to manually initiate the VPN connection process. Most of the link state changes and adapter

switches were handled transparently, and she can continue her work without having to manually connect to networks or subsequently remember to turn VPN clients on and off.

## THE CHALLENGES AHEAD

In the last two years the markets for wireless mobile data have gained momentum and are moving from early adopters to mainstream users. Wireless Local Area Networks (WLANs), General Packet Radio Service (GPRS), and Bluetooth\* technology, respectively, are the most common mobile data technologies, with each technology addressing complementary mobile data connectivity needs. Generally speaking, WLANs provide access connectivity within buildings or hotspots; GPRS data cards or GPRS phones enabled with Bluetooth technology offer wireless connectivity everywhere else.

As the adoption of these technologies increases so does the need for these technologies to seamlessly work together. The long-term goal is for the end-user to be oblivious of the data connection type, and still have access to the best network, without having to switch connections, restart applications, or reboot mobile computers. The software solution described in this paper is a first step in addressing this need by providing a mobile client that communicates with industry-standard infrastructure solutions.

Accomplishing the above would provide a mobile user with a truly seamless roaming experience using industry standards.

## ACKNOWLEDGMENTS

The authors express their gratitude and appreciation to the entire management, development, and software quality teams that worked on the Wireless PROSet and Adapter Switching product. We thank Prakash Iyer and Marc Meylemans for reviewing this paper and enhancing its content. Finally, we thank Marian Lacey for doing a great job editing this paper.

## AUTHORS' BIOGRAPHIES

**Allan Chin** is a staff software engineer in the Wireless Product Development division at Intel Corporation, and is currently based in San Diego, California. Allan has over 20 years of software engineering experience in a wide variety of fields spanning real-time, embedded, navigation systems to user-based Windows applications. Allan received his Masters degree in Computer Science

from Stevens Institute of Technology in 1984. He holds a Bachelors degree in Electrical Engineering from the University of Delaware. His e-mail is [allan.chin@intel.com](mailto:allan.chin@intel.com).

**Ajay Gupta** is a staff software engineer in Intel's Wireless Product Development (WPD) division at San Diego, California. His primary interests include network protocols, networked multimedia, and image processing. Ajay joined Intel in 1996. Prior to joining WPD, he worked on video conferencing and network load balancing. He has a Masters degree in Computer Science. His e-mail is [ajay.g.gupta@intel.com](mailto:ajay.g.gupta@intel.com).

**Ranjit Narjala** is a network software engineer in Intel's Corporate Technology Group at Portland, Oregon. His professional interests include wireless technologies, networking, and mobility protocols. Ranjit received his Masters degree in Information Networking from Carnegie Mellon University in 2000. He also holds a Bachelors degree in Computer Science and Engineering.. His e-mail is [ranjit.s.narjala@intel.com](mailto:ranjit.s.narjala@intel.com).

**Venkata Vallabhu** is a Senior Software Engineer in Intel's Wireless Product Development division at San Diego, California. He is responsible for the design and development of the components in wireless PROSet software at Intel. He has over six years of industry experience in software development in various Wireless Technology and Systems Engineering positions. Venkat holds a B.Tech degree in Electronics and Communications Engineering from Jawaharlal Nehru Technological University (India). His e-mail is [venkata.r.vallabhu@intel.com](mailto:venkata.r.vallabhu@intel.com).

Copyright © Intel Corporation 2003. This publication was downloaded from <http://developer.intel.com/>.

Legal notices at <http://www.intel.com/sites/corporate/tradmarx.htm>.

---

\* Bluetooth is a trademark owned by its proprietor and used by Intel under license.

For further information visit:

[developer.intel.com/technology/itj/index.htm](http://developer.intel.com/technology/itj/index.htm)