

Nettech Systems, Inc. Smart IP™ White Paper

Connecting Mobile Workers to your Enterprise via Wireless Computing

Wireless data technologies are making significant strides. While the mainstream market has not yet adopted wireless data, it has advanced to the point where software developers, systems integrators and in-house corporate developers are beginning to evaluate how to incorporate wireless data accessibility into their enterprise-wide mission critical applications.

The wireless data market has been slow to emerge because of the challenges faced with running existing applications over wireless networks and the time and investment required to modify these applications or to custom-develop mobile applications. There has needed to be a driving requirement to have immediate access to information in order to justify such an investment. That is why wireless data communication has gained a strong foothold in many “dispatch-oriented” vertical markets such as field service, transportation, utilities and public safety, where the return on investment is high.

Recently many “sales-oriented” organizations in markets such as healthcare, insurance and finance, have also begun to see a need to provide their mobile workers with real-time access to vital corporate information through messaging applications. In today’s competitive marketplace, waiting until you find a phone or you are back in the office to access data simply isn’t fast enough.

With the emergence of browser-based applications and Internet Protocol (IP)-based networks, this data is increasingly accessed through the Internet, intranets and custom-designed IP-based applications. Most organizations want to protect their investment in these applications by simply extending them over wireless networks to allow workers to access this mission-critical information in real-time. In addition, their mobile workers want access to e-mail, information services and other complementary applications.

Wired vs. Wireless Computing

The wired and wireless computing worlds operate under completely different paradigms. The wired world assumes a constant connection with high bandwidth and increasingly faster speeds. A wireless environment functions via intermittent connections over a narrow bandwidth pipe, operating at much slower speeds. These fundamental differences introduce a number of difficulties when organizations attempt to extend traditional “wired” applications over wireless networks.

To overcome the challenges faced in a wireless environment, software developers have had to custom develop optimized mobile applications, many times utilizing wireless middleware to manage the communication between the application and the wireless network.

Now, instead of custom developing an application, organizations are looking to simply extend their existing TCP/IP applications over IP-based wireless networks and maintain compatibility with their back-end systems. However, these applications simply don't perform well under the more adverse wireless conditions.

To address these issues, Princeton, New Jersey-based Nettech Systems recently introduced a wireless communication product, called Smart IP. Smart IP allows TCP/IP-based applications, such as browsers, FTP, email and custom-developed IP applications, to run reliably and efficiently over virtually all-wireless networks without modification to the application.

Wireless TCP/IP Obstacles

Applications such as browsers, e-mail and others utilize the TCP/IP transport protocol to communicate over the internet, intranets or LANs (and now, wireless networks). However, these applications tend to be "chatty." Also, since TCP/IP was designed for an always connected, high-speed, wired environment, it is overhead laden, making it difficult to run in a wireless environment.

For example, to establish a connection or "socket" TCP requires a three-packet-overhead "handshake." It also requires another three packets to disconnect. When you are talking about a browser-based application where every HTTP request (which is required for each graphic or file included in a page) opens and closes a socket, the overhead is excessive. This is particularly true in a wireless environment where you are paying according to the amount of data sent and each transmission uses battery power and slows the communication.

TCP also uses a "windowing" protocol that regularly sends acknowledgement packets to let the sender know the progress of the communication. With a wide pipe, numerous acknowledgements are not a problem. However, in a wireless environment, the difficulty comes in the number of acknowledgements that TCP sends. Typically, TCP sends one acknowledgement packet for every one to two packets of data that are sent. Once again, this is extremely high overhead for communication over a wireless network.

Also, because TCP/IP assumes it is communicating between two devices over a wired network the TCP protocol expects to receive these acknowledgements very rapidly. There are timers within the TCP protocol that cause additional retransmissions or a connection to have to be reestablished if these acknowledgement packets aren't received in a set time because it is assumed that the connection has been disrupted. In a wireless environment, where throughput speeds are much lower and users frequently move in and out of coverage or operate in "fringe" coverage conditions, the timers "expire," causing connections to drop and restart. To make matters worse, reestablishing that connection yields yet another three-packet handshake.

When operating under fringe wireless conditions, it is not uncommon for some packets of data not to make it through. With TCP, when this occurs it automatically resends all

packets since the missing one. This, once again, adds to your costs and diminishes the battery life.

Some organizations choose to overcome some of these obstacles by substituting the TCP protocol with UDP (User Datagram Protocol), which is part of the TCP/IP suite of internet protocols. While this transport does not require handshakes, it also does not provide acknowledgements, making it truly a “send and pray” protocol. This simply isn’t acceptable in most wireless applications because mobile users need to be sure their message got through and don’t need the hassle of manually resending messages.

To top it off, TCP applications themselves can often be very “chatty,” generating many communication transactions between devices. For example, these applications may require excessive handshakes (similar to the TCP protocol handshakes) to establish the communication. Also, many terminal emulation-based applications have many different screens and since it assumes a “dumb” terminal is being used it sends the data for the entire screen with each keystroke. Finally, TCP applications frequently use “fat” data formats that could be too large to send over the air cost-effectively.

While the TCP/IP protocol clearly isn’t designed for wireless communications, it has become a “de facto” industry networking and communication standard with the advent of internet and intranet computing. So the question becomes, if you want to leverage your TCP/IP applications for use in the field, how do you make it work reliably and efficiently?

Smart IP Solves the Problem

Nettech Systems, a leading supplier of wireless middleware, has recently introduced Smart IP, a product designed to overcome these obstacles. Smart IP allows TCP/IP based applications (including browsers, e-mail, FTP and custom-developed applications) to run efficiently over wireless networks without modification.

Prior to the introduction of Smart IP, users had two choices. They could either run their TCP/IP applications over IP-based networks, such as CDPD, without modifying them, albeit inefficiently, *or* modify their application to support the protocol of non-IP based networks such as ARDIS, BellSouth Wireless Data and others. To get an efficient and reliable mobile application, some level of development was required for either option. Now, not only can users run their applications efficiently over wireless IP networks, they can also run them over a plethora of other wireless networks without changing the application.

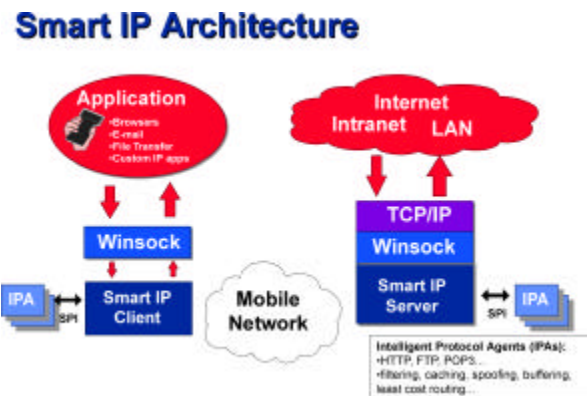
Smart IP optimizes the wireless data communication on two levels. First, it substitutes the TCP/IP transport protocol with Nettech’s optimized wireless transport layer. Next, it allows developers or corporations to create Intelligent Protocol Agents, which are used to streamline the application’s transactions and optimize high-level protocols.

Smart IP overcomes the obstacles associated with running TCP/IP applications over the air and provides users with improved performance, reduced airtime expenses and extended battery life. Overall, Smart IP:

- Lets organizations immediately run their existing standard Internet and Intranet applications, such as browsers, file transfer protocol (FTP), e-mail and Java and ActiveX applets, over any wireless network immediately without modification;
- Allows companies to develop wireless applications using the industry standard Winsock API;
- Provides reliable, effective data communication optimized for wireless by cutting the number of packets sent by as much as 80 % and reducing the amount of data being transmitted by 30-to-60 %;
- Allows enterprises to develop additional optimizations, such as message filtering or transaction buffering, to reduce the application's "chattiness" and further improve its ability to operate in a wireless environment.

How Smart IP Works

Smart IP consists of client and server software that use the Winsock API to communicate with TCP/IP applications. On the client side, messages are routed from the Winsock API to Smart IP (instead of TCP/IP) using proxy server technology, the same technology used to implement firewalls (see architecture diagram). During configuration, Smart IP is instructed to "listen" for TCP/IP traffic on certain ports. Through a simple configuration process, applications are instructed to communicate to those ports, which serve as the proxy server. The application believes it is talking to TCP/IP via the Winsock API. In reality, because Smart IP fully supports the Winsock API, that traffic is transparently redirected through the mobile network using the Smart IP transport protocol, which is highly optimized to operate efficiently in a wireless environment.



For example, assume that Smart IP client is running on a wireless network. The client has TCP/IP stack installed on it. However, there is no (active) connection outside of this machine. Instead, a Smart IP client is running on the machine at TCP port 5000. The user is running a browser and wants to talk to <http://www.NettechRF.com>. The browser is configured to use 127.0.0.1 port 5000 as a Proxy Server. So when it tries to connect to NettechRF.com, it connects through the Smart IP client, instead of TCP/IP, telling the Smart IP client where it wants to connect. The Smart IP client passes the data for the connection on to the Smart IP server over the wireless network using the optimized transport and the Smart IP server makes the actual connection to NettechRF.com.

After the communication passes through the wireless network it arrives at the Smart IP server. The Smart IP server then passes the data to and from the real server that the user wishes to connect to. The Smart IP server creates the TCP or UDP connections on

behalf of clients and keeps track of the type of connection (i.e. which method was used to create it), along with the client's address and port number. The Smart IP server also terminates connections, if necessary, based on either "close connection" messages, or an inactivity timer for each connection.

The Smart IP server can be configured to use another Proxy Server, in which case all TCP requests will go to the Proxy Server unmodified. This allows Smart IP to be used in conjunction with corporate firewalls, maintaining the security of the enterprise.

Smart IP also offers a Service Provider Interface (SPI) which can be used to call Intelligent Protocol Agents (IPAs). IPAs are small programs, either custom developed or offered by Nettech, that run on both the client and server and can be used to streamline application transactions or optimize high level protocols without modifying the application. IPAs are written for a specific application or protocol to reduce the "chattiness" of the application by performing such functions as caching, spoofing or batching.

The IPA takes the form of a DLL, which is loaded and called by the Smart IP client and/or server. It participates in all traffic flowing through Smart IP for a given application. Each IPA is defined to handle a specific protocol such as HTTP, FTP, GOPHER, WAIS, TELNET, etc., a user-defined protocol, or simply an application, recognized by the port number specified in the Smart IP configuration.

When available, an IPA is called by Smart IP (via the SPI) whenever a client message to that protocol, or a response from the server to that protocol, is received. When Smart IP receives the data, it will first route it to the IPA for the IPA to perform whatever function is required. Then, when conditions of the IPA are met (e.g. an actual request for data transfer is received), the message will be passed back to Smart IP for transmission over the wireless network.

For example, for each request passed to an IPA, the IPA may initiate one of the following actions:

- Pass the data onto the receiver (the data may be modified by the IPA)
- Do nothing at this time
- Indicate a response to the sending application (e.g. from cached data)
- Terminate the connection
- Create a slave client or server data connection

The Smart IP runtime includes an IPA for File Transfer Protocol (FTP). The optional Smart IP Software Development Kit (SDK), which is used to create IPAs, also includes the source code for the FTP IPA and other sample applications to assist in the creation of additional IPAs.

Solving the Transport Problem

Smart IP offers many significant advantages over running native TCP/IP over wireless networks (see chart below). First, it greatly reduces the overhead. Where TCP/IP

requires three packets to set up and take down a connection, Smart IP does not require any. Also, while TCP/IP sends an acknowledgement every one to two packets, Smart IP sends only one acknowledgement per message.

In wireless communication, it is not infrequent for some packets of the message to not make it through when users are in fringe conditions or moving in and out of coverage. When this happens using TCP/IP, it resends the entire message since the lost packet, whereas Smart IP resends only those packets that are missing.

Feature	Smart IP	TCP	UDP
Call setup (e.g. open socket)	0 packets	3 packets	0 packets
Acknowledgements	1 per message	At least 1 every 2 packets	None
Retries	Resend only missing packet(s)	Resend since lost packet	None
Call take down (e.g. close socket)	0 packets	3 packets	0 packets

Recent tests show that these efficiencies serve to cut the number of packets being transmitted by as much as 80 percent and reduce the amount of data being transmitted by 30-to-60 percent (see chart below). This, in turn, extends the battery life because not as much energy is being consumed by transmitting data and saves money since, with wireless networks, you pay for the amount of data you send.

In a side-by-side comparison conducted by Nettech Systems, Smart IP with the FTP IPA was tested versus native TCP running FTP as well as Smart IP versus native TCP for HTTP over a CDPD network. Here are some sample results:

FTP (Get 10,000 bytes):

- Native FTP: 53 packets, 12,879 bytes
- Smart IP: 10 packets, 6,485 bytes

HTTP (One page, eight images):

Without images:

- Native HTTP: 39 packets, 9,634 bytes
- Smart IP: 6 packets, 2,947 bytes

With images:

- Native HTTP: 314 packets, 67,688 bytes
- Smart IP: 80 packets, 47,421 bytes

Smart IP was designed to function in a wireless environment. Therefore, it effectively handles issues such as fringe conditions and moving in and out of coverage, by adapting automatically to changing conditions. The transport used by Smart IP can detect these conditions and counteracts them by slowing down the transmission and extending the

timers to avoid losing the connection. When conditions improve, the transport automatically speeds up the transmission. The capability effectively extends the coverage and avoids dropped calls, thereby providing a better overall user experience.

Using Intelligent Protocol Agents

To create an IPA, you first need to fully understand the data flow of the application. This will let you see where efficiencies can be added by doing things such as caching, spoofing or batching. IPAs can be developed as a DLL using standard programming tools such as Microsoft Visual C++ 5.0.

Smart IP loads and calls the IPA DLLs via the Service Provider Interface (SPI). The IPA then performs its function and sends the data back to Smart IP using a callback routine.

The IPA must first be initialized using the *Ipalnit* function. This function is called whenever a new connection is made and is used to identify the connection and specify the address of the Callback function.

After the IPA is initialized and it receives a “request” or accepts a connection with the application, Smart IP calls the *IpaNewAppData()* function of the IPA. In this function, the IPA usually converts data and returns them back to Smart IP by invoking the Callback (*IpaActSendRFData*), which causes the corresponding Connection object to put data in the outgoing queue of Smart IP.

The IPA may sometimes know what to send back to the TCP application or it may ignore the data if it does not mean anything useful to the peer TCP side. For example, if the IPA on the client is processing data that does not mean anything to the server, it will ignore the data. The IPA may also prepare standard replies for some data or cache replies and keep correspondence between requests and cached replies. The IPA can do so by sending (spoofing) a reply without initiating an actual transmission. It does this by asking Callback to do the *IpaActSendAppData* function.

When Smart IP receives a reply from the peer side, it causes the IPA function *IpaNewRFData* to be called. In the simplest case, the IPA decodes the data and then invokes the Callback *IpaActSendAppData* function to cause the data to be passed to the application.

There are a number of other IPA functions that are callable from Smart IP that are used to handle errors and connections and pass parameters. These numerous functions provide IPAs with great flexibility, allowing them to be easily created for every TCP/IP application to add efficiencies and streamline the applications’ transactions in order to provide greater performance over wireless networks.

Smart IP Support

Smart IP supports a broad range of wireless networks including ARDIS, BellSouth Wireless Data, CDPD, DataTAC 5000/6000 public mobile data networks; Cellular

(AMPS/CDMA/GSM) and wireline circuit switched networks; local area networks and wireless local area networks; Norcom Networks packet satellite; and Ericsson EDACS and Motorola Private DataTAC private packet radio networks.

Smart IP client runs on the Windows 3.1, Windows 95, and Windows NT (3.5 or later) operating systems. Coming soon, Smart IP client will also support Windows CE. The Smart IP server requires a Windows NT Server or Workstation 3.5 or later.

It has been tested to prove compatibility with a number of standard TCP/IP applications and protocols including Internet Explorer, Netscape Navigator and Communicator, Qualcomm Eudora, Microsoft Exchange Server 5.0 (accessed via the web using a browser), EnterpriseLink SmartTran, various FTP clients, HTTP, HTTPS (secure sockets), FTP, SMTP, POP3 and IMAP4.

Smart IP in Use

While everyone agrees that wireless networks may not yet be the best medium to recreationally browse graphics-intensive web sites, utilize multimedia applications or download very large files, they are ideal for providing real-time access to critical information. Smart IP makes this happen efficiently and reliably.

Organizations will use Smart IP to provide their mobile workers with wireless access to intranets, optimized web information services, e-mail and other critical corporate information. By using Smart IP, organizations will be able to wireless-enable existing applications without modification, protecting their investment in those applications; lower airtime expenses; provide better performance; extend battery life of mobile devices; and most importantly – allow them to increase productivity and improve customer satisfaction.

Smart IP is available from Nettech Systems. Nettech Systems can be reached by calling (609) 734-0300 or on the web at www.NettechRF.com.

InstantRF is a registered trademark of Nettech Systems, Inc. Smart IP is a trademark of Nettech Systems, Inc. All other names may be trademarks or registered trademarks of their respective companies.