The PARCTAB Mobile Computing System

Bill N. Schilit* Norman Adams

Rich Gold Michael M. Tso† Roy Want

Palo Alto Research Center
Xerox Corporation
3333 Coyote Hill Road
Palo Alto, CA 94304
{schilit,norman,gold,want}@parc.xerox.com
tso@lcs.mit.edu

1 Introduction

The PARCTAB is a personal digital assistant (PDA) that communicates via 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. The PARCTAB and a supporting infrastructure [1, 4] has been operational since March 1993 at the Computer Science Lab at Xerox PARC. The system currently comprises thirty cells and twenty-five PARCTABs and will expand in the near future.

There are three types of software components in the PARCTAB system: gateways, agents, and applications. Gateways implement a datagram service for sending and receiving packets using IR signals. Each tab is represented by an agent. An agent tracks the location of its tab and provides location independent reliable remote procedure calls. The protocols enforce security, preventing, for example, an unauthorized application from taking control of a tab. Applications are built using a library of widgets designed to accommodate the PARCTAB's low IR-communication bandwidth and small display area. A distinguished application, the "shell", permits a tab user to start and switch among applications.

2 System Summary

We see wireless communication as an important aspect of mobile computing. Communication allows users to ubiquitously connect to the information infrastructure being built from existing global networks. Because of this belief we have emphasized communication over local processing in the design of the PARCTAB system.

The PARCTAB runs a simple operating system. The tab functions more as a terminal than a general purpose computer; though it does permit object code and data downloading. The advantage of a terminal-based system is the simpler design, less costly production, and smaller size. The disadvantage is its reliance on a functioning network.

The PARCTAB IR network is a system for communication between stationary transceivers (*base stations*) and mobile systems as well as between two or more mobile systems. The IR network was designed specifically for the PARCTAB, though we also use the network with other portable computing devices.

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

* Columbia University, Department of Computer Science. Currently visiting Xerox Palo Alto Research Center.

† EECS Department, Massachusetts Institute of Technology.

Figure 1: PARCTAB System Components

The system components are shown in Figure 1. Tabs and transceivers are custom hardware. Gateways, agents and applications are software. The system employs three communication media: infrared is used for packets between transceivers and tabs; a serial line connects a host running the IR Gateway and the hardware transceiver; and Ethernet connects IR gateway programs with tab agents. Sun RPC is used for communication between gateways, agents and applications.

This system provides a flexible infrastructure for research into mobile computing. We're currently building other mobile devices that will use the same gateway and transceiver network. The system can scale to large buildings full of mobile computers and the incremental cost of extending the network is small.

3 Tab System Hardware

The PARCTAB hardware has been designed for use within an office environment. One objective was to create a small and ergonomically designed device that could be carried throughout the work day. This goal influenced the pragmatic aspects of the design: weight, size, power consumption, and the use of standard components.

The PARCTAB fits into 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 piezoelectric tone generator provides audio feedback, and an I^2C bus connector can be used for expansion. Power is supplied to the PARCTAB by rechargeable nickel cadmium batteries with 12 hours of continuous use between charges. In practice the time between recharges is much longer, approximately a week given the expected usage. To ensure the device turns off when not in use, the system enters power saving mode after a brief period of inactivity.

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 small room sized "cell" and is copied from the infrared medium by destinations which select it according to the packet's address. To cover a wider area, transceivers can be installed on additional 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. The current maximum data rate is 19200 bits per second. Although this is slower than many radio based networks, the small cell size permits a higher aggregate bandwidth for multi-cell systems.

IR transceivers, whether stationary or mobile, contain electronics to send and receive IR signals, verify checksums, buffer transmit and receive packets. The transceiver is also capable of sending and receiving packets at multiple data rates and generating link-level acknowledgment packets. Transceivers listen before broadcasting infrared and do not send while the channel is busy. Transceivers cannot detect collisions during transmission. The packets sent from the transceiver contain type, length, source address, destination address, payload, and checksum.

One or more base station transceivers are connected to a host workstation using the serial port. The host controls the base station by sending commands over the serial line. The base station forwards IR packets to the host as soon as they are received. Multiple base stations can be daisy-chained on a single serial line. In this case the host must poll base stations before they can send data. A daisy-chained configuration is useful for long runs with low expected utilization (e.g. hallways), but is less efficient than a configuration with a single line per base station.

Gateway software on each host exports an Ethernet-based RPC interface for communicating with the base station. The gateway provides an unreliable packet delivery service similar to UDP. Above this protocol level is an at-most-once RPC employing sequence numbers and timeouts.

4 Tab Software

Communication between an application running on a host and the tab always goes through the tab's agent. The agent handles packet retry and switches between different transceivers as the tab moves. The agent also moderates tab sharing by allowing only one application to communicate with the tab at a time.

There are two styles of communication in our system. Events from the tab are asynchronous (one way), unreliable, and a sequence number is used for filtering multi-path duplicates at the agent[1]. In the second style, RPC from applications are reliable (modulo Unix), and a sequence number for filtering retransmission-duplicates at the tab, and match responses to requests at the agent. The agent employs a non-terminating back-off-retry technique so that intermittently connected tabs always receive the request.

As an example of the communication steps involved, consider the situation of an application making the tab beep. The application sends the *beep* command to the agent which verifies the sender and forwards the request to the IR gateway where the tab was last seen. If there is no response the agent retries the send after a back-off delay. Any incoming tab packets from a different transceiver indicate that the tab has moved, and reset the target IR gateway. The IR gateway receives packets from the agent and forwards them over the serial line to a transciever. The transceiver in turn broadcasts the packets over the IR medium.

When the packet is received by the tab, it is filtered by its address field. The tab verifies the checksum and the packet sequence number. The functions in the packet are executed and a response containing a status code is generated.

Communication from the tab to the application starts with the user hitting a button or touching the screen. The tab broadcasts these events using IR. Zero or more base station transceivers receive the packet, and sends it over a serial line to an IR gateway. The IR gateway forwards the packet to the tab agent which, after filtering duplicates, forwards the packet to the foreground application.

5 Application Interface

The tab has built-in character and icon font sets that are used by text drawing functions. Another group of functions sends compressed bitmaps to the tab display. Additional functions permit reading and writing external memory, generating tones, and setting parameters for UI processing, such as local painting of pen marks.

The user library for programming tab applications provides the basic tab functions described above plus some higher level support. The library manages binding between applications and the tab agent at startup, batching tab functions into a single packet, reading and drawing Unix bitmap format files. In addition the library hides the details of Unix RPC.

The system supports multiple applications running at once, although only one may communicate with the tab at a time. In addition to pen and button events from the tab, applications may receive suspend, resume, and quit events sent from the agent. The tab agent initially starts a distinguished application, the shell, which is given a capability to connect other applications to the tab. The shell starts all applications and controls which is the foreground application, much like job control under a Unix shell.

[1] We plan on experimenting with link-level acknowledgment to provide reliable events

6 Tab Applications

We have built a number of PARCTAB applications to explore problems in ubiquitous information access, user interfaces for small pen computers, environment control, and application migration. This section describes a few representative applications that have been implemented.

- *Birddog*. A reverse pager application that reports peoples location from active badge sightings [3].
- *Dictionary & Thesaurus*. Allows accessing a large dictionary and thesaurus. It would be difficult to store this amount of information onto a device the size of the tab.
- *Responsive environment control*. The tab can be used as a general purpose, programmable remote control. In one research project the tab is used to control lights and temperature for the current location. [2].
- *Calendar*. The calendar application provides a tab interface to Sun's calendar manager program. A user may view and modify appointments while away from their workstation.
- *Weather*. The weather application is an example of the tab accessing time dependent information. The program displays the current weather conditions as reported by a rooftop weather station as well as the area forecast obtained from an Internet host.
- *Migration Control*. The tab can be used to control window migration. Users can select from a list of X displays and cause windows from a remote display to be migrated to a nearby display.

7 Future Work

We have built the PARCTAB system to enable research in ubiquitous computing. In the future we will be building applications to explore areas such as:

- Interactive information services.
- Messaging services.
- Group collaboration.
- Data preloading for disconnected operation.
- Adaptive mobile applications.

References

[1] Norman Adams, Rich Gold, Bill N. Schilit, Michael Tso, and Roy Want. An infrared network for mobile computers. In *Proceedings USENIX Symposium on Mobile & Location-independent Computing*, pages 41–52. USENIX Association, August 1993.

[2] Scott Elrod, Gene Hall, Rick Costanza, Michael Dixon, and Jim des Rivieres. Responsive office environments. *CACM*, 36(7):84–85, July 1993. In Special Issue, Computer-Augmented Environments.

[3] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 1992.

[4] Mark Weiser. Some computer science issues in ubiquitous computing. *CACM*, 36(7):74–83, July 1993. In Special Issue, Computer-Augmented Environments.