BlueSky: A Cordless Networking Solution for Palmtop Computers

Pravin Bhagwat*, Ibrahim Korpeoglu†, Chatschik Bisdikian*,
Mahmoud Naghshineh*, Satish K. Tripathi‡

*IBM T. J. Watson Research Center, P.0.Box 218, Yorktown Heights, NY 10598
†Computer Science Department, University of Maryland, College Park, MD 20742
‡Bourns College of Engineering, University of California, Riverside, CA 92521
Email: {pravinb,bisdik,mahmoud}@us.ibm.com
korpe@cs.umd.edu, tripathi@engr.ucr.edu

Abstract

Palmtop computers are rapidly becoming the popular platform of choice for running personal productivity applications, but their networking capabilities are still lagging behind user expectations. Due to memory size, cost, and power considerations most palmtop computers support only limited form of point-to-point communication, namely, connection via a PSTN modem or a serial RS-232 cable. Due to wired point-to-point nature of these interfaces, satisfactory solutions for multi-point communication and direct LAN connection do not yet exist.

The BlueSky project aims at providing a low-cost, low-power, indoor RF wireless networking solution for handheld devices. Our solution consists of two components; a BlueSky wireless attachment that plugs into the standard serial port of any palmtop device, and a LAN access point that acts as a layer 2/3 bridge between the wireless and the wired part of the network. Palmtop devices use dialup networking software and PPP to connect via BlueSky to the network. BlueSky enables MobileIP style seamless roaming between access points without requiring any changes to the communication stack of plamtop devices. Our roaming enhancements to PPP are implemented in the PPP server, the access point, and the BlueSky attachment, all of which are completely transparent to the palmtop device.

In this paper, we present our design rationale and implementation experience of building the BlueSky system. We show in what ways low cost, low power, and form factor constraints affect the choice of protocols and function placement in wireless networking systems. We use PPP based mobility solution as an example to illustrate what design adjustments and compromises one has to make to build a working solution. Based on our experience, we recommend making a slight modification to the dialup networking stack of palmtop computers. Our proposed modification offers an efficient, lower cost, and lighter-weight wireless networking solution which is particularly attractive for enabling mobility over emerging short range, low power wireless technologies, such as IrDA, HomeRF and Bluetooth.

1 Introduction

The majority of handheld computing devices, such as palmtop computers, consumer electronic devices, pagers, cellular phones, and digital cameras are equipped with serial port communication interfaces. Such devices can be connected to other computer systems using RS-232 serial cables or modem connected telephone lines. As palmtop devices become an indispensable part of our everyday life, more flexible wireless connectivity options are needed. In an office environment, for example, it is desirable to connect palmtop devices directly to the wired LAN infrastructure. Similarly, solutions are needed for facilitating communication among a collection of palmtop devices within a short range.

Wireless access enables mobility and ease of use. Due to power and cost constraints, it is desirable to equip palmtop devices with only short distance (10m - 20m) wireless links. Hence, a large number of access points should be installed to cover a usage area such as an office building. Such a solution implies that the wireless access and mobility will be confined to pockets of connectivity regions formed by access points. However, this is not a very significant limitation considering that these palmtop devices will be mostly used inside closed areas, such as office buildings, airports, shopping centers, hotel rooms, and homes. It is worth mentioning that the existing wireless LAN systems are not well suited for providing short distance wireless connectivity for palmtop devices. First, most palmtop devices are only equipped with an asynchronous RS-232 serial port interface which cannot be easily connected to a packet oriented PC card interface of wireless LAN systems. Secondly, most wireless LANs (such as 802.11 & hyperlan [12]) consume more power (100 mW - 1 W) than what AAA batteries can provide over a reasonable length of time and usage.

Since palmtop devices are inexpensive and consume very low power, any networking solution must have similar properties in terms of cost, power and complexity. Secondly, since short links cause frequent handoffs, achieving fast handoffs and minimizing control traffic are crucial design goals. Providing secure communication is another important design consideration. During each handoff, mobile devices should be authenticated and data confidentiality and integrity should be preserved. Since PPP is a connection-oriented protocol, after each handoff a new connection is established to the new access point. Typical connection establishment requires exchange of upto 10-20 control messages between PPP peers, causing high delays during handoffs.

In this paper, we describe the design and implementation of an experimental prototype called BlueSky which we have built with the specific aim of enabling very short distance wireless connectivity and personal area networking among handheld devices. We call it a cordless networking solution as it has been built using a radio module from a cordless phone. Its cost, range, and power consumption are comparable to a cordless phone. Users can plug-in the BlueSky adapter into the serial port of their palmtop devices and access networking services over the wireless link. The access to wired networking resources is provided using a BlueSky access point which acts as a layer 2/3 bridge between Ethernet and wireless medium. Once a connection is made, users can browse the web, run host-sync, print documents, send faxes, or access any other networking service.

BlueSky mobility solution also enables users of palmtop devices to roam between access points without dropping active connections. It achieves fast hand-offs and generates very little control traffic on bandwidth constrained wireless links. Our solution is based on keeping the PPP connection alive while the mobile device moves from the range of one access point to another. This is achieved by moving the PPP peer function away from all the access points and aggregating it into a server located in the Intranet. A simple tunneling protocol is used to forward PPP traffic from the access point to the PPP server. When the mobile device connects to a new access point, a new tunnel from the access point to the PPP server is established and the PPP traffic from the server is redirected over the new tunnel. The end-to-end PPP session state at the client as well as the server is not affected by mobility or tunnel redirection. Thus, the overhead of PPP session establishment is avoided. The mobility solution can be viewed as a layer 2 link emulation protocol which provides a virtual link between the palmtop device and the PPP server. Even after a palmtop device moves, the virtual link connection between the device and the PPP server is kept intact. The most attractive feature of the BlueSky solution is that no new software or communication stack upgrades are needed. Handheld devices view the BlueSky wireless attachment as a modem and use the existing dialup and PPP software to connect to the network. Our roaming enhancements to PPP are implemented in the PPP server, the access point, and the BlueSky attachment. The added functionality is completely transparent to the end device.

The paper is organized as follows. In section 2 we present some background on PPP and describe the limitations of the existing wireless systems. Section 3 describes the components of the BlueSky system, including the attachment, the access point and the MAC protocol. Section 4 describes how a PPP based seamless roaming solution is built using the BlueSky components. Finally, in section 5 we describe the relevance of our work in the context of the emerging short range wireless technologies and present our conclusions.

2 Background and Design Considerations

The Point-to-Point Protocol (PPP) [17] is the most commonly used protocol for connecting palmtops and notebook computers to the Internet. PPP was originally designed to connect PCs via circuit-switched serial lines, such as phone wires, to the network. Over the years, PPP has evolved to include support for authentication, encryption, dynamic address assignment, and header as well as data compression. For mobile computing devices these features are of paramount importance.

For a network administrator it is absolutely critical to authenticate a mobile user before providing access to the networking resources. Users need to protect their data against the threat of eavesdropping over wireless links. The IP address allocation feature of PPP eliminates the need to run DHCP client [10] on the mobile user device. Finally, the header and data compression features of PPP enable efficient use of bandwidth over wireless channels.

PPP is already widely deployed on pervasive devices, notebook computers, PCs, workstations, servers, telecommunication systems, and routing and switching equipment. It is anticipated that PPP will continue to be the protocol of choice for connecting pervasive devices to the network. As new wireless technologies (indoor and outdoor) are widely deployed, efficient and low cost methods of leveraging PPP connectivity will be needed.

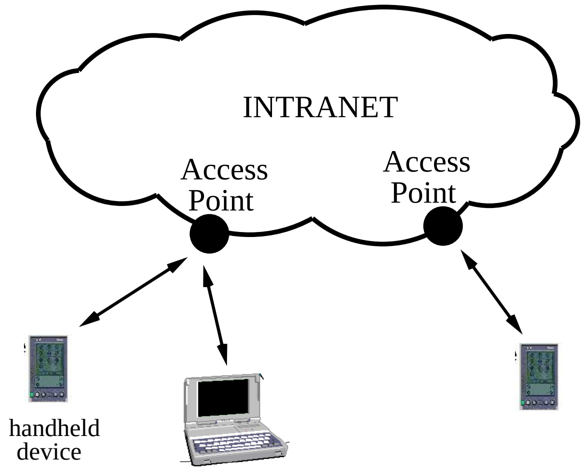

Figure 1: Short Range Wireless Link

Figure 1 shows an example scenario involving palmtop computers and a notebook computer connected via wireless links to LAN Access Points. Each computer is using dialup PPP software to connect to the nearest access point. The scenario of Figure 1 is similar to a computer connected to a PPP server via an RS-232 cable. The AP is a PPP server and the wireless link is a replacement for RS-232. Similar to a PPP server, AP is responsible for forwarding IP packets between the network and the PPP connection.

Figure 2 shows the layers of the communication protocols involved when the PDA (Personal Digital Assistant) device is connected to the network via a wireless adapter. The IP packet is first encapsulated within the PPP packet and before the packet is transmitted through the serial port an HDLC-like framing [16] is applied. The wireless attachment receives a bit-stream through the RS-232 port and is responsible for transmitting it to the other end of the wireless link. When the wireless link is circuit oriented, extending a serial line over the wireless link is easy. However, the new short range wireless links being developed by different standards forums [1, 2] are packet oriented, not circuit-oriented. Emulating a circuit-oriented protocol over a packet based network can be complex and inefficient. In Figure 2 wireless MAC represents a generic packet oriented wireless link layer which is present in all packet oriented wireless link technologies. An RS-232 emulation layer presents a byte-oriented interface to the higher layers. The same layer also includes emulation for control signals such as RTS, CTS, and DTR between the two ends of the wireless link. In section 3.4 we describe the challenges of building a serial line emulation layer over the BlueSky wireless interface and the specific design simplifications that enabled us to reduce the complexity and cost of the attachment as well as the whole solution.

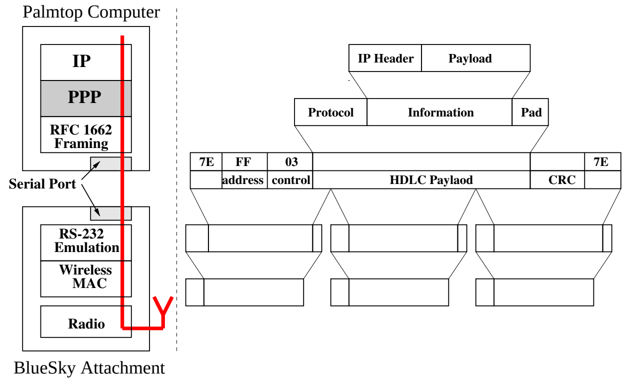

Figure 2: Palmtop Protocol Stack

Although various existing indoor and outdoor wireless network technologies can be adapted for enabling wireless access for palmtop devices, none of the technologies provide a cost effective solution. Wide area solutions such as CDPD, ARDIS, RAM are not suitable for indoor use. Metricom's Richochet wireless modem design is an attractive alternative [3, 8]. However, such solutions are intended to be a wide area wireless service, not a plug-n-play, build your own kind of a wireless system. They also consume approximately 100 time more power than any PDA device, are twice as expensive as any PDA, and do not provide any low-cost method of connecting a PDA directly to a LAN. A pair of wireless modems available in the industry today can be used to form a wireless RS-232 link, which can be used to connect a palmtop device directly to a PPP server. This solution, however, has three limitations: multiple users cannot access the network simultaneously; roaming is not supported; and the configuration is expensive.

3 The BlueSky System

From an end-system viewpoint, a BlueSky adapter appears like a modem attachment (similar to Metricom's Ricochet modem [3]). However, our design reflects several tradeoffs that were aimed at reducing power consumption, complexity, and overall cost of the solution. First of all, due to distance-power relationship, we preferred to use a short-range wireless link (10m -20m) [5]. We chose the same radio module which is used in today's cordless phones. These radios are extremely low-power (1 mW - 10 mW) and are already produced in mass quantities to afford a low price. An A/D converter added in front of the analog radio provides a digital link which can sustain up to 150 Kbps data rate using a simple FSK modulator. Using a MAC protocol, this bandwidth can be shared among 10-15 mobile devices. Since palmtops are memory, processing, and display limited, 150 Kbps link speed is enough to sustain the bandwidth requirements of these devices even at peak load periods.

3.1 Wireless attachment

The BlueSky wireless attachment plugs into the serial port of any palmtop device. It draws power from its own battery and stays powered-on (like a cell phone) up to 6-8 hours a day. The wireless link operates at 150Kb/s, but this bandwidth is shared among multiple users in a given cell. Thus, depending on the load in a given cell, mobile users will see bandwidth anywhere between 10Kbps to the maximum speed of the serial port. Our current prototype is built using a general purpose PC board and is therefore bigger than a PDA (see Figure 3), but its form factor can be reduced by porting its function to an embedded micro controller.

Figure 3: Wireless Attachment Box

Figure 4: Wireless Attachment Development Board

3.2 Access Point

The access point is a multiplexing/demultiplexing device. It acts as a layer 2.5 forwarder between the wired and wireless network. Multiple wireless devices within the communication range of the access point can simultaneously use the access point services and connect to the backbone. Note that an access point only controls one communication channel, which in turn is shared among multiple mobile units via a MAC protocol. Neighboring access points use different channels so as to avoid interference from neighboring/overlapping cells (similar to cordless phones). Since PDAs send PPP packets over the serial port, the access point acts as a PPP tunnel endpoint. It is capable of managing multiple simultaneous PPP sessions, authenticating users, and allocating IP addresses. As an optimization feature, the access point also supports a PPP proxy mode (as described in section 4.2), whereby it delegates all PPP processing function to a PPP server in the backend network.

3.3 Media Access Control

The simplest solution for enabling networking over BlueSky is to model the wireless link as a replacement for an RS-232 cable and then to run PPP between the PDA and the access point. Providing this functionality is relatively straight forward and several commercial wireless RS-232 cable solutions (such as Ricochet) already support this. However, this solution only works as long as only one PDA is connected to the access point. For providing multi-point communication capability, a MAC protocol is needed for time sharing the radio link between multiple devices. The BlueSky radio module provides a full duplex link where both uplink (PDA to AP) and downlink (AP to PDA) use different channels (or frequency bands) for transmission. All PDAs listen on the downlink frequency while the AP listens on the uplink frequency. To provide multi-acess capability, it is thus necessary to build a protocol for coordinating transmission of different PDAs on the uplink frequency band. When a PDA is transmitting, other PDAs must turn their transmitters off. This is accomplished by adding a simple on-off switch to the cordless phone radio which in turn is controlled using the MAC protocol. The access point transmitter stays powered-on at all times.

The MAC protocol used in the BlueSky system [6] is a polling-based, asymmetric protocol. We do not make use of the full duplex capability of the medium since we wanted the same protocol to also work over time division duplex channels. Our decision to use a polling based MAC is a departure from the 802.11 style CSMA MAC. After considering the cost, complexity, and power consumption tradeoffs, we preferred using a TDMA style polling MAC over CSMA style MACs. First and foremost, a software implementation is not attractive in order to get acceptable performance in CSMA, and secondly 802.11 network controllers are neither cost effective nor designed for low-power operation. Since minimizing cost and complexity of a PDA device was our primary goal, a polling based MAC was preferable as it allowed us to easily move complexity away from the PDA into to the access point, thus providing an extremely low cost, power efficient design point for the wireless adapter.

A BlueSky access point acts as an arbitrator of the wireless medium. It is responsible for sharing bandwidth among PDAs, registering new devices into its cell, and also carrying out handoff processing. The portion of the MAC protocol that runs on the access point, is called AP-MAC. The corresponding portion of the MAC protocol on the mobile device is called RS-MAC (remote-station MAC). AP-MAC is in control of the medium; it can send a packet to any PDA any time. All PDAs process the packet header on the downlink and if the destination address matches the address of the receiving station, the packet is received and passed to the RS-MAC. AP-MAC periodically polls each registered PDA and the RS-MAC responds to the commands received from the AP-MAC.

To register with an access point, the PDA must wait until the AP-MAC issues an *invite-station* poll message. Since several PDAs may be waiting to hear this message, a slotted aloha-like protocol is used to arbitrate access to the medium among competing senders. During all other times, the PDA must hear a *poll-and-receive-payload* poll message before it can transmit a packet on the uplink. An access point transports payloads to PDAs using *send-payload* messages. AP-MAC also includes a scheduler component which schedules MAC-level events such as *send-payload*, *poll-and-receive-payload*, *invite-stations*, *register a station*. Within the framework of the BlueSky MAC, a number of scheduling algorithms such as round-robin, fair-queuing, power-conscious scheduling, etc. can be used. Design of RS-MAC is considerably simpler than AP-MAC requiring less memory, processing, and power. A simple stop-and-go ARQ scheme provides reliable transport of payloads over the wireless channel. In addion a sleep mode is also supported to conserve power during idle periods.

3.4 Serial line emulation over packet based MAC

The BlueSky MAC is packet based, i.e., it transports data in the unit of packets, not bytes. Nonetheless, its purpose is to transport a bit-stream received from a PDA's serial port to the access point. In order for the MAC to provide serial line interface, the following three functions should be supported:

- assembling the bit-stream into bytes and, subsequently, into packet
- in-sequence, reliable delivery of serial stream
- providing emulation of control signals, such as RTS, CTS, DTR, RI, etc.

Converting the bit-stream into bytes is accomplished (rather simply) using an RS-232 chip in the attachment. However, the problem of assembling bytes into packets is nontrivial because a serial byte-stream does not support any notion of packet boundary. A simple strategy would be to buffer the MTU worth of bytes before transmitting the packet, but this may cause blocking if the number of bytes to be transferred are less than the link MTU. The other alternative would be to start packet transmission as soon as UART's 16-byte receive buffer is drained, thus running the risk of generating too many small packets. This problem is not unique to this layer; it also occurs, for example, at the TCP layer. When the send buffer holds small amount of data (less than the path MTU), the TCP sender needs to decide whether to wait for more data or begin the transmission of the next packet. According to the Nagle algorithm [13, 18], only one packet is transmitted per round trip time. Unfortunately, a Nagle-like technique will not work at the MAC layer since the round trip time of small size packets is small compared to the time it takes to receive the MTU worth of data through the serial port. A potential solution would be a timeout based transmission allowing enough data to accumulate in the send buffer. A large timeout will ensure good link utilization, but will adversely affect the response of interactive sessions. On the other hand, a short timeout will improve response time at the cost of wasting link bandwidth. Thus, there is a tradeoff between optimizing link efficiency vs. minimizing delay and choosing a good value for the timeout can be difficult.

The second challenge is to provide reliable, in-sequence delivery of the byte-stream. This requires the MAC protocol to also perform error-detection, loss-detection, retransmission, duplicate detection, and re-sequencing. This is because bytes are normally not lost, reordered, or duplicated on serial lines. The service provided by the serial line emulation layer must be consistent with the behavior that the higher layers expect from the serial interface. The complexity of implementing these functions is comparable to implementing a TCP like protocol at the link layer (with window size = 1 and without congestion control features).

Finally, a serial line emulation is not complete until a method is designed to propagate control signals across the wireless link. This can be done by defining special control packets which are sent (reliably) over the link whenever a control signal is asserted by either side of the link. Reception of these control packets would cause state transition (on or off) at each end, causing an OS specific upcall or interrupt propagation to the higher layers. The protocol for serial line emulation is not symmetric since each end-point needs to interpret control signals differently depending on whether it is a DTE or a DCE device.

The complexity of providing serial line emulation is at odds with the goal of simplifying protocol design. The design of the serial line emulation can be simplified by eliminating those requirements which are not absolutely necessary. First, it is safe to assume that the protocol above the serial line is packet based and it uses a framing format which can be used to mark packet boundaries. For example, when the higher layer protocol is PPP, packets are encapsulated in an HDLC-like frame structure as specified in [16]. Each frame starts and ends with the character 0x7E and all occurrences of 0x7E inside the frame are byte stuffed. By scanning the start-of-frame and end-of-frame markers in the received byte stream, the MAC layer can buffer a complete frame and then transmit it as payload of the MAC packet. This will work in most cases except when the received frame size is bigger than the link MTU. As pointed out in [9], this can happen even if the negotiated MTU at the PPP layer is smaller or equal to the link MTU. Whenever a higher layer packet contains escape characters, byte-stuffing inserts extra characters into the frame, causing the frame size to double in the worst case. Consistent Overhead Byte Stuffing (COBS) [9] is a potential solution to this problem. COBS bounds the overhead of byte stuffing, ensuring that at most one extra byte is added for every 254 bytes of packet data. This allows the PPP MTU to be set within 0.4% of the link MTU without the danger of requiring fragmentation and reassembly.

We could not use COBS because COBS would have required changes to the PPP stack of the PDA device. Our present solution involves setting the PPP MTU equal to the link MTU and performing fragmentation and re-assembly wheneve a frame larger than the link MTU is received by the MAC layer. Since byte stuffing can produce a frame which is at most twice the link MTU, breaking the frame into two segments and using a simple ARQ with 1 bit sequence number is sufficient to deal with this problem. If the PPP software of the PDA can be modified, a solution simpler than COBS would be to eliminate RFC 1662 framing all together and run PPP directly over the MAC protocol. Considering that PPP is packet oriented and the MAC protocol is also packet oriented, the two intermediate layers of serialization and de-serialization can be removed without compromising any functionality. Eliminating serial line emulation not only improves link efficiency, but also simplifies protocol complexity.

Exploiting the frame structure of the higher layer protocol significantly simplifies the packetization logic at the MAC layer. The only limitation is that pure serial line applications (which use the link as a bit pipe) cannot be used. This is not a problem considering that a higher layer packet based protocol is always used for connecting PDAs to the network. Serial port applications that depend on control signals are always point-to-point in nature and thus cannot be used anyway when the PDA is connected to the access point.

Further design simplifications are possible when it is assumed that the higher layer protocol is PPP. PPP is designed to run over serial lines, but it does not depend on control signals except for detecting when the link is up or down. Eliminating control signal emulation only makes the MAC simple and does not affect the operation of the PPP protocol. The resulting design of the serial line emulation layer is very simple; it forwards one PPP frame at a time; occasionally runs a simple segmentation and reassembly logic when the received frame is larger than MTU. The final code size is small enough to fit on a PIC 17 microcontroller with 128K memory.

4 LAN connection using Dialup Networking

Figure 5 shows a PDA using the BlueSky wireless adapter connected to an access point. To the dialup networking stack, the BlueSky adapter appears like a modem. The PDA stack is configured such that no AT command strings are sent; instead PPP is started as soon as the connect button is pressed. A PPP session is always initiated by the PDA, but the other end of the PPP session need not always be terminated at the access point. In the following, we describe two alternatives for terminating PPP sessions (both are supported in the BlueSky system) and their relative benefits and limitations.

4.1 Terminating PPP at the access point

Using the simplified serial protocol described in section 3.4, PPP LCP (link control protocol) [17] packets are first sent to the access point. When the PPP session is terminated at the access point, the access point is responsible for carrying out LCP, followed by NCP (network control protocol) negotiation. Similar to any PPP server, the AP should be capable of supporting the following functions:

1. user authentication using PAP or CHAP [15]
2. acquiring new IP addresses and assigning them to the PPP client
3. proxy ARP for the IP address allocated to the PPP client
4. perform IP forwarding between ethernet and PPP link
5. manage multiple PPP sessions if more than one PDA is connected to the AP

Functions 1-4 are part of all PPP servers, but 5 requires emulating a separate virtual serial port for every device connected to the access point. Physically there is only one serial port at the access point which is connected to the BlueSky AP module (see Figure 5). The serial port driver reads MAC frames and makes the payload of those frames available via the corresponding virtual com port to the PPP server.

The approach of using the access point as a PPP server works only as long as users remain within the wireless coverage area (cell) of the access point. Each cell crossing involves tearing down the old PPP session and reestablishing a new PPP session, requiring conscious user intervention,

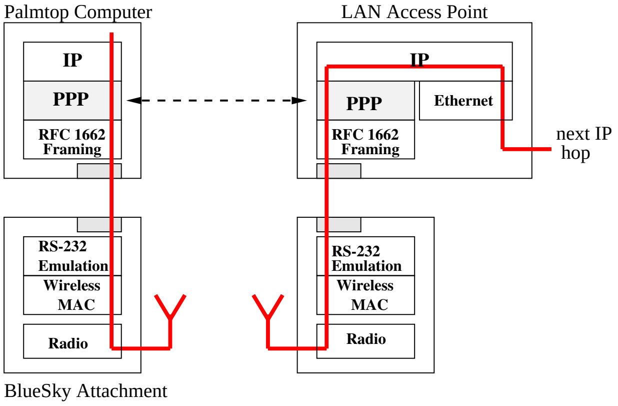

Figure 5: Terminating PPP at Access Point

such as connect button press event and restarting all applications. To enhance end-user experience, it is desirable to enable smooth roaming of palmtop devices throughout the coverage area. At the same time, it is also desirable to simplify management and configuration requirements of the solution. The design shown in Figure 5 does not provide roaming capability. Moreover, it requires each access point to participate in the authentication protocol. Since the BlueSky system uses a short range radio, a large number of access points will be needed to provide full wireless coverage in a typical office building, making managing and configuring tasks quite difficult.

4.2 Terminating PPP at the server: the BlueSky Approach

The BlueSky approach for mobility is to establish a PPP connection once and preserve it upon handoffs. This is possible provided the two ends of the PPP session remain fixed. Figure 6 shows a protocol architecture diagram enabling this feature. The PPP connection originating from the PDA devices is not terminated at the access point; instead a tunneling protocol is used to forward PPP frames to a PPP server. When the PDA moves, it carries PPP session state with it. After connecting to a new access point, it instructs the new AP to redirect the tunnel endpoint from the old AP to the new AP. The PPP server only needs to associate an already existing PPP session to the new tunnel endpoint, after which both PPP peers can resume sending packets to each other. Since the PPP connection is kept intact during handoff, the upper layers, including IP, are not affected. Conceptually, this can be viewed as a layer 2- mobility solution, since it makes the layer below PPP mobility aware.

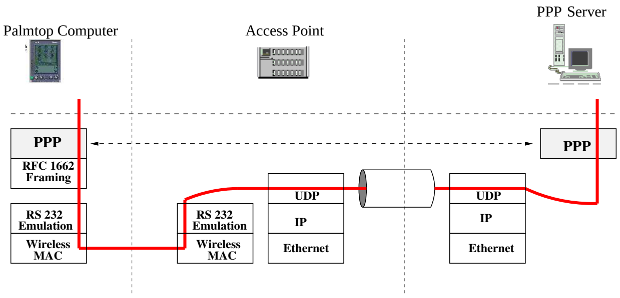

Figure 6: Terminating PPP at Backend Server

Why invent a new mobility solution when Mobile IP [14, 11] already exists? It is worth addressing this question before presenting technical details of the solution. Supporting PPP is an absolute requirement on all PDAs, since dialup networking using a modem or a cell phone are the most commonly used applications. Secondly, unlike laptops and notebook computers which come equipped with PC cards, PCI, ISA, and USB interfaces, serial port is the predominant and, quite often, the only communication interface available on PDAs. Since IP expects a packet interface, PPP cannot be dispensed with as long as the lower media is serial. Mobile IP can be used on top of PPP, but that won't solve the handoff latency problem at the PPP layer since each handoff will force the PDA to establish a new PPP session, requiring user intervention or causing long delays (our traces show that up to 17 messages are exchanged during the PPP session establishment!). Even if the limitations of serial communication are eliminated in the future, PPP may still be preferable due to its authentication, address allocation, compression, encryption, and billing features, all of which are critical to any mobile user.

The PPP based mobility solution enables seamless roaming using existing dialup networking software. It also offers an efficient, easy to manage design for the access point. In Figure 6, the AP only acts as a byte relay agent; it does not perform any PPP processing. The responsibility for authenticating the users, managing username/passwords, and allocating IP addresses lies with the PPP server which can be placed in a central location under the control of a site administrator. Conceptually, this approach is similar to Layer 2 Tunneling Protocol (L2TP) [19]; the access point is similar to an LAC (L2TP Access Concentrator) and the PPP server is similar to an LNS (L2TP Network Server). Similar to L2TP, our system also tunnels layer 2 packets from the access point to the PPP server. Support for seamless roaming is what distinguishes our work from L2TP. We also make several optimizations specific to the short-range wireless mobile environment which result in a design that is less complex and more efficient than L2TP.

4.2.1 Tunneling from AP to PPP server

An important aspect of the mobility solution is the protocol used for tunneling a byte-stream from the access point to the PPP server. The tunneling protocol is responsible for establishment, redirection, and clearance of tunnel sessions between the access point and the PPP server. There are two types of messages: control messages and data massages. Control messages are used to establish, maintain, redirect, and release tunnel sessions. Data messages are used to transport user data such as PPP frames.

Tunnel Transport As shown in Figure 6, the tunnel extends the wireless serial line up to the PPP server. The transport service provided by the tunneling layer is similar to the serial line emulation layer used above the wireless MAC protocol. Instead of treating the user data as a byte-stream, the tunnel transport mechanism scans for higher layer packet boundaries and forwards each higher layer packet (PPP frame) in a separate UDP packet. Since packet reordering may occur during transport, sequencing information should also be carried in each UDP packet. Loss recovery is optional. Instead of retransmitting lost packets over the tunnel, we rely on higher layer protocols to initiate loss recovery. The problem with adding reliability at the tunneling layer is that retransmissions may adversely interfere with the loss recovery mechanism used at higher layers. Aside from the complexity and overhead concerns, this was another reason why we did not choose TCP for tunneling. Providing reliability, nonetheless, is necessary for tunnel control traffic, since those messages are not sent by a reliable higher layer.

Efficiency of transport is an important consideration. Whenever a packet larger than the path MTU is sent over the tunnel, fragmentation occurs. To avoid this problem, the tunnel MTU is usually set equal to the path MTU - the tunnel header size, so that each higher layer packet will fit perfectly inside an IP datagram. Unfortunately, this trick does not work when PPP frames are transported over the tunnel. Even if the PPP MTU size is set equal to the (adjusted) tunnel MTU, packets bigger than the tunnel MTU may be generated due to byte stuffing. Consistent Overhead Byte Stuffing (COBS) [9] can again be used as a solution. In this case we avoid the problem by stripping RFC 1662 framing [16] at the access point. The output is a fixed (bounded) size PPP packet which can be transported over the tunnel without fragmentation.

Enabling Seamless Roaming Roaming is supported using tunnel redirection. For each device connected to the access point, a separate tunnel is established from the access point to the PPP server. Thus, there is a one-to-one relationship between a device and its associated tunnel. In the current implementation, for reasons of simplicity, we use the IEEE MAC address of the wireless adapter as the tunnel identifier. As soon as a PDA registers with the access point, a tunnel to the PPP server is established (see Figure 7). The PPP server records the association between the tunnel and the MAC address of the PDA device and waits to hear the first PPP LCP packet from the PDA. At this point PPP session messages can flow between the server and the PDA as if the two were connected directly over a circuit switched connection.

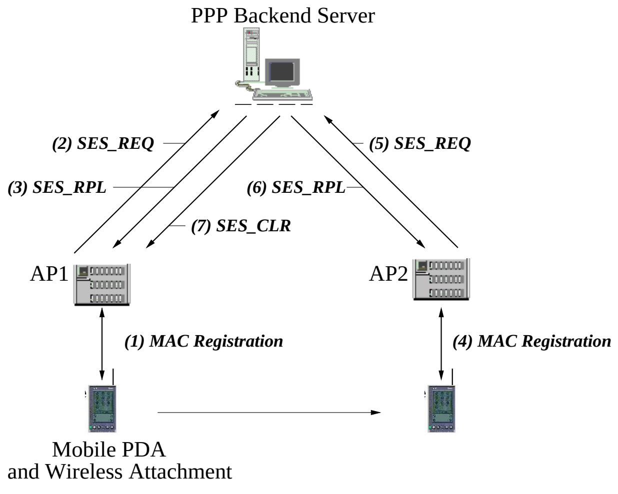

Figure 7: Tunnel Session Establishment and Redirection

When the PDA moves out of range of the access point and into the range of a second access point, a MAC level registration first occurs between the PDA and the second access point. The second access point automatically initiates the tunnel establishment protocol providing the MAC address of the PDA device. Since a tunnel session from the old access point to the PPP server already exists, the new tunnel establishment messages is interpreted as a request for tunnel redirection. As soon as the PPP server sends an acknowledgment for the redirection request, the PPP traffic between PDA and PPP server can resume.

There are several obvious ways of making the tunnel redirection process more secure. A potential solution could be for the PPP server and the PDA to negotiate a shared secret (tunnel id) at the time of a PPP session establishment. The shared secret can then be presented to the PPP server along with the request for tunnel redirection. As an additional security measure, after each handoff the PPP server can always challenge the PDA to verify its identity using CHAP (Challenge Handshake Authentication Protocol) [15].

4.3 Dialup Networking Stack Enhancements

To some extent the design of the BlueSky system is influenced by the requirement to interoperate with the existing handheld devices. More efficient solutions can be built if some changes can be made to the dialup networking stack of the palmtop devices. As we pointed out in section 3.4 and 4.2.1, transport inefficiency and protocol complexity stems primarily from the necessity of performing packet-to-serial and serial-to-packet conversions. Since the new generation of wireless links are packet-oriented, it will be more efficient to run PPP in packet mode (without the HDLC-framing). Figure 8 shows an example of the protocol stack with the suggested modification. Compare Figure 8 with Figure 6 to see the processing layers that have been eliminated. Other benefits include:

- elimination of redundant processing at palmtops
- no byte stuffing is needed which ensures that PPP packet size will be deterministic
- no fragmentation and reassembly logic is needed at the MAC layer
- similarly, the tunneling protocol does not have to deal with the problem of variable frame size, resulting in complexity reduction
- forwarding logic at the access point is simplified.

If the PPP sever is connected to the same LAN segment as the access point, then the access point design can be further simplified. If the negotiated PPP MTU is less than the ethernet MTU, the PPP packet can be forwarded directly over the ethernet (see Figure 8). This will result in a very simple design for a layer 2 access point. Such an access point will be a pure plug-n-play type and not require any administration, or configuration support. It is also our hope that any degree of complexity reduction will also translate into cost reduction for the equipment as well as the overall solution.

5 Discussion and Conclusion

In the very near future, palmtop devices will be equipped with a short range wireless link such as Bluetooth [1]. Providing support for legacy networking protocols and applications over those links will be a natural requirement. The purpose of the BlueSky project is to explore the wireless networking challenges that lie ahead of us. Instead of inventing new ways of doing networking, we limited our focus to adapting existing solutions to the needs of the future. Our decision to pursue a PPP based mobile networking solution is grounded on the belief that the future of mobile networking will be reached via the path of evolution, not revolution.

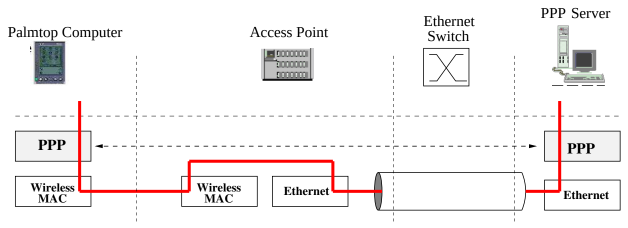

Figure 8: Potential Enhancements to Dialup Networking

Handheld devices can be connected to the network today. However, as these devices become commodity items, networking solutions will also need to focus on providing more functionality at lower cost. Unless mobile networking solutions also become power-conscious and cost-aware, the solutions of today will not scale to the networking needs of tomorrow. In this paper, we looked at the problem of providing dialup networking access via short distance RF links. During the course of our investigation and prototype building, we discovered several design simplifications which have the potential of providing the intended benefits. Our solution offers efficiency improvement and cost reduction over existing methods. The feasibility of the ideas presented in this paper have been tested on the BlueSky testbed, but we have yet to conduct detailed performance tests to quantify the improvements. Specifically, the issues related to handoff latency [7, 4], synchronization of tunnel state after packet loss events, etc. need to be investigated further.

Although we didn't explicitly address the issue of device to device and multi-point communication, our current prototype supports it using PPP. Either two devices can establish a peer-to-peer PPP session over the radio link, or they can both connect to the same access point using PPP. When more than two devices are present all can connect to the same access point using PPP and then communicate with each other. Support for multicast is desirable, but it has not been implemented yet.

References

[1] BlueTooth Special Interest Group. http://www.bluetooth.com.

[2] HomeRF Working Group. http://www.homerf.com.

[3] Richochet Wireless Network. http://www.metricom.com.

[4] H. Balakrishnan et al. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. In *Proceedings of ACM SIGCOMM'96*, 1996.

[5] P. Bhagwat, C. Bisdikian, I. Korpeoglu, A. Krishna, and M. Naghshineh. System Design Issues for Low-Power, Low-Cost Short Range Wireless Networking. In *IEEE International Conference on Personal Wireless Communications (ICPWC)*, February 1999.

[6] C. Bisdikian, P. Bhagwat, B. P. Gaucher, F. J. Janniello, I. Korpeoglu, M. Naghshineh, and P. Pandoh. WiSAP: A Wireless Personal Access Network for Handheld Computing Devices. *IEEE Personal Communications*, December 1998.

[7] R. Caceres and L. Iftode. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. *IEEE JSAC*, 13(5):850–857, June 1995.

[8] S. Cheshire and M. Baker. A Wireless Network in MosquitoNet. *IEEE Micro*, 16(1):44–52, February 1996.

[9] S. Cheshire and M. Baker. Consistent Overhead Byte Stuffing. In *Proceedings of ACM SIGCOMM'97*, pages 209–220, Cannes, France, 1997.

[10] R. Droms. Dynamic Host Configuration Protocol, March 1997. RFC 1231.

[11] D. B. Johnson and C. Perkins. Mobility Support in IPv6, November 1998. draft-ietf-mobileip-ipv6-07-.txt.

[12] R. O. LaMaire, A. Krishna, P. Bhagwat, and J. Panian. Wireless LANs and Mobile Networking: Standards and Future Directions. *IEEE Communication*, 34(8):86–94, August 1996.

[13] J. Nagle. Congestion Control in TCP/IP Networks, January 1984. RFC 896.

[14] C. Perkins. Mobile IP Specification. RFC 2002.

[15] W. Simpson. PPP Challenge Handshake Authenticaton Protocol, 1994. RFC 1994.

[16] W. Simpson. PPP in HDLC-like Framing, July 1994. RFC 1662.

[17] W. Simpson. The Point-to-Point Protocol, July 1994. RFC 1661.

[18] W. R. Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*, volume 1. Addison-Wesley, 1994.

[19] A. Valencie et al. Layer Two Tunneling Protocol (L2TP), October 1998. draft-ietf-ppext-l2tp-12.txt.