An Infrared Network for Mobile Computers

Norman Adams
Palo Alto Research Center
Xerox Corporation
norman@parc.xerox.com

Rich Gold
Palo Alto Research Center
Xerox Corporation
richgold@parc.xerox.com

Bill N. Schilit[*]
Computer Science Department
Columbia University
schilit@parc.xerox.com

Michael M. Tso
EECS Department
MIT
tso@lcs.mit.edu

Roy Want
Palo Alto Research Center
Xerox Corporation
want@parc.xerox.com

Abstract

The ParcTab infrared network provides a flexible infrastructure for research into wireless mobile computing. The network consists of a collection of room-sized cells each wired with a base station transceiver. Mobile computers communicate with transceivers through a carrier sense multiple access (CSMA) protocol and act as terminals for applications executing on remote hosts. Each mobile computer is represented by a proxy, or *agent*, accessible to applications at a fixed network address. In the ParcTab system it is the agent that is responsible for delivering requests to its corresponding mobile computer, and tracking the mobile as it moves from cell to cell.

1 Introduction

Personal digital assistants (PDAs) perform the function of paper organizers with the added benefits of digital control. Like paper organizers, PDAs fit in the palm of the hand and can be carried with you throughout the day. However, the true advantage of PDAs are realized by communication with networks of more powerful computers, enhancing the capabilities of these miniature machines.

We have built ParcTab and a supporting infrastructure. ParcTab is a PDA that communicates using infrared (IR) data-packets to a network of IR transceivers. The infrared network is designed for in-building use, where each room becomes a communication cell. In contrast to the approach used by other PDAs, most ParcTab applications run on remote hosts and therefore depend on reliable communication through the IR network. The infrastructure provides reliability as well as uninterrupted service when a ParcTab moves from cell to cell.

This paper focuses on the Unix-based infrared network we have built in support of the ParcTab. The first section provides a system summary. The next three sections describe the physical, data link, and network layers. We conclude with a performance assessment.

2 System Summary

The ParcTab IR network provides communication between stationary transceivers (*base stations*) and mobile systems[1]. Although the IR network was designed specifically for a hand held PDA called the ParcTab (hence the name of the network), the network is used with other portable computing devices. This section presents an overview of the ParcTab system components.

[1] This work was supported by Xerox. Portions were also paid for by ARPA under contract DABT63-91-C-0027.

[*] Currently visiting Xerox Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304

[1] Mobile to mobile communication is part of the design but has not yet been implemented.

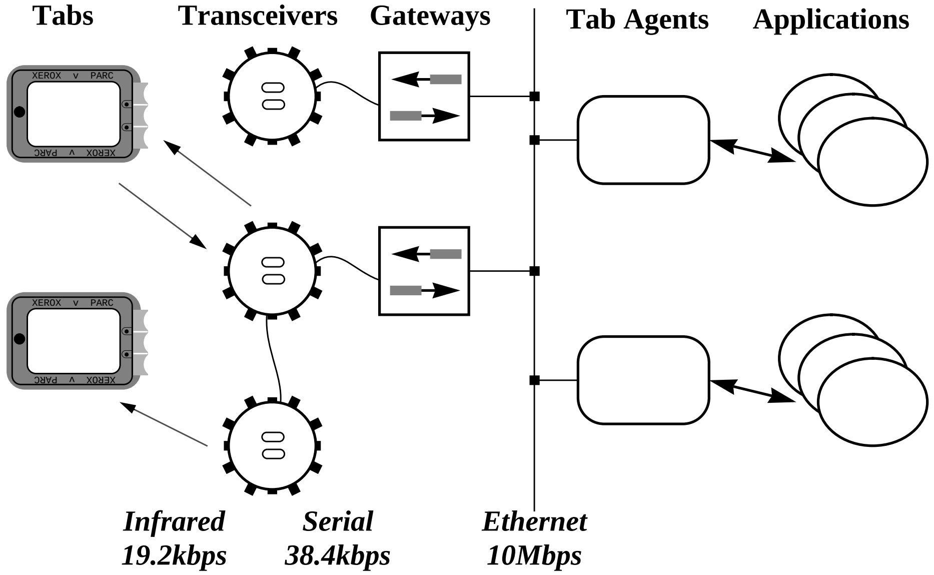

Figure 1: PARCTAB System Components

The system components are shown in Figure 1. Tabs and transceivers are custom hardware components. Gateways, agents and applications are Unix processes. The system employs three communication media: infrared is used for packets between transceivers and tabs; a serial line connects a host running an IR gateway and a base station transceiver; and an Ethernet connects IR gateway processes with agents. Sun RPC is used for communication between gateways, agents and applications. These components are described below.

The PARCTAB PDA is designed to fit in the palm of the hand and incorporates three buttons as part of the grip. Output is displayed on a touch sensitive 128 x 64 pixel resolution display. A number of IR diodes for transmission are spaced around the tab case, and a multi-directional receiver is built into the top of the unit. The tab functions more as a graphics terminal than a general purpose computer. However, the tab interface includes functions to download object code and data.

One or more base stations can be connected to a host workstation through a serial port. A device control protocol is used between the host and its connected base stations. In the case of multiple base stations sharing the serial line, the control protocol is synchronous: each base station must be polled by the host before it can send data. This configuration is useful for long runs with low expected utilization (e.g., hallways). For better efficiency, the host serial line can be dedicated to a single base station, and the base station operates asynchronously: the base station forwards IR packets to the host as soon as they are received.

Gateway processes on each host export a Sun RPC [2] interface for access to the base station. The RPC interface allows clients to send a packet unreliably over the infrared medium. When transceivers receive packets from the gateway they broadcast them into the infrared medium. The packets sent from the transceiver contain type, length, source address, destination address, payload, and checksum. Above this protocol layer is an at-most-once RPC employing sequence numbers and timeouts.

Our experimental network uses baseband modulated infrared to carry variable length data packets with a maximum size of 256 bytes. A packet broadcast into the infrared band is received by all transceivers within a room-sized “cell,” and is copied from the infrared medium by destinations which select it according to the packet’s address. To handle demands for growth, more transceivers can be installed on other workstations. Infrared transducers have the benefit that they are small, low power and low cost. Since infrared cannot penetrate opaque materials such as walls and doors, the network has good isolation properties.

This system provides a flexible infrastructure for research into mobile computing. We’re currently designing other mobile devices that will use the same gateway and transceiver network. Over a period of time, the system can be scaled

```
 0                                       1
 0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                              SYNC                             |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|        transceiver #          | R |      command code         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|    argument/reply length      |        arguments/results...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+-...

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                       command checksum                        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 2: IR Transceiver Command Format

to support many mobile computers, as the incremental cost of extending the network is small.

3 Network Transceiver Hardware

The IR transceiver contains electronics to send and receive signals in the IR medium, verify checksums, and buffer transmit and receive packets in a FIFO. The transceiver is also capable of sending and receiving packets at multiple data rates and generating data-link layer acknowledgment packets. The current version of the transceiver is capable of transmission over the infrared medium at 9.6kbps and 19.2kbps. Although this is slower than many radio-based networks, the small cell size permits a high aggregate bandwidth for multi-cell systems.

We have designed IR transceivers in four different packages. Both the PARCTAB and a pad-sized mobile computer contain built-in transceivers. The base station is an externally powered transceiver along with a serial-line interface. This unit fits into a 4-inch square clear styrene box, and can be attached to a ceiling, a bookshelf, or simply placed on a desk in rooms wired as cells. Finally, we are building a self-powered transceiver, the *velcro-station*, for use with a wider range of portable machines such as the HP-95 and Apple Powerbook.

The infrared communication capabilities of the PARCTAB network are somewhat different from a wire-based Ethernet [6]. Quantitatively, our IR network transmits at 19.2kbps, or about 500 times slower than a 10Mbps Ethernet. The infrared network is also more susceptible to environmental noise caused by, among other things, bright sunlight and fluorescent lights. Qualitative differences include the fact that packets might traverse multiple paths from the mobile host because of overlapping transceiver cells. Finally, the infrared transceiver is able to listen for a clear channel before sending, but because of hardware limitations, it is not possible to detect collisions once transmission has started.

The rest of this section describes transceiver functions. We first explain how the base station transceiver communicates using device commands over an RS-232 serial line. This is followed by a description of the infrared physical layer and media access layers which are implemented by a microprocessor within the transceiver.

3.1 Transceiver Device Control

The base station and self-powered velcro-station connect to a host through a serial-line interface. There are a number of reasons for this design instead of, for example, a bus interface. First, the serial-line transceiver can be used with a number of machines containing different bus architectures without modification. Indeed, some mobile systems do not have extension bus slots whereas they almost always have serial-line ports. Serial-line devices are low cost and easy to install. In the case of a base station they can also be configured in daisy chains. Finally, although serial interfaces are slower than DMA interfaces, this has not affected our current system because the IR interface is also quite slow.

Communication between the base station transceiver and host workstation is achieved by a command/response control protocol. Commands are sent to one of the transceivers on the serial line and the transceiver responds with a command result. The four main purposes of the control protocol are: to allow for multiple transceivers, to provide framing for outgoing IR packets, to detect errors, and to configure the operation of the transceiver.

The command format is shown in Figure 2. The first two bytes are a synchronization signal that uniquely identify the start of a command. Any `SYNC` bytes contained in the data are quoted. Following this is the transceiver number which selects a transceiver on a serial line (for multiple-transceiver configurations), the command code, arguments, and checksum. The checksum is necessary because we do not assume error-free communication over the serial line to the transceiver. The `R` field of the command code is set for all command responses from the transceiver.

The command format provides a well defined interface for transceiver-host interactions. The command set implemented by transceivers has evolved over time and consists of functions for debugging, configuration, and transmission. Debugging commands include setting a loopback mode and a mode that ignores checksums. Configuration commands permit device and FIFO reset, setting an address filter, and setting asynchronous mode so that incoming packets are immediately passed to the host. Finally, data transmission is accomplished with a command to broadcast the command argument over the IR medium. When asynchronous mode is disabled, infrared packets are read using a poll command.

3.2 Infrared Physical Layer

Infrared light emitting diodes (LEDs), similar to those used in the remote controls of consumer electronics, are the transducers used to implement the physical layer of the PARCTAB protocol. These diodes emit light at a wavelength of 850nm. In order to guarantee the IR emissions are diffuse and hence they will travel in all directions in a room, wide-angle LEDs are used. For reception, detector diodes in conjunction with a preamplifier demodulate the incoming signals.

Information is encoded onto the IR carrier by switching the carrier on and off in such a way that narrow pulses of IR radiation are emitted and the gap between the pulses carry the channel information. The modulation technique is called pulse position modulation. The emitted pulses are the delimiters for the encoded data and these have a constant duration of 4us. This signal encoding has the property that the average energy emitted is small. If a greater intensity of radiation is required to ensure a suitable signal range, the pulses can be emitted at a greater intensity and hence a greater instantaneous power. However, to compensate the average power can be kept constant by decreasing the duration of the transmitted pulses.

3.3 Infrared Media Access Layer

The protocols used to access the IR medium in the PARCTAB system are experimental and will continue to be developed and improved in future work. At present, the system uses a carrier sense multiple access (CSMA) protocol without collision detection or collision avoidance. Lost packets are handled by mechanisms implemented in the higher protocols layers. However, the protocol design allows for data link layer acknowledgment of packets through the use of an option in the packet header. The acknowledgment from the destination transceiver has a good chance of being received without collision because it is sent immediately upon packet receipt.

PARCTABs are used as terminals and therefore the average IR packet size for downlink communication tends to be much larger than the uplink packet size. Typically, uplink packets—from mobiles to room transceivers—carry key presses and pen events which are small. Downlinked packets carry display updates, such as bitmaps and text, and are comparatively large. All downlink packets sent to a communication cell are serialized by the cell's base station and therefore do not collide with each other. Packet collisions can be caused by the asynchronous button presses and pen events from the tab, or by base-stations in overlapping cells.

The communications cells used in this system are very small and are sometimes referred to as "nanocells." In other wireless mobile networks, variants of the CSMA/CA protocol, e.g., MACA [5], have been suggested to improve bandwidth utilization in the event of "hidden terminals". It is our belief that hidden terminals are not a significant problem between cells in a cellular system where the cells are well isolated. Also, in the PARCTAB network, hidden terminals within one cell do not tend to occur because the cells are small and the emitted power from one mobile can nearly always be sensed by another.

3.4 Link Layer Packet Format

The tab network packet format is shown in Figure 3. Our design goal was to provide sufficient flexibility so that encapsulating other protocols would be easy and efficient while providing the low level capabilities which we believe are important for mobile protocols. The `type` field is used to identify packet formats, allowing multiple IR protocols to coexist with this system. The `speed` field allows sending at multiple baud rates, currently either 9.6kbps and

```
 0                                       1
 0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|    type   | - | - |   speed   |            length               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                      destination address...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+-...
/                         source address...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                                                               /
\                            payload                            \
/                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IR packet checksum                        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 3: Link layer IR Packet Format

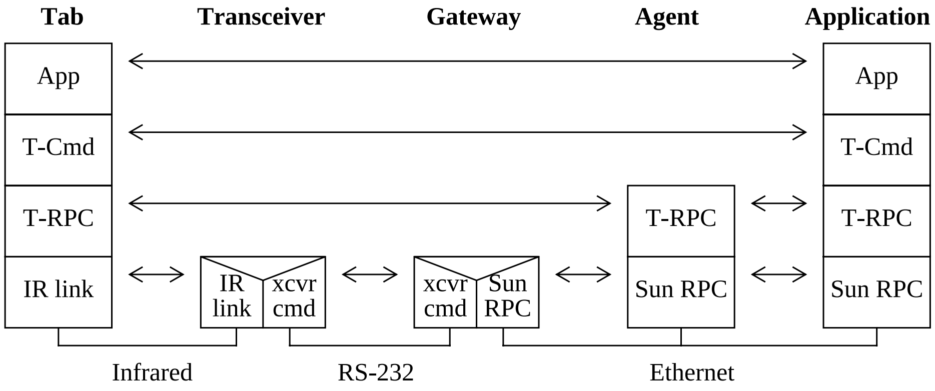

Figure 4: PARCTAB Network Layers

19.2kbps baud. The type byte is always sent at the lower speed. The checksum is computed using an algorithm [3] that is suited to the limited arithmetic capabilities of small microprocessors.

4 Network Layer Implementation

The functions of the network layer include routing and heterogeneous network operation. In broadcast networks, such as Ethernet, the routing of packets is quite simple. In the case of a mobile host network, routing can be more complicated.

The network layer is implemented by a collection of independent processes called *tab agents*. Each tab is represented by an agent which is the only component that must know how to route packets to the tab. The tab agent serves many of the same functions as Mobile Support Routers (MSRs) in the Columbia Mobile IP proposal [4]. However, instead of employing MSRs to handle routing for a group of cells, our network assigns agents to handle routing for specific mobile hosts.

IR packets are received by one or more transceivers and transferred to the cell's IR gateway. A packet's source address is used to forward the packet to the tab agent. In order to translate between an infrared tab address and an agent process's Internet address, a name service is used. IR gateways use an address cache to avoid referencing the name service on each packet.

When packets are sent in the opposite direction, from application to the tab, the location of the mobile host is required. Location information is maintained at the agent by the use of return addresses and beacons. When an IR packet arrives at the agent it contains a return address for the cell that generated the packet. Any packet destined for the tab is sent to the most recently received cell address. The tab agent caches these addresses. This approach is adequate as long as tabs generate packets when moving from one cell to another. Because users cannot be relied on to create events to update their location, the tab “beacons.” The beacon is a tab-generated event that is sent periodically whenever the tab is idle. Like all tab events, the beacon includes a return address which the agent uses to update the cell address for the tab.

5 Application-Tab Communication

This section describes communication between the tab and applications running as Unix processes on remote workstations (see figure 4). There are two styles of communication in our system: asynchronous (one way) events from the tab, which are unreliable; and at-most-once RPCs between the applications and the tab, which are reliable. These are described below and illustrated by an example.

5.1 Application-To-Tab RPC

PARCTAB applications use a command protocol (T-Cmd), see figure 4, for controlling tabs. Each T-Cmd operation corresponds to a particular tab capability such as displaying bitmaps or generating a tone. T-Cmds are converted to a packet format suitable for transmission to a tab using T-RPC. A round trip T-RPC involves Ethernet, serial line, and IR net transmissions, and provides reliable packet delivery to a mobile host. T-RPC addresses two problems: lost packets, caused by collisions and transmission errors; and movement of tabs between cells.

The T-RPC is similar to the approach described in [1]. A T-RPC request is sent to the tab (actually broadcast from the cell transceiver where the tab was last sighted). If no response appears within a certain delay, then the packet is resent. If a packet is seen by another cell’s transceiver, then the T-RPC request is immediately resent to the new location. Sequence numbers are used to implement an at-most-once RPC mechanism. Tabs receiving a duplicate request respond with the previous reply. In the case that the tab receives a T-RPC request while it is still executing the previous request, it will send a reply indicating it is busy. The T-RPC implementation backs-off the retransmission of requests when the tab is missing or busy.

Reliability at the T-RPC level instead of lower levels of the tab protocol has simplified our system design. Our original design proposed that the gateway (MAC layer) would retransmit packets for some period, and if that failed, then the tab agent (network layer) would retransmit to a different gateway. Event though an extra Ethernet RPC is necessary in the retransmit loop when retransmission occurs at the agent, our current design is effective because Ethernet communication has much lower latency than tab infrared communication.

5.2 Tab-To-Application Events

The tab transmits button and pen events as IR packets. These events are not acknowledged by the receiver and are therefore unreliable. It is left to the user to retry a button or pen action until a response is seen.

The agent can receive duplicate events when multiple transceivers receive a tab event and each forwards a copy to the host. The agent filters duplicate events and discards old events rather than delivering them out of order. Sequence numbers are used for both of these purposes. An event is ignored unless its sequence number is greater than the sequence number of the previously processed event.

5.3 Example

One PARCTAB application draws a piano keyboard and accepts pen events that cause tones to be played. The low-level communication involved in this process is described below.

- The application generates a bitmap draw request for the tab to execute. The application sends a T-RPC request to the tab agent and then waits until the agent responds with a reply from the tab before continuing.
- The agent receiving the application T-RPC sends a packet to the IR gateway where the tab was last sighted. Since the agent is responsible for reliable delivery, it will resend until a valid T-RPC response is received.

- The IR gateway receiving the packet encapsulates it in a transceiver command and transfers it over the serial line to the base station, which in turn broadcasts the packet over the infrared medium.
- When the PARCTAB receives the T-RPC request, it draws the bitmap and generates a T-RPC response which it transmits over the infrared medium. Any base stations receiving the IR packet transfer it over the serial line to the cell's IR gateway.
- The IR gateway upon receipt of an IR packet from the serial line strips off packets headers and forwards the packet to the tab agent.
- The tab agent receives the packet containing the T-RPC response and matches it to the pending T-RPC request. This causes the T-RPC to return to the application.
- The T-RPC containing the bitmap draw had the effect of displaying a button on the tab display. The user tapping the touch-sensitive screen generates a tab pen event that is broadcast over the IR medium. Transceivers receiving the event write it to the serial line for processing by the IR gateway. The IR gateway forwards the event to the tab agent, and the tab agent forwards the event to the application.
- When the application receives a pen event it sends a T-RPC packet containing a tone command making the tab play a note in response to the keyboard button being pressed.

6 Performance & Analysis

The distributed infrastructure supporting the ParcTab network is built from a collection of Sun Microsystems SPARC 2 workstations[2] running SunOS 4.1.3. and are connected by a 10Mbps Ethernet.

6.1 System Measurement Methodology

To monitor the behavior and performance of the system we use three sources of information. First, programs include a large number of trace statements. A trace statement is a print statement that executes only when its controlling trace variable is set. The programmer defines the set of trace variables for a program, and associates a trace variable with each trace statement. The trace variables in a program are collected together and can be viewed and modified during the program's execution. Each program also exports a network accessible command interpreter that the implementor can use to change parameters referenced by the program during execution. Second, we used a kernel events tracing package. By inserting `vtrace` system calls into programs, and by running the IR gateway, agent and application on the same machine, we are able to get a precise time line of events across all the Unix processes in this system. Third, we used an oscilloscope to determine the timing of relevant hardware events:

- serial-line transmission of T-RPC request from host to transceiver
- IR transmission of T-RPC request from transceiver
- IR transmission of T-RPC reply from tab
- serial-line transmission of T-RPC reply from transceiver to host

Using this information we removed a number of performance bottlenecks in the system. We found that Sun's serial-line driver imposes a 20ms delay between delivering consecutive buffers of data received on the serial line. This delay was removed by writing a loadable device driver. In addition, trace statements were expensive even when not selected to print, because the parameters were always evaluated. Some of these parameters invoked procedures for the textual formatting of large arrays of data, and hence the delay.

When the original measurements were performed, each tab packet contained only a single tab function. We have since implemented the grouping of multiple tab functions into a single command packet[3], thus increasing the average size of packets and improving overall throughput.

[2] A Sparc 2 workstation typically has an instruction rate of 20 MIPS.

[3] Grouping works for functions that do not return a value other than a success or failure. The tab will process a list of functions contained in a command buffer in order until all of them have been executed or an error occurs.

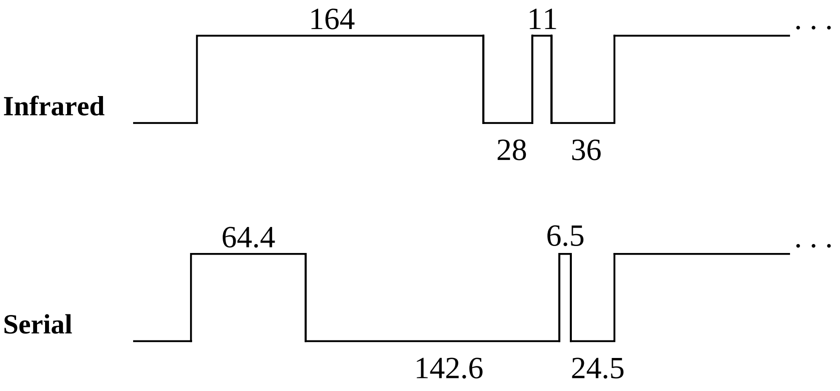

Figure 5: Infrared and Serial Packet Transmission Times (ms)

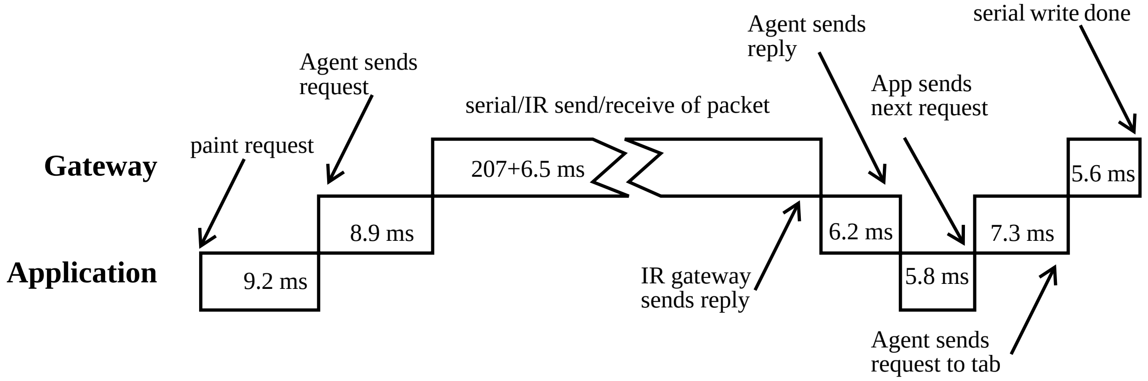

Figure 6: Unix Process Event Trace

6.2 System Timing

Figure 5 charts the infrared and serial-line activity in the course of transmitting a 240-byte packet to the tab. The sequence starts with a 64.4ms transmission on the serial line of the packet plus 7 bytes of transceiver command overhead. The transceiver begins broadcasting bytes over the IR medium 4ms after it begins receiving the packet. The IR transmission time is 164ms since the infrared data rate is 19.2kbps versus 38.4kbps for the serial line. The 28ms idle period following the IR transmit is taken up with tab execution time for the T-RPC request contained in the packet. Following this event, the IR transmission of the T-RPC reply, a 14-byte packet, takes 11ms. The transceiver transfers the 14-byte packet plus 7 bytes of transceiver command overhead on the serial line in 6.5ms. There is no overlap between serial transmission and IR transmission in this case because the packet checksum is verified before forwarding it over the serial link. The remaining 24.5ms before the next transmission is due to communication between the IR gateway, agent, and the application as shown in Figure 6.

7 Conclusions

The general approach used by the PARCTAB system has proven to be beneficial from many standpoints. The time and effort required to install a communication cell in a building where networked workstations are already in place is small. Typically we find an IR transceiver can be installed in about 15 minutes. By making use of existing workstation serial ports, the network-management costs are also low. Moreover, because the project could enforce its own policies at a level higher than the Unix system infrastructure, it was not necessary to involve the system administrators in setting up this experimental system.

Another success of the project was in the choice of general protocols for communication across the serial line and through the IR medium. The serial-line protocol allowed us to transfer IR packets from a workstation to a transceiver. Once in place this code remained static through most of the ParcTab system development. The same is true for the IR-packet format. The data payload contained in these packets did however change to suit the functionality of the tab as we experimented with different UIs and data compression techniques. The interfaces chosen for separating the transceiver and the IR packet delivery system have stood up to many architectural changes without modification. In fact, the transceivers could deliver the packet structure of other protocols such as IP or ATM without change. We conclude the interfaces for these protocols were well chosen for this research vehicle.

In the initial design phase of the system there was concern that building an architecture to support the PARCTAB distributed among many Unix processes, on two or more machines, would lead to unacceptable latencies when implementing interactive mobile-applications. In practice, for our design this has not been the case. Most round trip delays are under 250ms, hence comparable with human reaction times, and the system feels responsive in its use. This is particularly important so that the PARCTAB gives an impression the computing power is palm of the user's hand.

A competitive design for the PARCTAB network would be to implement Mobile IP [4]. This approach conflicts with the goal of designing a small, portable device and a low-cost infrastructure to support it. Mobile IP would require a tab resident implementation of IP, and the transceivers would become IP routers. These system requirements are beyond the capabilities of both the current base station hardware and the PARCTAB.

8 Future Work

There are still many improvements that would enhance the tab system. The flexibility designed into our own system has not been fully explored, for instance, enabling link layer acknowledgment for tab event-packets will change the characteristics of the system bringing improvement in reliability, but at the same time, a possible reduction in packet throughput. The media access protocol in use could be replaced by other protocols, such as MACA [5], designed for the mobile environment. Many of these questions are not easy to answer by simulation simply because the usage model is not known. The existing PARCTAB system allows us to gain understanding of the usage model and to plan for the future.

Currently, many companies are on the verge of making communication products that could significantly improve the bandwidth provided by the IR transceivers used in the experimental tab system. A 1Mbps product suitable for use with laptops should be available by the end of this year. Upgrading the system's IR physical-layer will not require changes in the packet delivery mechanism provided by the rest of the architecture. Incremental changes like this ensure a flexible development path and the longevity of the system.

Mobile computing provides many new opportunities for us to develop novel applications. The most important new dimension these devices bring is **location** and hence context information. Location tells the system where you are and context tells it who you are with. The authors have already had some experience with another device, the active badge [7] which provides location information about people who wear it in a suitably wired building. There are many applications that could modify their behavior based on location. Programs can be designed to automatically make decisions governed by context rules that have been specified by a user. The investigation of this new environment is part of on-going work and will be the subject of future papers.

In summary, the PARCTAB system is novel and open for experimentation. It allows research into the field of mobile computing and also allows investigation of new applications in advance of commercial developments in this area.

References

[1] A.D. Birrel and B.J. Nelson. Implementing remote procedure calls. *ACM Trans. On Computer Systems*, 2:39–59, Feb 1984.

[2] John R. Corbin. *The art of distributed applications : programming techniques for remote procedure call*. Springer-Verlag, New York, 1991.

[3] John G. Fletcher. An arithmetic checksum for serial transmissions. *IEEE Transactions on Communications*, COM-30:247–251, Jan 1982.

[4] J. Ioannidis, D. Duchamp, and Jr. and S. Deering G. Q. Maguire. Protocols for supporting mobile ip hosts. Technical Report RFC xxxx, Mobile Hosts Working Group, June 1992. *Draft RFC available by anonymous ftp from site parcftp.xerox.com:/pub/mobile-ip/columbia-draft-june-92*.

[5] Phil Karn. MACA - a new channel access method for packet radio. In *Proceedings, 9th Computer Networking Conference*, pages 134–141. ARRL/CRRL Amateur Radio, Sept 1990.

[6] Robert M. Metcalfe and David R. Boggs. Ethernet: distributed packet switching for local computer networks. *Communications of the ACM*, 19(7), Dec 1976.

[7] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, Jan 1992.