BellSouth Wireless Data

Internet Access Service
Description and
Programming Guide
Version 1.0

BellSouth Wireless Data

10 Woodbridge Center Drive
Woodbridge, NJ 07095
732-602-5500

BELLSOUTH®

Table of Contents

Service Description

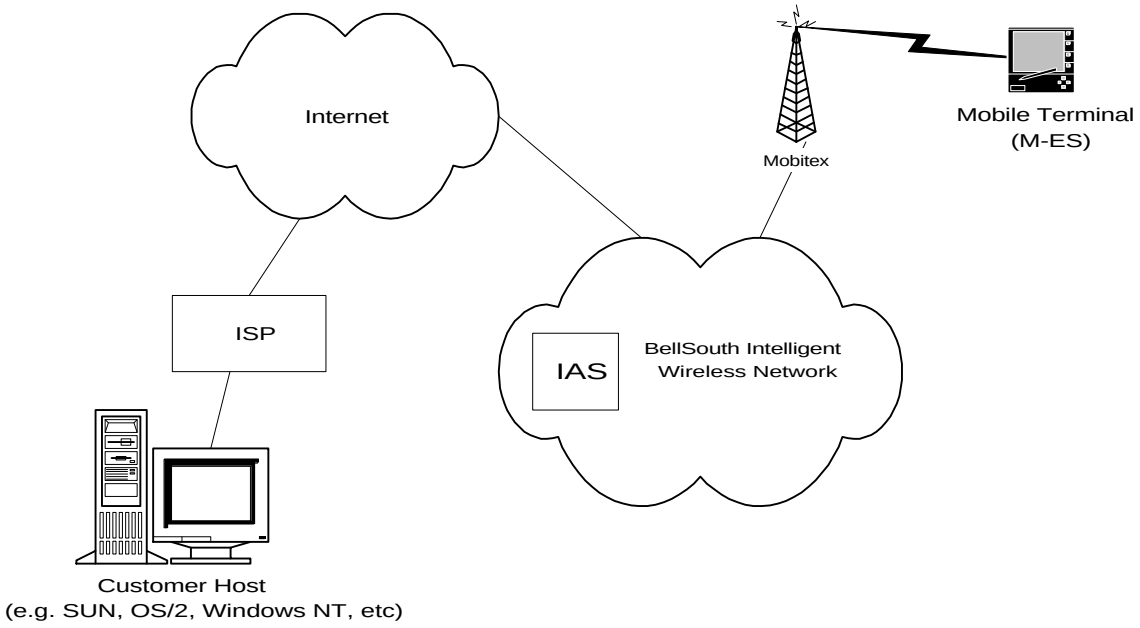
Introduction

The Internet Access Service (IAS) is a host connectivity option designed to allow a customer's host application to connect to the BellSouth Wireless Data Intelligent Wireless Network via the Internet. It is easy to implement because it only requires that the customer have a connection to the Internet with a fixed IP address from the Internet Service Provider (ISP) of their choice. No leased lines or special hardware are required. It is also less expensive than a leased line X.25 connection.

The IAS is fully interoperable with other BellSouth services including the Multi-Network Access Service and Mobigate. It is representative of BellSouth's continued commitment to providing our customers with flexible wireless solutions.

Typical Customer Connection Scenario

A Typical IAS Connection

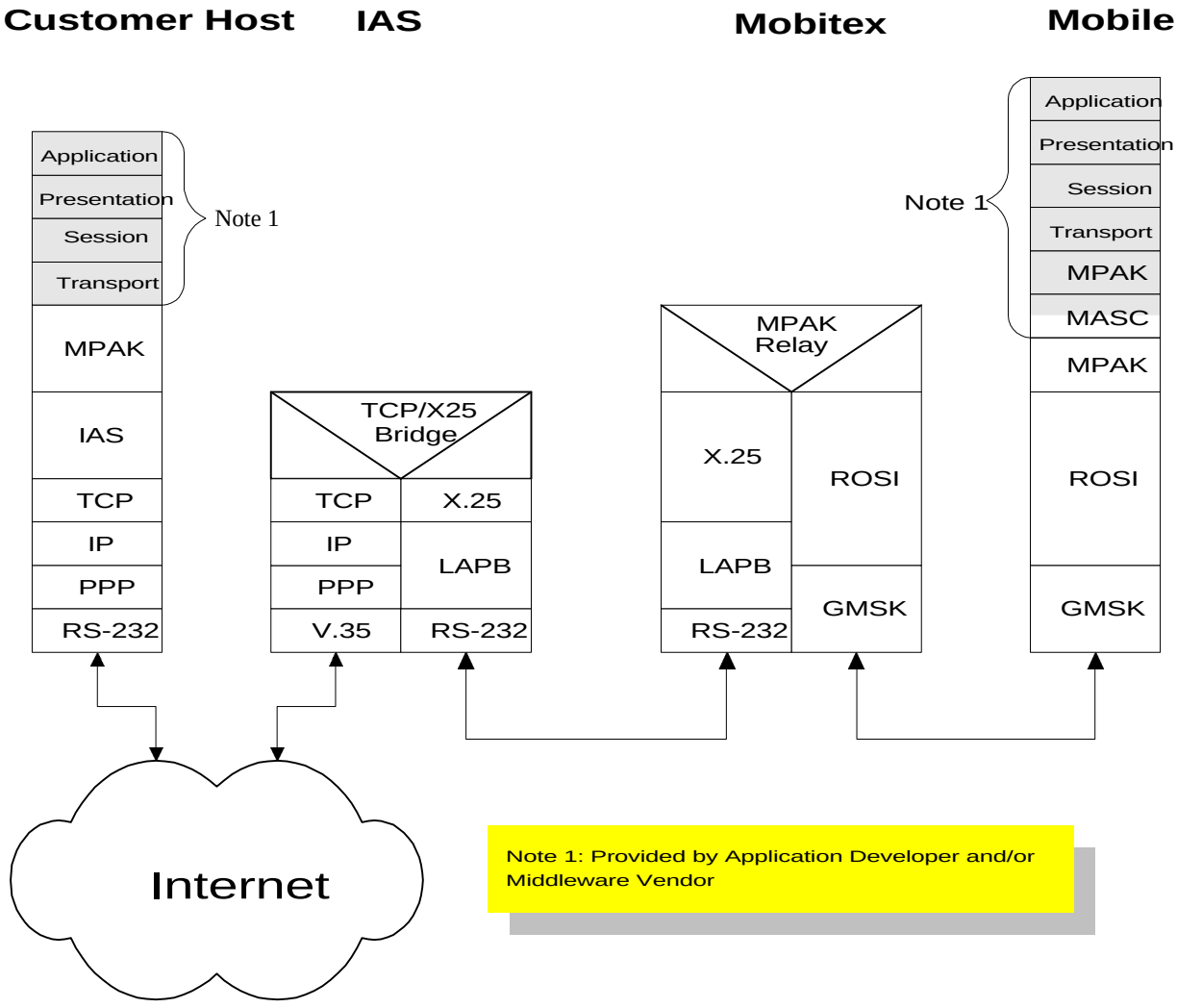


As can be seen from the diagram above, the IAS is an integral component of the BellSouth Intelligent Wireless Network, which provides an expanding set of connectivity options on both the wireline as well as the wireless side of BellSouth's network.

Customer Applications

As previously noted, the customer host connects to BellSouth through the Internet via an Internet Service Provider of their choice. The customer host accesses the BellSouth network via an application which embeds MPAKs into a TCP data stream. This eases application development by virtue of the fact that TCP/IP programming is generally easier than its X.25 counterpart. It should be

noted that customer data is not passed as TCP/IP data through BellSouth's network. Rather, TCP is used as a conduit through the Internet to deliver customer data to BellSouth. The diagram below shows the protocol layering. .



Although the protocol components on the customer host look complicated, in actuality they are not. The customer application consisting of the top 4 components are common to any application utilizing BellSouth's network. The MPAK component is available from BellSouth as portable code and the TCP/IP and PPP components are often bundled with the operating system, or available as a reasonably priced add-on. This leaves only the IAS component which is described in more detail below.

This utilization of TCP/IP imposes a unique requirement on the customer host application (the Mobile application is unaffected). Because TCP is a "stream" protocol, there are no data boundaries based on a particular quantity of data; data "streams" through the TCP connection. Mobitex on the other hand, being a datagram oriented system, can only deal with data which has identifiable boundaries. The native unit of data for Mobitex with identifiable size is an MPAK. Thus, it is necessary for the host application to inform the IAS where MPAKs begin and end as they flow through the TCP stream.

This task is achieved by prepending a simple 4 byte header to each MPAK as it is injected into the TCP stream. This 4 byte header contains a value indicating the length of the MPAK which follows. Likewise the customer application must be prepared to capture this 4 byte header from data arriving on the TCP stream, and using the length contained therein, extract a complete MPAK.

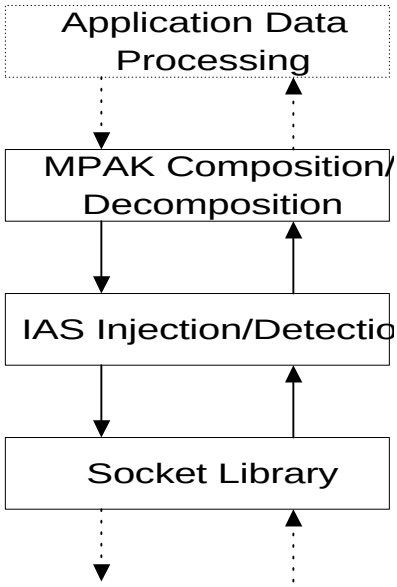
Host Programming Guide

Introduction

This section is intended provide an outline of the steps required to utilize the Internet Access Service within a customer host application. It is intended for programmers and others knowledgeable of sockets programming. Simple examples are provided to facilitate comprehension.

Host Application Structure

The diagram below shows how an application should be partitioned. This document only covers the injection and detection of user data (e.g. MPAK) into/out of a TCP stream. Adjacent layers are shown only to provide the reader with a perspective of where the IAS utilization code fits.



The Host is the Client

The customer host on which the application executes is always the client. It must open a TCP connection to the IAS and leave it up for mobiles to initiate communications with the application. There are no provisions for the IAS to initiate a TCP connection to the customer host. Further, this would require the customer host to act as a server which is certainly a more complicated programming task.

Header File

Below is shown the header file for using the IAS. This header file only includes the macros and data structures necessary for constructing the 4 byte header. It does not include the header files for sockets programming, nor does it include any information regarding the composition or decomposition of MPAKs.

```
//--------------------------------------------------------------------------
// Frame Header
```

```
//--------------------------------------------------------------------------
typedef struct xotfhdr {
        unsigned char version;  // version associated with frame header
        unsigned char flags; // frame control flags - MUST BE ZERO (0)
        unsigned short length;  // # of bytes following header
} XOTFHDR, *PXOTFHDR;
#define XOFHDRSZ        sizeof(XOTFHDR)


//--------------------------------------------------------------------------
// Version Define
//--------------------------------------------------------------------------
#define XOTVER_STD                      0x02
```

It should be noted here that the *length* field of the *xotfhdr* data structure must be converted to network byte order by the programmer.

IP Addresses and Port Numbers

The IAS Internet Protocol address and the corresponding port number for the IAS will be provided by BellSouth at subscription time. The programmer should make provisions to facilitate changing these parameters.

The customer must use a valid IP address for their host address; otherwise, the Internet will not be able to route IP datagrams properly.

Sample WINSOCK Code

Shown below is a series of code fragments for programming within the Microsoft Windows Socket Programming environment. These fragments should be modifiable to support other socket programming environments. Error handling code has been omitted for clarity.

Connect to the Internet Access Service

The following code fragment demonstrates the steps necessary to connect to the IAS.

```
SOCKET hSock;
INT err;
WSADATA wsaData;
SOCKADDR_IN remoteAddr;

// Start up WINSOCK
err = WSAStartup (0x0101, &wsaData);
if (err == SOCKET_ERROR)
        // check wsaData

// open the socket
hXSock = socket (AF_INET, SOCK_STREAM, 0);
if (hSock == INVALID_SOCKET)
        // handle invalid socket appropriately

memset (&remoteAddr, 0,sizeof (remoteAddr));

// “ServPort” assigned by BellSouth
remoteAddr.sin_port = htons (ServPort);
remoteAddr.sin_family = AF_INET;

// “IasIpAddr” assigned by BellSouth
remoteAddr.sin_addr = IasIpAddress;

// create a client side connection
err = connect (hSock, (PSOCKADDR) &remoteAddr, sizeof(remoteAddr));
if (err == SOCKET_ERROR)
        // handle connect error
```

.
•
•

Write Data

The following code fragment demonstrates the steps necessary to write data. Assume that *pMpakBuf* and *MpakLength* are a buffer pointer and integer count respectively passed into the code shown below.

```
UCHAR *pIasBuf;
int         pIasLength;

pIaslength = XOFHDRSZ + MpakLength;
pIasBuf = (PXOTFHDR) malloc(pIaslength);
if (!pIasBuf)
        // malloc failed - process appropriately

memset(pIasBuf, 0, (XOFHDRSZ );  // initialize the header
// set up the 4 byte IAS header
pIasBuf->version = XOTVER_STD;
pIasbuf->length = htons(MpakLength);

// now append the data passed in pMpakBuf to the new buffer
memcpy( (pIasBuf + 4), pMpakBuf, MpakLength);

// time to write the data into the TCP stream
if( send (hSock, pIasBuf, pIaslength, 0) != pIasLength)
        // not all bytes sent; handle error
```

.
•
•

Reading Data

Reading data is little bit more complicated than writing because the size of any received MPAK cannot be known beforehand. The recommended method of handling this situation is to post a read for just the 4 byte IAS header, and then based on the length contained therein, post a read for the remaining number of byte which should then constitute a complete MPAK. This process is continually repeated. The following code fragment demonstrates this.

```
UCHAR *pIasBuf, *pMpakBuf;
int         MpakLength;

pIasBuf = (PXOTFHDR) malloc(XOFHDRSZ); // malloc 4 bytes
if (!pIasBuf)
        // malloc failed - process appropriately

memset(pIasBuf, 0, XOFHDRSZ);

// post a read for 4 bytes
if( recv(hSock, pIasBuf, XOFHDRSZ, 0) != XOFHDRSZ) {
        // process the error
}
// presuming everything is OK upto this point, continue
// we got the header; now read it
if (pIasBuf->version != XOTVER_STD) { // check the version
        // handle an error in the IAS header
}
// get the length; convert from network to host order
MpakLength = ntohs(pIasBuf->length);

// get a buffer to hold the MPAK
pMpakBuf = malloc(MpakLength);
if (!pMpakBuf)
        // malloc failed - process appropriately

// finally read the actual MPAK
if( recv(hSock, pMpakBuf, MpakLength , 0) != MpakLength) {
        // process the error
}
```

.
.
.

Other Considerations

Due to the fact that the IAS provides service through the Internet, the user may experience temporary outages resulting in a loss of an established TCP session. These outages have many causes. For example, the customer's Internet Service Provider (ISP) may experience temporary problems, problems may arise in the various intermediate routers, or BellSouth's ISP may experience difficulties. Additionally, the local Telco may experience problems resulting in circuit loss any where along the path from customer host to BellSouth's switching center.

As conditions indicated above will most likely occur at some point in time, there are several tactics which can be used proactively to minimize the impact of service disruption and enhance usability.

Reconnect After Disconnect

A customer host application should generally attempt to re-establish a TCP session immediately after detecting the closing of a TCP session. The reason for TCP session termination should be a time-out condition, but other reasons may be possible. This approach insures the host application is always connected to the BellSouth Intelligent Wireless Network in order to receive mobile initiated traffic.

Use A Transport Protocol On Top Of MPAKs

The use of a transport protocol on top of the MPAK layer is strongly recommended under any circumstances, but is especially desirable when utilizing the IAS. The IAS uses TCP to deliver MPAK datagrams to a bridge which re-encapsulates them into X.25 packets. Thus, there is a transition from one connection oriented network to another. The bridge functionality of the IAS does not guarantee delivery end to end, so the need for a transport layer which does so is apparent. Further, it is desirable that the transport layer is oblivious to disconnects/reconnects at the TCP layer underneath it. This is actually a requirement for rudimentary session layer support.

Summary

There are several additional steps the host application programmer must complete to build an application which utilizes the Internet Access Service, but these steps are relatively straightforward and impose a minimal burden on a new design. For added reliability when using a telecommunications medium such as the Internet, additional facilities may implemented as deemed appropriate.