

Metronaut: A Wearable Computer with Sensing and Global Communication Capabilities

Asim Smailagic, Richard Martin
Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, PA 15213
asim@cs.cmu.edu, martin+@cs.cmu.edu

Abstract

Metronaut is a novel wearable computer which captures information, senses position, provides wide range communications, consumes less than one watt of power, and weighs less than one pound. Metronaut employs a bar code reader for information input and position location, a two-way pager for communications, and an ARM processor for computation. Metronaut's application is schedule negotiation and guidance instructions for a visitor to the CMU campus. The visitor's position is determined from reading bar codes at information signs around campus. Modifications to the schedule are negotiated using the two-way pager for communications with the campus computing infrastructure. Metronaut is alternatively powered by a mechanical flywheel converting kinetic energy to electrical energy.

Keywords: wearable computers, mobile computing, sensing, two-way pager, bar code, navigation.

1 Introduction

The Metronaut mobile computer exemplifies some of our early vision [1], [2] in being able to sense information and locate position via bar code reader, provide wide range communications via two-way pager, supporting applications like navigation, messaging, and scheduling and still consuming less than one watt of power and weighing less than one pound.

Metronaut's application is schedule negotiation and guidance instructions for a visitor to the CMU campus. The visitor's position is determined from reading bar

codes at information signs around campus. Modifications to the schedule are negotiated using the two-way pager for communications with the campus computing infrastructure. Metronaut is alternatively powered by a mechanical flywheel converting kinetic energy to electrical energy.

2 Application

The initial applications of the mobile unit include navigation, personal scheduling, and collaboration between units and a central information server. The unit features simple protocols for data entry, data retrieval, and robust communications. The wide range communications link and the bar code reader for input, supports a great number of other applications as well.

Several applications employing a bar code reader, such as inventory assessment and package identification already exist today, but only as parts of niche systems, usually with wireless communication that is very local in scope. In contrast, Metronaut has a very wide-area communications system (the pager), and is a general purpose device potentially able to exploit a large number of applications. Three applications are described that demonstrate the breadth and capabilities of the system: scheduling, navigation, and messaging. The following scenario illustrates the combine use of these three applications.

A visitor to the CMU campus can follow a busy schedule of meetings, and also make updates as needed. There is a set of bar code stickers (Stickernet) around the campus that contains encoded location information. The visitor can register their position upon arrival on

campus by scanning a bar code on a nearby positioning sticker. Metronaut then automatically provides directions to the first meeting location via instructions displayed on its LCD screen. The visitor can verify along the way that they are passing correct landmarks by scanning in additional bar codes. An intelligent agent negotiates with other people's schedules and makes updates by sending and receiving e-mail messages via the two-way pager. Also he or she can update their schedule with an upcoming event advertised by scanning in a bar code at the bottom of a flier announcing the event.

2.1 Scheduling

The scheduling application:

- Displays user's schedule
- Schedules individual and group events
- Provides convenient entry of schedule events

In addition to managing and displaying a user schedule, this application is able to collaboratively schedule an event with a group of people. Scheduling decisions and agreements can be made by freely "roaming" people without having to explicitly contact each other through other communication mechanisms. Even more interesting is the ability for users to quickly schedule events they see advertised around them without having to manually record the information and transfer it to the scheduling program later. Instead, they simply scan in the bar code for an advertised event which effortlessly and correctly enters it into their schedule.

2.2 Navigation

The navigation application:

- Displays user's location
- Shows directions to new locations

To support this application, technologies such as GPS (Global Positioning System) and RF transmitters and receivers were considered. However, the high power consumption of GPS (about 500mW), coupled with inadequate precision made it inappropriate for the application. In contrast, an infrastructure of bar code labels, called "Stickernet" identifying major locations and navigational points is extremely low cost, low power, and high precision, with virtually no degradation of the actual navigation application. The only difficulty introduced is that the user can only identify their location at pre-barcode points, but since coding density

can be increased at little cost and personal navigation naturally references a series of particular locations (rather than being more free-form, like idealized straight-line flying), this detail has little impact on the navigation application.

2.3 Messaging

The messaging application consists of reading and sending e-mail.

3 System Architecture

The overall system architecture consists of four functional components:

- The Metronaut
- The ground-based networked server system
- The SkyTel paging network
- The SkyTel network interface

The server side is split into two components, the SkyTel Network Interface and the actual Server. There are several reasons why this arrangement is advantageous, even though it appears more complicated, Figure 1. This arrangement makes the system independent of the communication system employed by the client. Currently, the SkyTel paging network with a Motorola Tango two-way pager are used, but were that to change in the future (perhaps to CDPD), the Server would not have to change much. Also, an independent server can be useful to other types of clients, such as non-roaming desktop machines with a network connection (i.e. you can modify your schedule from your computer at the office which has high network bandwidth, without having to suffer the battery life, bandwidth, and latency limitations of your mobile device).

Communication proceeds from the hand-held device via the pager to the SkyTel™ paging network, to the system server and back. When users request changes to their schedule, the Metronaut uses its built-in Motorola Tango™ pager to send a request through the SkyTel paging network to the SkyTel Network Interface, software and hardware that is able to receive and send messages through the paging network. The SkyTel Network Interface will then forward requests to the actual server, which is on the Internet. The server will process requests, access its databases, and send replies back to the Metronaut. Figure 2 illustrates the integration of

Metronaut, two-way pager, the server, and the interface design overlaid upon the SkyTel™ infrastructure.

3.1 Client Software

The client software resides on the Metronaut and includes the Nucleus Real-Time Microkernel, Device Drivers, GUI, and Applications. The operating system is Accelerated Technology's Nucleus PLUS, which is a real-time multi-tasking executive available for the ARM 7500 processor has a small memory footprint (<40KB). The following software components were implemented:

- Device Drivers: LCD display, Touch screen, Two-way pager, Bar code scanner
- GUI: functions to Draw pixels, Draw lines, Draw ASCII characters and strings, Read Touch screen position
- Applications: Scheduling, Navigation, Messaging

3.2 Server Software

The Web server utilizes the Common Gateway Interface (CGI) to activate a set of Perl scripts for the functional layers. Perl is often used with Web servers and there are extensions to interface with mSQL database engine that is used. The following databases and database managers were developed: location database, schedule database, navigation manager, and schedule database manager. The location database is organized by location codes and contains location data types and their relationship. The navigation database is made up of textual directions on how to get to a specific place. The schedule database is organized by user and holds a time-ordered series of events. The messaging manager provides a standard e-mail account on the server and standard e-mail messaging functionality for a roaming user.

After receiving a request from the Metronaut client through the SkyTel Network Interface, the server will run the appropriate script to process the request. This script accesses: the location database to retrieve information on locations and navigation paths, the schedule database to add and remove events, and the users standard electronic mailbox to retrieve and send electronic mail.

3.3 Hardware Architecture

Since low power consumption was a major design requirement, the ARM 7500 processor emerged as the best candidate from the Technology Survey. The ARM 7500 runs at 33 MHz and consumes 430 mW. It is an integrated CPU based on the ARM7 processor core. This core is part of the Advanced RISC Machines (ARM) family of general-purpose 32-bit microprocessors. The VLSI ARM 7500 is a fully static CMOS implementation of the core, allowing the clock to be stopped in any part of the cycle, with extremely low residual power consumption and no loss of state. The core is surrounded by a DRAM/ROM controller, VGA and LCD interface, several expansion bus interfaces (along with internal address decoding and DMA control)

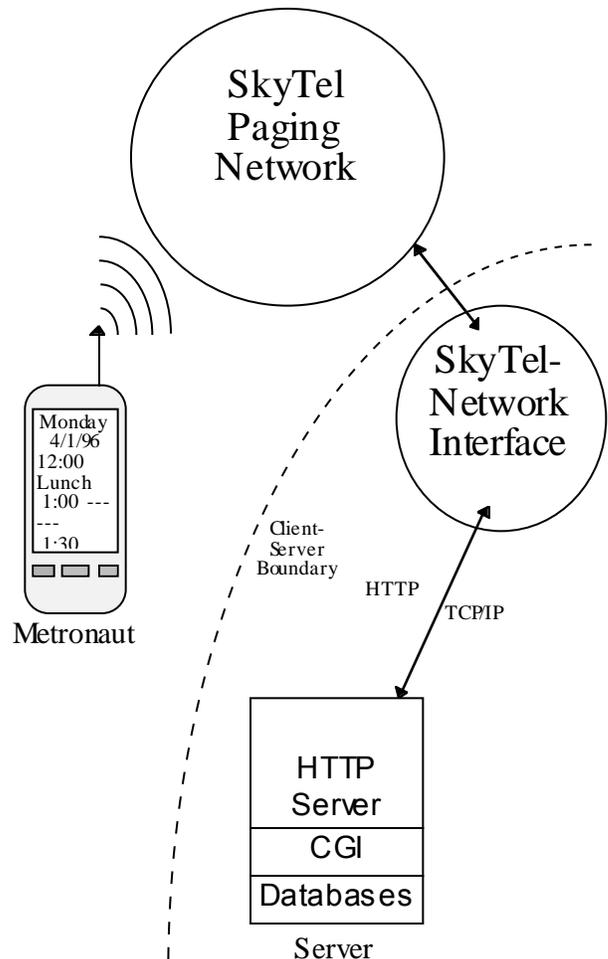


Figure 1 Application Configuration

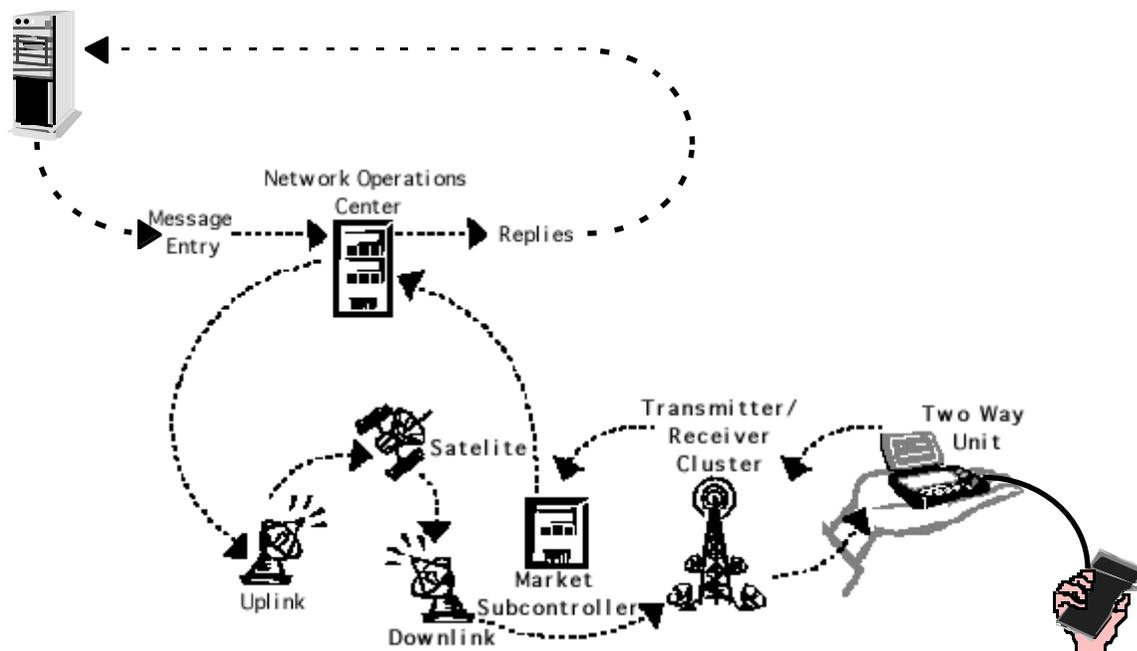


Figure 2 Communications Integration

and some miscellaneous I/O interfaces, as shown in Figure 3. The 2 MB Toshiba's DRAM was chosen for random-access memory, which is a self-refresh, low-power DRAM, requiring 3.3V. The 512KB EPROM provided the desired storage in a by-16 organization, which allowed for reduced chipcount and came in a 3.3V low power version.

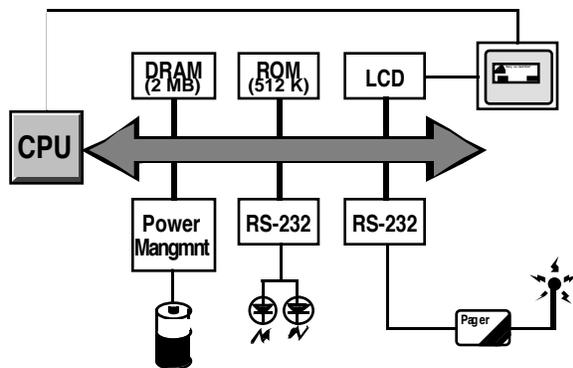


Figure 3: System Block Diagram

The DRAM and EPROM are connected via the ARM 7500's memory bus, and are controlled by its memory controller. The A/D converter and Quad-UART (QUART) use one of the ARM 7500 expansion buses

(simple I/O bus). The LCD is driven directly by the ARM 7500, and the touchscreen is digitized by the A/D converter. The LCD is Seiko's 192x192 reflective display, consuming 140 mW. The QUART provides the RS-232 serial ports for the pager and the bar code scanner. Figure 4 presents a photograph of the Metronaut with two-way pager, bar code scanner, LCD, and touchscreen.

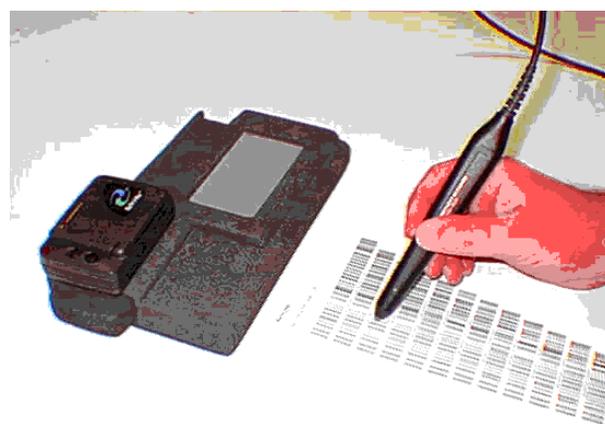


Figure 4 Metronaut System

3.3.1 Serial Device Interface

The Metronaut's serial device interface is built upon three layers of abstraction as shown in Figure 5. Resident at the top layer, denoted "Application Level", are the high-level functions that can be used in the application software to perform complex communication tasks quickly and easily. The middle layer contains the software device drivers. These functions are used by the Application Level to access the hardware. This layer also implements specific communications protocols, such as

the two-way pager's Xmodem protocol. The last layer is the physical hardware. This layer is responsible for the actual transmission and reception of bits of data from the serial devices themselves. This layered approach to the serial device interface makes writing software and debugging hardware easier by decomposing the overall system into its components.

In the following paragraphs, each peripheral device will be discussed separately followed by a summary of how the serial device interface operates together as a system.

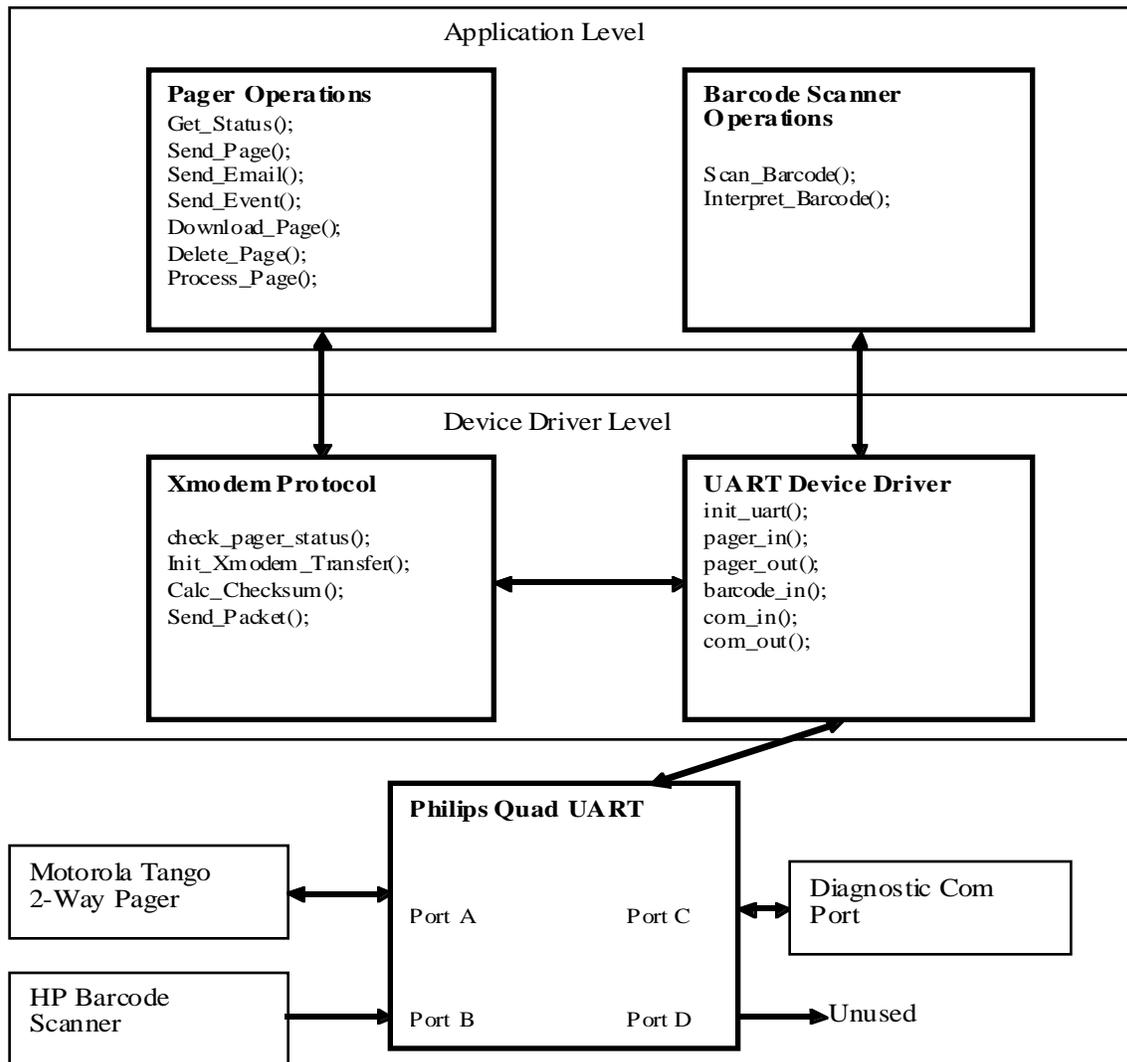


Figure 5 Metronaut Serial Interface

3.3.2 Bar Code Scanner

The HP bar code was easily integrated into the Metronaut. Whenever the user scans a bar code, the scanner automatically translates the bar code pattern into ASCII text format and sends the characters to Port B of the Quad UART. The UART then places the received characters into a 16-level deep queue so they may be read out by the Metronaut. The Hardware Layer only needs to provide power for the bar code reader. Although the bar code reader has several configuration options that can be set by sending escape character sequences to the reader, this functionality is not needed by the Metronaut application and is left unused. At the device driver level, no special protocol is necessary to communicate with the bar code scanner. The function `barcode_in()` fetches the bar code data one character at a time from the UART's queue, reassembles the characters, and returns an ASCII string representing exactly the bar code label that was scanned. The Application Layer functions encapsulate the lower-level functions to perform complete operations with just one step.

In the Metronaut application, bar code stickers that contain encoded location information are placed in important locations around a city or campus. The Metronaut software uses the Application Layer functions to decode the information encoded in these bar codes and is able to determine where the user currently is. Secondly, scheduling information can be encoded into a bar code. After scanning in such a bar code, the Metronaut can decode the name, date, time and location of important events and pass this information to the scheduling application.

3.3.3 Two-Way Pager

The hardware level interface to the two way pager is very similar to that of the bar code scanner. The pager transmits characters one at a time which are received by the UART and placed into a queue. Characters that are to be transmitted to the pager are placed into a queue of their own. Additionally, the hardware level provides the voltage conversion necessary to match the pager's specifications with the UART's requirements. The device driver level is broken into two parts for the pager. The first part, like that of the bar code scanner, simply sends and receives characters directly from the UART's queues. The second part implements the Xmodem protocol which the pager uses for all communications. This protocol is included in the device driver level because of its relative complexity.

The Xmodem protocol encapsulates all commands, messages, pages, and other data to be sent to the pager in packets of fixed size. The protocol also uses checksums to perform basic error checking. The protocol functions combined with the low-level device driver functions provide the basic functionality necessary for communication with the two-way pager. At the application level, the Motorola Tango two-way pager provides the Metronaut with a channel for communication with a server computer. With the high-level functions, the Metronaut application software can easily use the pager to retrieve the user's e-mail, scheduling information, and location information from the server computer.

4 Communication Experiments

Two options for wireless communications, two-way pager [3] and Cellular Digital Packet Data (CDPD) [4], [5], were evaluated. For off campus wireless access, CDPD was used in earlier research projects. The CDPD service employs idle voice channels on the existing Advanced Mobile Phone Service (AMPS) cellular telephone network to transmit data packets at a raw data rate of 19.2 kbps. In general the CDPD latency (defined as the time required to exchange packets sequentially in both directions) was found to usually fall in a narrow band around .5 seconds, however latencies of up to 12 seconds occurred sporadically. CDPD throughput demonstrated considerable variation with an average value of almost 11 kbps going to the mobile unit and almost 10 kbps coming from the mobile unit. Table 1 compares the measured performance to the maximum theoretical throughput of a CDPD channel, with the deterministic overhead bits added by the CDPD air interface protocol removed. The difference between theoretical and measured throughput is due to the added size of the TCP/IP headers,

Direction	Theoretical [kbps]	Measured [kbps]
To Mobile Unit	11.8	11
From Mobile Unit	13.3	10

Table 1 CDPD Throughput Comparisons

which are not considered in the theoretical throughput calculations. Additional measurements on the CDPD system are on-going to isolate the cause of the occasional connection hanging that has been experienced. CDPD

consumes approximately one watt, and its adapter alone weighs close to half a pound. Also, CDPD service is sometimes available in only part of a metropolitan area.

The two-way pager provides much wider coverage and its components are small and lighter weight. The interface to two-way paging devices is simply a serial line. A disadvantage of the two-way pager is a much lower bandwidth. In order to meet the goal of “one pound one watt,” and to provide a wide area coverage, the two-way pager was adopted. It uses Motorola’s FLEX protocol which provides a theoretical bandwidth of 9.6 kbps to the network and 6.4 kbps to the mobile user.

Page sent via Internet	4- 6 minutes
Page sent using modem	≤1 minute

Table 2 Two-way Pager Latency

Experiments have indicated that if a page is sent over a modem as opposed to the internet, the latency can be reduced to less than one minute, as shown in Table 2.

5 Conclusions

Metronaut is a low weight, low power consuming wearable computer employing a two-way pager for communications and a bar code reader for information input and position location. It includes a custom designed printed circuit board and serial device interface to connect the two-way pager and bar code scanner. Metronaut also employs a touch screen for input and LCD for displaying information. The overall system architecture also includes a network server, SkyTel paging network, and SkyTel network interface. The current applications are navigation, scheduling, and messaging. The concept of a bar code Stickernet on campus objects has been introduced. These stickers act as hyperlinks to relevant information on the

internet. The sticker net provides a link between the point of information need a user experiences while interacting with real objects and the point of information delivery from the Web. Metronaut can be applied to other application domains such as medical patient monitoring, inventory, and inspection.

6 Acknowledgments

This work was supported by the Engineering Design Research Center at Carnegie Mellon University, an NSF Engineering Research Center, and Defense Advanced Research Project Agency Contract DABT 63-95-C-0026. We would like to acknowledge Dan Siewiorek for his valuable insights and contributions, and the students who contributed to the project: Joseph Rowlands, Berend Ozceri, Bohuslav Rychlik, Wes Horner, and Andrew Mihal.

7 References

- [1] Smailagic, A., Siewiorek, D. P., “The CMU Mobile Computers: A New Generation of Computer Systems”, Proc. IEEE COMPCON 94 International Conference, IEEE Computer Society Press, Los Alamitos, CA, pp. 467-473, Feb. 1994.
- [2] Smailagic, A., Siewiorek, D. P., “A Case Study in Embedded- Systems Design: The VuMan 2 Wearable Computer”, IEEE Design and Test of Computers, vol. 10, no. 3, pp. 56-67, 1993.
- [3] 2- Way Paging Reference Manual, SkyTel Corp., March 1996.
- [4] CDPD Consortium, “Cellular Digital Packet Data System Specifications,” Release 1.0, July 1993.
- [5] Johnson, D., Drum, J., “Preliminary CDPD Performance Testing Results”, Information Networking Institute report, Carnegie Mellon University, 1996.

