Always On, Always Connected Mobile Computing

Michael M. Tso
Daniel J. Gillespie
David A. Romrell
{Mike_Tso, Dan_Gillespie, Dave_Romrell}@ccm.jf.intel.com
Mobile Communications Operation
Mobile and Handheld Products Group
Intel Corporation

ABSTRACT**:** We discuss an Always On, Always Connected (AOAC) usage model for notebook PC's, and how it is enabled by the availability of ubiquitously deployed wireless data infrastructures. An experimental wide area information filtering and delivery system was used to identify missing technologies required by AOAC PC's. We present three findings: the critical role of narrowband data messaging (such as paging) in making data pushing applications financially viable; modifications to the Windows* communications infrastructure to integrate wireless data pipes with existing PC API's; and a new transport protocol which adds advanced data capabilities to narrowband wireless messaging networks, as well as providing a familiar sockets interface that hides applications from the differences in networks (e.g. 2 way paging and GSM's SMS appear the same to the application). Some other interesting aspects of the experimental system are also described, including the use of cell level location information for information filtering.

Introduction

For past few years, the Intel Architecture Labs has been exploring the space of an Always On, Always Connected usage model for the PC. In short, this vision is "the data I want finds me." This paper presents some of the findings in the mobile computing and communications area from Intel's Mobile Communications Operation.

The notebook PC has won in the market place as the platform of choice for mobile computing, largely because the notebook PC has had equivalent functionality and performance as the desktop PC. Of all other factors, communication pipes have the biggest effect on the mobile usage model.

Wide area wireless data is the most strategically important new pipe becoming available. They bring anytime, anywhere communications to the notebook PC, potentially creating new classes of uses and users. We believe the opportunities here are between huge and enormous.

One key application class for AOAC those which "push" data to the user, i.e. your data finds you, unsolicited. Our prototyping and experimentation with different business models in this area shows evidence that a widely deployed, inexpensive wireless network, preferably providing store and forward capabilities, is critical to any cost effective implementation of "pushed" data. We believe this medium must be used in conjunction with broadband (and perhaps more cost sensitive) media such as circuit switched cellular. We propose an architectural approach which uses the cost insensitive medium to deliver short notifications for pending "pushable" data, and let the user choose if, when and how to connect to retrieve the data. Narrowband data messaging (e.g. 2 way paging, Short Messaging Service in digital cellular) appears to be the most promising for meeting the notification need.

Enabling AOAC requires us to first wireless communications capabilities into the PC communication software infrastructure, so that applications can access the new communication pipe. The approach we chose was to map existing standard APIs to new wireless pipes, while paying equal attention to both broadband and narrowband wireless pipes based on our finding discussed above. We also defined a new data transport protocol for narrowband data messaging. These high level API's allow applications to be transport independent, i.e. uniform APIs allow the same application to work over GSM, CDMA, or CDPD. And by using existing APIs, we reduce the application programmers' learning curve.

We prototyped an experimental wide area information delivery system (nicknamed "InfoCast") to test new application ideas made possible once wireless capabilities are easily available to the application programmer. In the paper, we will briefly discuss a few aspects such as using the user's cellular location information (in addition to traditional means) to filter information, and using a hybrid communications scheme to increased perceived network bandwidth.

The AOAC Vision

Our research is motivated by our desire to find exciting new uses for the notebook PC. The notebook PC is the undisputed mobile computing device based on popularity. At over 10 million new units per year and a CAGR of around 30%, we believe the notebook PC will continue to be the device of choice for mobile computing and data communications.

Besides mobility, the main differences between a notebook and desktop PC are the power source and communications. Communication has the most profound effect on usage. We believe discontinuities in mobile usage are fundamentally driven by the availability of new communication pipes.

Wireless is a new data communications pipe that hasn't yet been fully utilized by applications. Wireless is interesting because it is truly anytime, anywhere connectivity. We believe that ubiquitously deployed, affordable wireless data communications will catapult the notebook PC into an even more exciting usage model: always on, always connected

(AOAC). The advantages of AOAC over traditional wired communications are:

1. the user can communicate anytime, anywhere, without looking for plugs
2. the user can be reached anytime, anywhere.

While (1) above is basically an ease of use issue, (2) enables a new class of applications which "push" data to the user. In a nutshell, "pushing" is the vision that "the data I want finds me."

"Pushing" can enhance many existing communications applications which use "pulling" (or user initiated). For example, instead of dialing in and waiting for email to download, your LAN or Internet email can be automatically forwarded to your notebook PC in the background, so they are always available instantly when you view. Another example, calendar (or pricing database) changes can be automatically synchronized with the scheduling database in your office so both you and your colleagues in the office always have up to date information. Automatic updates to cached web pages is yet another example.

"Pushing" also enables new classes of applications. It is particularly compelling for applications involving asynchronous event notification, such as news, traffic, and weather information. Urgent notification, either generated by a person or triggered by events such as stock dropping below a certain price, is another application class. Being always reachable also means other people can reach you, e.g. a colleague video-phoning your notebook PC or your system administrator downloading the latest application upgrade at their convenience. Your notebook PC can also act as an answering machine for voice, data, and fax calls.

Wide area wireless networks which support roaming enable a new class of location sensitive applications. Cell level location information is available in the network and can be exposed to applications to make interesting uses, such as sending the user local restaurant or entertainment information in the evening or the latest traffic report for the user's vicinity.

Adding Wireless Support to the Windows* 95 Communication Infrastructure

Overview

The success of a new usage model relies almost entirely on an abundant supply of new, useful applications. The key challenge posed by AOAC is in providing application developers with tools and API's for the new wireless pipes. Wireless media such as cellular and paging require support for unique features such as call control, device power management, signal strength, and device statistics monitors. The most popular notebook operating system, Windows 95, had inadequate support for wireless communications.

But which Wireless Mediums are critical?

The abundance of incompatible wireless network technologies required us to prioritize which wireless technologies to integrate first. Our first criteria is for seamless roaming and high availability, the wireless network must be widely deployed. This made us pay immediate attention to digital cellular networks such as GSM, whose coverage and subscriber base has been growing at a phenomenal rate of 40% CAGR according to Dataquest. The second wireless medium that is critical for data pushing, narrowband messaging (e.g. SMS or 2 way paging), wasn't obvious at first.

Billing is an important consideration for implementing pushed data. Traditional network protocols (e.g. TCP/IP) and applications (e.g. ftp) assume connection time is essentially free. AOAC introduces two new factors that dramatically influence applications and usage models:

1. wireless media, at least for bulk data transfer, will not be free due to increased demands on the limited spectrum available
2. the user may or may not find the unsolicited ("pushed") data useful, but the wireless network resource to deliver that data has already been used.

A naïve approach to pushed data is for the server to establish a connection to the notebook application and transparently push data over this link. This produces the desired result, in that the notebook appears always connected, and data is pushed without requiring the user to connect and request it. However, the user is billed for server initiated network utilization, and after arrival the user may find the content was not worth the cost.

We believe one practical implementation of data pushing is to always push short "notifications for pending data" rather than the data itself. As in the above case, the server has data to deliver to the client. However, instead of a server initiated bulk transfer connection, the server sends an inexpensive, small wireless message which notifies the user of available content. This message contains simple data that include a "teaser" to describe the data, and properties that indicate usefulness (e.g. size, source, target application, download time, etc.). Thus, user determines billing by choosing what content is interesting, when to download it, and over what medium.

For example, a notification may indicate an urgent email message available on a server. The mobile enabled email application can receive the page notification. Based on this page which may include the email subject, author, urgency, and size, the application can estimate cost and download time over different mediums and the user can select how to download the content.

A second problem with always connected mobile PC's, is availability. In many cases, the mobile PC may be unavailable because of RF interference, out of coverage, or powered down to conserve power. Paging uses store and forward messaging that enables the notebook to be always reachable. In the above example, the urgent email notification will be stored in the paging network and delivered as the user becomes available.

It is our thesis that in addition to broadband media, a widely deployed, inexpensive wireless medium is required as the notification channel to implement data pushing economically. This medium need not be synchronous, nor high bandwidth, but it is desirable to be reliable (i.e. at least include acknowledgment) and provide some store and forward capabilities. We believe a class of narrowband messaging solutions (e.g. Short Messaging Service (SMS) in digital cellular networks, and US NB-PCS) meet these requirements. Thus enabling narrowband messaging on the PC platform is a key element of our AOAC program.

Thus our first technology development task is to add digital cellular and narrowband messaging support to Windows.

The Case for Using Standard API's

Defining a set of new communication API's for wireless data is straightforward. But this approach will not attract many developers because the increased time to market associated with a learning curve. Thus we chose to painstakingly retrofit the existing Windows 95 communications infrastructure with wireless support. This approach also has the advantage of allowing many existing applications to use wireless modems and network cards without modification. For example, e-mail can be sent over SMS (GSM's 2 way short messaging service) from off the shelf MS Exchange without any modifications. After adding SMS support to MAPI, we were able to develop a distributed chess game using SMS as the transport in two days. The same application would have taken many weeks if we couldn't use the tools available for writing MAPI applications.

The Windows 95 communication software infrastructure, as illustrated in Figure 1, consists of three API's: the Messaging API[9], the Telephony API [16] and Winsock 2 [17]. MAPI manages all messaging data types, such as email and fax. TAPI is used by applications to establish calls, perform call control, and get and set device attributes such as signal strength or power management level. Winsock 2 is an interface for packet data communications based on many QoS enhancements to Winsock 1.1 which was closely modeled after BSD sockets.

We have found that it is possible to seamless integrate wireless into these interfaces with minimal changes to the API's. The following changes are needed to the existing communications infrastructure.

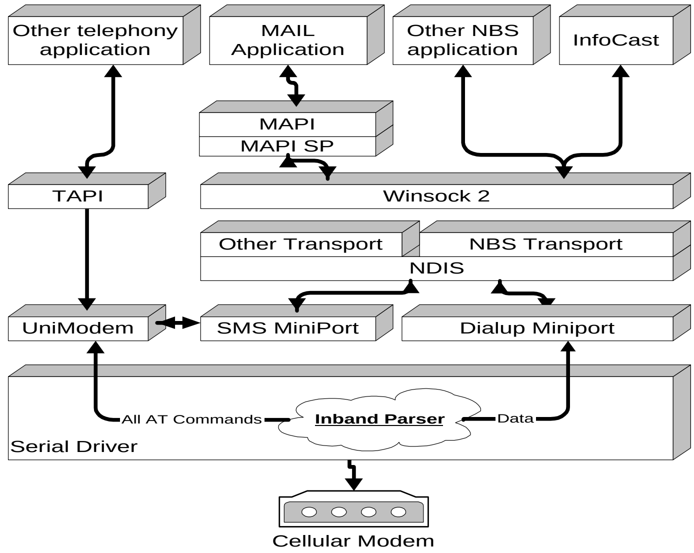

Figure 1: Windows Communications Architecture

1) Support for Cellular Specific AT Commands in the Windows 95 Unimodem driver

This support is only needed when the wireless device uses extended AT commands specific to the device. There are many such standards available, such as IS-99 for CDMA, IS-135 for TDMA, and GSM 7.05[7] and 7.07[8]. These commands offer applications or OS services (such as protocol stacks) added and sometimes necessary control to the wireless specific aspects of the device. It is desirable to create a single set of operations for managing wireless devices in general that are not network specific as the AT command sets are.

Unimodem[14] was created to solve this same problem of the proliferation of many different AT command extensions for POTS modems. Unimodem uses a registry which dynamically maps modem specific AT commands to high level TAPI calls so that applications only deal with individual functions such as "call forwarding" instead of AT strings. Working with a group of leading companies in the PC and Telecommunications industry, we have defined a specification for extending Unimodem to support a generic family of new wireless capabilities. Our solution adds three sets of features to Unimodem: advanced call control is handled through TAPI, advanced device management and configuration is handled by Control Panel applets using the Unimodem Registry, and support for narrowband messaging (e.g. SMS) is accomplished by cooperation with NDIS, Winsock2, and MAPI.

2) Support for ITU V.80[15]

In our architecture, V.80 support is required for wireless devices which send and receive in-band messages. Many devices emulate AT modems, and because modems are interfaced via a purely serial interface, any asynchronous messages to or from the device during the transmission of data must be embedded in the data (i.e. in-band) with escape characters. In-band messages can be control or data messages, e.g. GSM terminals (and other digital cellular devices) are capable of responding with signal strength notifications or transmitting an SMS during a data call. V.80 defines the format for in-band messages (originally designed for DSVD modems).

In order to support devices which support in-band SMS, the Windows serial driver (Serial.sys) must be modified to include an in-band parser. When a V.80 data segment is received from the wireless device, it must be inspected by an in-band parser to determine the payload type. If the sampled data is not modem AT control responses, the message is forwarded to the Winsock 2 stack through normal path for dial-up networking. If the block contains regular AT commands, it is sent to Unimodem for processing. When the Unimodem driver receives the AT Command, it must determine if the command is a normal AT command or an SMS AT command. Following this all normal modem AT commands and responses are processed inside the Unimodem driver while SMS messages are handed to the SMS miniport. Once SMS Messages are handed to an SMS NDIS miniport, they are packaged and handed up to a Winsock2 NBS Stack described below. Thus, all SMS messages will be funneled to the Winsock interface.

3) Narrowband Miniport

To use Winsock, narrowband messages coming from Unimodem or the wireless network device itself must be delivered through the Network Device Interface Specification (NDIS) interface. A special module called a narrowband miniport must be supplied so that the data is properly formatted for handoff to Winsock and the NarrowBand Sockets transport. It should be noted that the above diagram shows the configuration for an SMS miniport. However, any wireless network device's miniport could be used in conjunction with section 5 of the NDIS Specification.

4) MAPI Service Provider for Narrowband Messages

This module will allow narrowband messages (such as SMS or 2 way paging) to be interpreted as e-mail on the client PC. The required implementation is a MAPI Transport Service Provider that listens on a particular NBS port. Services in the NBS layer will allow the mail size to exceed the size of the page itself. Thus, users can communicate using Internet mail or the cellular phone paging network. Of course the email may itself be treated as data for mail enabled applications, such as the SMS chess we described earlier.

Narrowband Socket Protocol

Narrowband Socket provides a familiar sockets API and transport layer services for narrowband messaging networks. We expect narrowband messaging to be used to send small notification messages to mobile users, and the notification messages may themselves contain useful information such as news or email headlines. The value in this medium is that it provides a way to always reach the user at very moderate costs.

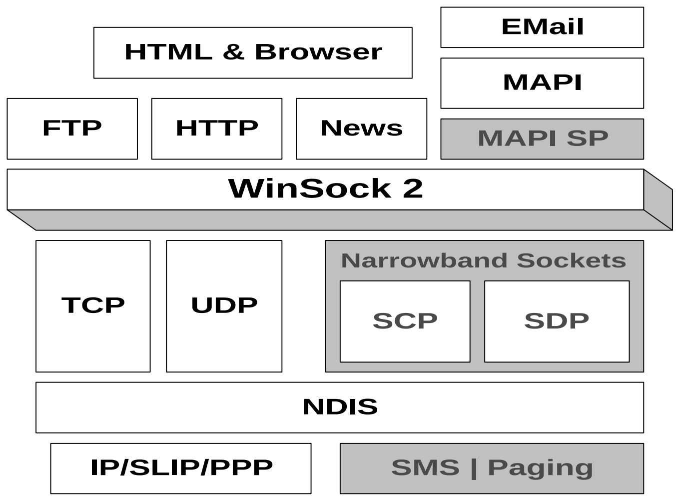

Figure 2: Narrowband Socket System Architecture

However, existing narrowband services such as paging or SMS present many challenges to developers. They have very small packet sizes, limited or no support for reliable delivery, no application level addressing (e.g. no way of distinguishing if a message should be routed to MS Exchange or our distributed chess game), and proprietary interfaces to the hardware. The result is that application development is complex, applications must be adapted to work on different networks, and no application level sharing of the wireless channel.

To resolve these issues, we defined a new transport protocol under the Winsock2 API called Narrowband Sockets (NBS). Figure 2 illustrates the architecture. NBS provides a network independent API for wireless messaging. It provides a simple connection oriented protocol (SCP) and a simple connectionless protocol (SDP) that are analogous to TCP and UDP. In addition to traditional transport level services such as fragmentation and reassembly, reliability, sequencing and flow control, SCP and SDP also provide application level addressing similar to TCP and UDP ports. The end result is that developers can easily create applications using an already familiar API and standard PC tools, and their applications will run over any wireless messaging network, as well as running concurrently with other apps over the same channel.

InfoCast: An AOAC Research Vehicle

Having architected the integration of critical wireless pipes to the PC, we decided it was time to experiment with the target rich applications area for AOAC. "InfoCast" is an internal nickname for a wide area information filtering and delivery system. It was developed as a research vehicle to test many of our ideas.

InfoCast clients listen passively over a communication channel for news and other data being "pushed" by the InfoCast Servers. Thus news, information, advertising and weather are trickled into the client in the background, so that the user will always have instant access to the most up to date information. Below is a high level system overview of InfoCast

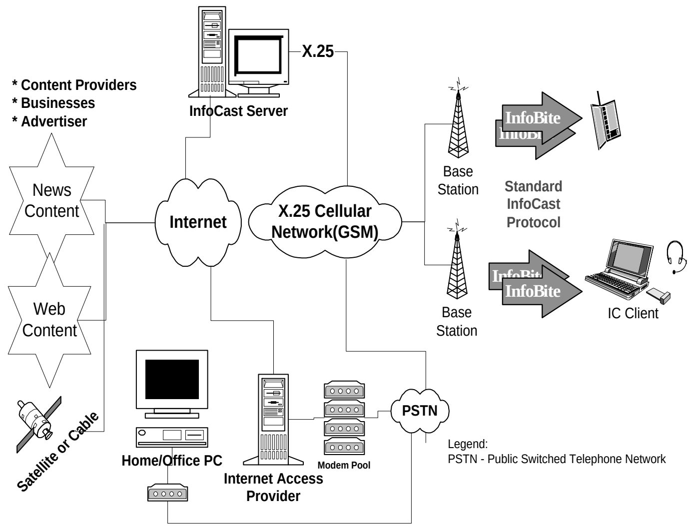

Figure 3: InfoCast System Architecture

Figure 3 illustrates the InfoCast architecture. The system consists of content sources, InfoCast servers, and InfoCast clients. Content sources can be news wire services, cable TV stations, USENET news groups, WWW sites, or any other type of media. Each InfoCast Server is responsible for a particular geographical region. They disseminate content from various sources, scale the content for transmission over the available bandwidth, run filters and transmit the digested content to clients. In our prototype, InfoCast clients are connected to InfoCast servers via a GSM network.

We experimented with many new ideas and technologies with InfoCast. Some interesting ones are:

- Hybrid Media - InfoCast uses both packet switched and circuit switched cellular for data transfer to strike a balance between power management, economy, and rich, real time data types. Either point to point or cell level broadcast Short Messaging Service (SMS) is used to send snippets of information to all mobile clients in an area. These snippets are called Infobites and are effectively those "pending notifications" previously described. The user can choose to retrieve full or richer versions of the snippets by activating broadband network services such as circuit switched cellular.

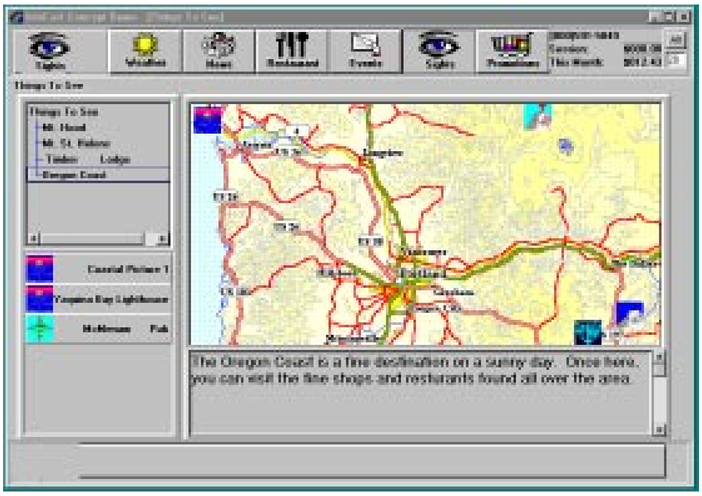

Figure 4: User Interface For InfoCast

InfoCast has a rich user interface with maps (for displaying location sensitive data such as restaurants), icons, and logos using a cache of objects stored locally on the notebook PC. The snippet only needs to refer to these icons or maps by integer ID's, saving much bandwidth. The client uses broadband connections to retrieve objects not in the local cache and get changes to the cached objects. This hybrid technique gives remarkable perceived bandwidth to the low bandwidth SMS channel.

- Information Filtering - In addition to traditional self learning text based filters, the InfoCast server also runs location and temporal filters. Useful location sensitive content include traffic and weather reports, nearby restaurants, movie theaters, and hotels, community event calendars, and points of interest. Cell level broadcast is a very efficient means of transmitting this information. Temporal filters would send "just in time" information such as advertising a restaurant's daily specials around dinner time. We obtain the location information by accessing the user's current cell ID. This information is available in most wireless networks although software modifications and security mechanisms are needed to allow application access. The Cell ID is also available in the wireless devices, but until now this information has never been accessible to applications. We have lobbied ETSI to include AT commands for GSM terminals to allow application access to this information in GSM specification 7.07 [8]. We highly encourage other device manufactures to consider adding this capability.

- InfoBite Protocol. - This is an application level protocol built above NBS. InfoBites are snippets of information containing ultra compressed fields for content branding, time to live, client categorical information, title, and one or more InfoActions. InfoActions are programmable actions the user can take, such as dialing numbers, sending feedback to content providers, placing reservations, or retrieving richer versions of the story (typically via a URL). The client decides how an InfoBite is presented to the user. Figure 4 shows one such rendition. InfoBites also contain control messages which allow the client to determine its location and synchronize its object cache with the server. The protocol is designed for extensibility and can accommodate any media packet or data payload size.

Related Work

AOAC isn't a new idea, recently it has received much attention for the home PC, as described in the OnNow [11] specification. Wireless PDAs such as the Sony MagicLink and the Nokia Communicator 9000 also adopt this usage model. However, the notebook PC faces a set of different tradeoffs from the home PC in power and communication availability, and its compute power and display can handle richer information than PDAs. We have been focusing on usage and problems for the notebook PC platform, some of which are specific to Windows* and others (like Narrowband Sockets) can be generalized to other operating systems.

We have learned much from previous work in disconnected or weakly connected operations. Examples include Noble[2], Tso [6], Terry[5], and Joseph [1]. We are looking to apply many of these ideas in our system for disconnections caused by radio interference or denial of service.

Schilit[3] described location sensitive applications. However, our work is unique in that we are using information readily available in the wireless infrastructure and do not require the addition of new user equipment (such as a GPS) or the installation of new infrastructure, such as for the ParcTab [4]. Furthermore, much of the location management ideas suggested by Schilit et al can also be used in our system.

PointCast* is an Internet based personalized news service program. But it is based on automated (using screen saver) client pull which causes severe network congestion when many clients pull at the same time, e.g. during lunch time. "Infocast" achieves better load management by scheduling "pushing" to avoid network congestion, and using cell level broadcast to more efficiently use the bandwidth. Many paging based wireless news services are also available commercially, such as the AirMedia* service from ExMachina (www.exmachina.com/amiinfo.shtml). These systems can easily benefit from adding mobility management and location specific filtering.

RIM's RadSock [13] is a bsd sockets based implementation for RIM's 2 way paging services similar to NBS. NBS not only incorporates more features available through Winsock2 (such as quality of service parameters), it also supports an extensible architecture that supports many other forms of narrowband messaging, adding SMS from digital cellular networks and other 2 way paging services.

The Wireless NDIS Extensions found in the Microsoft* NDIS 4.0 [10] standard, proposed by the PCCA added many features for controlling wireless specific device attributes. These extensions are in kernel mode, primarily intended to be used by Winsock2 protocol stacks rather than user mode applications. They are designed for packet radio devices such as CDPD or Ardis. The Unimodem Extensions we proposed adds features required by cellular modems, and are at user level, allowing application programs or control panel applets to easily manipulate wireless devices.

Conclusions

We articulated a usage model evolution path for the notebook PC based on the availability of new communication pipes. We argued that wireless pipes present potentially enormous opportunities by enabling the Always On, Always Connected usage model on notebook PC's. AOAC notebook PC's provide ease of use benefits as well as eliminate much latency associated with the user initiated connectivity model (pulling) with an automated connectivity model (pushing). A key realization here is that in addition to broadband networks, an inexpensive but ubiquitously deployed wireless medium such as narrowband messaging is required for data pushing to have a viable business model. We described a research vehicle which contained many innovations for wide area information delivery, and demonstrating the viability of location sensitive applications using existing infrastructure. We described our solution for integrating wireless pipes into existing PC communications software infrastructure, and a new transport protocol for transmitting data over narrowband messaging channels. We believe these are but some of the many components which need to be put in place to realize the AOAC goal for the notebook PC. Intel is actively working with the PC industry and the Telecommunications industry to accelerate this evolution.

Acknowledgments

We are grateful to Patrik Gustafsson and Petri Poyhonen of Nokia who contributed a great deal to the NBS work. Petri Heinonen (Nokia), Dave Andersen (Intel), Joe Decuir (Microsoft), and Dan Perry (Microsoft) provided vital input and support for the extension work in the Unimodem, NDIS, and Winsock2 areas. And Geoffery Peters (Intel) was a key player during the early conceptual stages and prototyping of "infocast."

References

[1] A. Joseph, A. deLespinasse, J. Tauber, D. Gifford, and F. Kaashoek. Rover: A Toolkit for Mobile Information Access. In Proceedings of the Fifteenth Symposium on Operating Systems Principles, December 1995.

[2] B. Noble, M. Price, and M. Satyanarayanan. A Programming Interface for Application-Aware Adaptation in Mobile Computing. In Proceedings of the Second USENIX Symposium on Mobile & Location-Independent Computing, April 1995.

[3] B. Schilit, N. Adams, and R. Want. Context-Aware Computing Applications. In Proceedings of the Workshop on Mobile Computing Systems and Applications, December 1994.

[4] B. Schilit, N. Adams, R. Gold, M. Tso, and R. Want. The PARCTAB Mobile Computing System. In Proceedings of the Fourth Workshop on Workstation Operating Systems, October 1993.

[5] D. Terry, M. Theimer, K. Petersen, A. Demers, M. Spreitzer, and C. Hauser. Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System. In

Proceedings of the 15th ACM Symposium on Operating Systems Principles, December 1995.

[6] M. Tso. Using Property Specifications to Achieve Graceful Disconnected Operation in an Intermittent Mobile Computing Environment. MIT M.S. Thesis, (also available from Xerox PARC, technical report CSL-93-8), June 1993.

[7] (DTE - DCE) Interface for Short Message Service (SMS) and Cell Broadcast Service (CBS), (GSM 07.05), European Telecommunications Standards Institute, May 1996

[8] AT command set for GSM Mobile Equipment (ME); (GSM 07.07), European Telecommunications Standards Institute, May 1996

[9] WIN32* Messaging Application Program Interface (MAPI) Programmmer's Reference. Microsoft Corporation, 1996, WIN32 Software Development Kit, April 1996.

[10] Network Driver Interface Specification (NDIS) Version 4.0. Microsoft Corporation, October 1995.

[11] Advanced Configuration and Power Interface Specification, Intel, Microsoft, & Toshiba Corporations, 1996, (http://www.teleport.com/~acpi/)

[12] R. Want, B. Schilit, N. Adams, R. Gold, K. Petersen, D. Goldberg, J. Ellis, and M. Weiser. The PARCTAB Ubiquitous Computing Experiment.

[13] RAD-I/O Connectivity Tools, v. 2.51, February 1996. Available from Research In Motion, radinfo@rim.net.

[14] UniModem V Voice Modem Functional Specification. Microsoft Corporation, (http://www.microsoft.com/kb/softlib/mslfiles/unimodv.exe), May 1996

[15] V.80, In Band DCE Control and Synchronous Data Modes for Asynchronous DTE, ITU PN-3319, June 1996.

[16] Win32 Implementation of Windows Telephony (TAPI Version 2.0). Microsoft Corporation, January 1996.

[17] Windows Sockets 2 Application Programming Interface. Intel Corporation (http://www.intel.com/ial/winsock2), May 1995.