A Technical Journal for PDA Developers

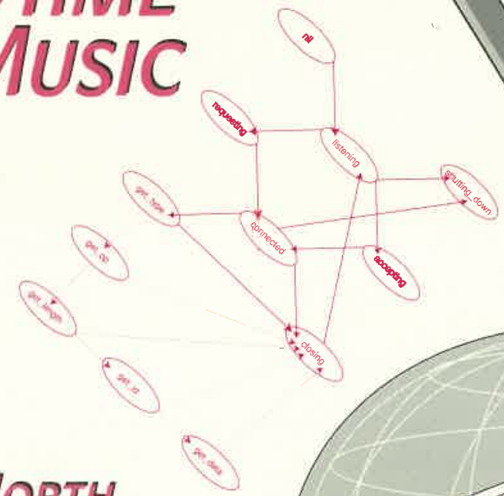
PDA DEVELOPERS

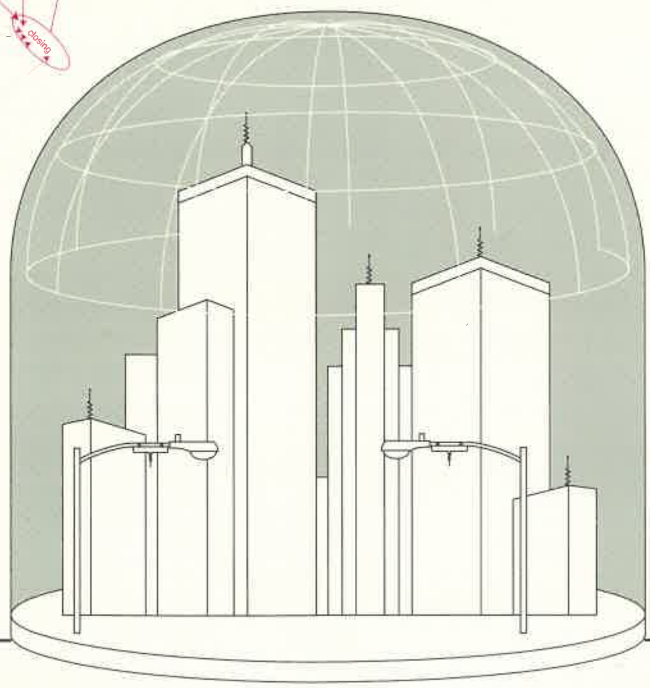
Volume 4.4 • July/August 1996

$13.95 US • $14.95 Canada

JAVAOS – FIRST LOOK
RICOCHET
NEWTON UNITS
IZL AND THE OMNIGO EXPRESS
REAL NEWTON COMMS
PILOT WORLDTIME
MAGIC CAP MUSIC
PLUS

LOCALIZATION
DYNAMIC BUTTONS
NS BASIC
ALLPEN SOFTWARE
PDA DEVELOPERS NORTH
AND MUCH MORE...





07
0 74470 84374 0

```
0740 REM delete RI, return to main display
0750 GOSUB 0460 //Delete the selected item
0770 HIDE itemSel, itemSelBtns
0780 GOTO 0140 //Show main view
```

It deletes the selected item and then hides the recent item screen. Finally, it branches back to the main screen display.

If the user taps **Del Selected** this subroutine is executed:

```
0820 REM delete the selected items
0822 numDel = LENGTH(riScreen[0].selectedItems)
0824 IF numDel = LENGTH(riFrame.items) THEN GOTO 740
0830 FOR i = 0 TO numDel -1
0840 ARRAYREMOVECOUNT(riFrame.items,
        riScreen[0].selectedItems[i]-i, 1)
0850 NEXT i
0860 PUT ch, riFrame
0865 itemArray[mainScreen[0].selection] :=
      SUBSTR(riFrame.tag,3,STRLEN(riFrame.tag)) &
      " (" & LENGTH(riFrame.items) & " items)"
0870 recentItems := []
0880 FOR i = 0 TO LENGTH(riFrame.items) -1
0890 ADDARRAYSLOT(recentItems, riFrame.items[i].item)
0900 NEXT i
0910 riScreen[0].listItems := recentItems
0911 riScreen[0].selectedItems := []
0912 U.riScreen[0]:SETUPLIST()
0914 U.riScreen[0]:REDOCHILDREN()
0915 SETVALUE(riScreen[1], 'text, "recent items for: " &
      itemArray[mainScreen[0].selection])
0916 RETURN
```

I first determine if they selected all the items in lines 822-824. If they did I just branch to the **Del All** routine. If they didn't, I iterate over the list of selection numbers in `riScreen[0].selectedItems`. This array has one element for each selected item. I remove each selected element from the recent item frame in lines 830-850. The edited frame is stored back to the file in line 860. Lines 865-914 update the recent items display to reflect the new list. Line 865 updates the item text shown in the main screen, line 870 clears the selections from the `text list`, and lines 880-915 update the `text list` to remove the deleted items.

When the user taps **Done** I hide the recent items screen and go back to the main screen display:

```
0920 REM done with RI review
0930 HIDE itemSel, itemSelBtns
0940 GOTO 0140 //Show main view
```

Test Drive

You've already seen the displays produced by the both versions of this program. If you built both, you'll notice that the visual version is noticeably faster. It's snappy to use, incorporates familiar user-interface elements, and is very functional. The version with the text-based interface is cumbersome to use, slow, and not very Newton-like. Note that both versions are still missing some niceties, such as confirmation dialogs before deletion.

I jumped into visual programs with both feet this time. Next time I plan on discussing some of the little details I glossed over this time, like stacking order, updating, extracting values from widgets, and more. The RIE program contains most of these features, so I'll probably revisit it next time. ✔

The RIE source code can be found on this month's source code disk.

NS BASIC 3.5

The NS BASIC corporation continues its aggressive policy by recently releasing yet another version (3.5) of NS BASIC. This version includes several additions to both the NS BASIC environment and the language.

Environmental Update

The NS BASIC development environment is now split into several parts that you can selectively install. If you need a feature, then you install its component. The components are: NS BASIC, the interpreter; MakePackage, the stand-alone, package-building tool; Visual Designer, a tool that lets you lay out most of the visual components of your application directly on your Newton; and the NS BASIC runtime for distributing "skinny" packages.

The interpreter is now a svelte 124K installed, down from almost 200K in the previous version. One great new feature of this version is the **Enable Break** check box. When enabled, you get a nice big **Stop** button that you can tap to halt the currently running program. Now you don't have to reach for the reset button just because your program is looping forever. This version of NS BASIC also implements a new storage approach for programs that reduces their size a bit, and provides a boost to performance as well.

The MakePackage part is for creating stand-alone packages from your programs. This feature has been expanded from version 3.0. You can now make skinny or fat packages. Skinny packages must be distributed with the NS BASIC runtime package, but they are very small. Fat packages include the runtime package, which makes them much bigger. For example, skinny RIE is 14K; fat RIE is 102K

Visual Designer

The Visual Designer is directly integrated into the NS BASIC environment. You access it by entering and then editing using the new `WIDGETDEF` statement. You create new widgets by tapping **New**, then selecting the desired widget from the pop-up menu. You then can move , re-size, and set the initial property values for the widget. This makes most programs that use widgets a breeze to create. No more trial and error (or tedious work with graph paper) to get the `viewBounds` correct for each widget.

You still need to do some work outside of the Visual Designer. Plain windows (like those used for buttons) and the `APP` widget are not currently supported in the Visual Designer. The idea really is to lay out the guts of each screen in the Visual Designer and then `SHOW/HIDE` them as needed on top of a single `APP` widget that is always visible.

Language Features

There are some changes to the language to support the Visual Designer. The `WIDGETDEF` statement is one. Another is that window specs can now contain the name of the widget. By using this feature you can combine several window specs in a single array and then create them all with a single `WINDOW` statement. The `SHOW` and `HIDE` statements have also been expanded to accept arrays as well as single window IDs in a list. Your widget programming should be much easier with these enhancements.

This version also introduces statement labels in addition to line numbers. You can add a label before any statement and then use that label instead of a line number, as the target for a `GOTO`, `GOSUB`, or `windowSpec` field for instance. I had some problems using labels in the RIE so I took them out, but they look like a great addition to the language. We're one step closer to getting rid of those pesky line numbers. ✔

Previews

The Ricochet Network

Metricom. Inc.
http://www.metricom.com

Ricochet is both a wireless networking service and a wireless modem based on technology from Los Gatos California-based Metricom. If you use the service, you can get unlimited wireless Internet access for $29.95 a month, plus $10 a month for leasing the modem ($299 if you buy the modem with a on-year service plan). For an additional $5 per month, you can add wireless telephone access to dial-up on-line services like CompuServe and America Online. You can also use Ricochet modems to create small, private wireless networks, with custom protocol support.

Metricom's Ricochet technology has several advantages over other types of wireless services:

- *It's versatile. The modems support AT, PPP, and SLIP protocols. They can be used with most communications software and practically any device that has a serial connection.*
- *It uses unlicensed spectrum, the same spectrum used for garage-door openers (902-928 MHz). Metricom didn't have to spend billions of dollars buying the spectrum from the FCC, making it possible to keep their costs down.*
- *It's scalable, from individual users in Metricom's coverage areas to large campuses. It's possible to start with just a few radio modems (the device that coordinates connections between individual modems and the Internet) and add them as needed to increase the geographical size of the network.*
- *It's faster than most of today's wireless services. The basic network is rated at 100 Kbps; 14.4 - 28.8 Kbps throughput is typical.*
- *It's programmable. Each modem has a special Star mode which lets you incorporate your own protocols for specialized needs. In Star mode you can take advantage of the full 100 Kbps bandwidth.*

There's only one serious disadvantage to Metricom's wireless service: it's not fully deployed throughout the United States. Currently Ricochet is available in most of Silicon Valley and the San Francisco Bay area, Washington D.C., and a few other cities. (Contact Metricom for a complete, up-to-date list.) If you want to do a custom campus-wide installation, you don't need local Metricom service. Instead, Metricom can install an Internet access point for you.

We've wanted to do a story on Metricom for some time now, but never had the right vehicle. Recently, we had a chance to see Metricom's modem manual for developers, and were impressed with its clarity, its objectivity, and with how well it summarizes Ricochet. We asked Metricom if we could reproduce it here.

Metricom's Ricochet Network is a wide-area wireless data communications service. Ricochet subscribers make data connections from desktop, laptop and PDA (Personal Digital Assistant) devices via wireless modems to local area networks and a variety of regional, national and international networks. Subscribers to Metricom's Ricochet Network can access:

- Internet services (e.g., Netscape, Mosaic, NetCruiser)
- Dial-up services (e.g., Prodigy, AOL, cc:Mail Mobile)
- Other modems (wired or other Ricochet modems)

Ricochet uses the license-free Part 15 Band at 902-928 Megahertz (MHz). Minimum power and transmitter duty cycles are used in the network to maintain high throughput and "neighbor friendly" operation. Metricom systems employ very robust radio and protocol design to maximize reliability.

Ricochet Network Overview

The Ricochet Network combines the advantages of Metricom's high-wired network and the convenience and flexibility of Radio Frequency (RF) networks. It consists of a private Internet Protocol (IP)-based backbone network with connections to other networks and RF devices. Radios throughout a coverage area collect signals from Ricochet modems and pass them onto the wired backbone, where they are processed and passed through to connection service points.

Ricochet Modems

Ricochet modems are AT-compatible modems which, instead of using a wired telephone network, make their connections using Radio Frequency. Users can make modem connections anywhere within a coverage area without having to have a dedicated wired connection between their Ricochet modem and the desired connection service. Ricochet modems also can communicate with other Ricochet modems on a peer-to-peer basis.

The Ricochet modem has two connectors, a serial port for computer connection and an AC adapter connector. Except for the cable used to connect it to the computer, the same Ricochet modem can be used with either a PC or Macintosh computer.

Network Radios

Network radios typically are installed on poletops throughout a geographic area. Radios form the mesh of the Ricochet Network, routing packets from radio to radio, funneling them onto Metricom's wired backbone network. Note that unlike Ricochet network radios, Ricochet modems do not perform radio packet routing functions.

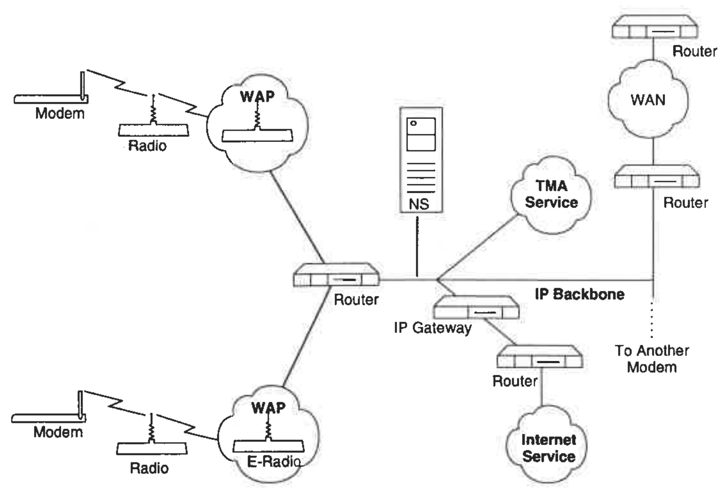


Figure 1 - Overview of the Ricochet Network.

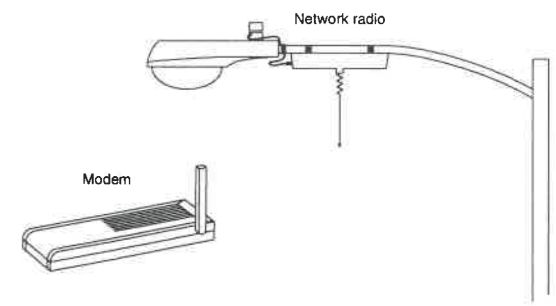


Figure 2 - Ricochet radios.

Name Server

The Name Server resides on the Metricom backbone and provides connection validation and path information to Ricochet modems. All connection requests go to the Name Server first before a connection can finally be made.

The Name Server is integral to operating Ricochet's dial-up and Internet access services. When a Ricochet modem is first powered on, it sends a packet to the Name Server containing the Ricochet modem's serial number. The Name Server performs two functions: it validates both the subscription and the service request. If either the subscriber serial number is invalid or the service being requested is not what the subscriber has purchased, the access request is denied. Only after the Ricochet modem successfully registers with the Name Server can the subscriber obtain Internet or phone-line service access on the Ricochet network.

Peer-to-peer operation does not require Name Server registration.

Ricochet IP Gateways

Gateways reside between the Metricom backbone network and the Internet. They provide connection services to the global Internet and other enterprise IP-based networks.

When packets are Internet-bound, they are routed to the IP Gateway, which transforms them into IP packets so they can be transmitted to hosts on the Internet.

Bridges

RF-to-phone line connections required for Ricochet's Telephone Modem Access (TMA) service are made using specially configured Ricochet modems. When a Ricochet modem is configured to operate in bridge mode, it translates signals from other Ricochet modems into signals that a standard wired modem can receive. Subscribers can set up their own personal TMA, too. This is discussed in the *Ricochet User's Guide.*

Ricochet Ethernet Radio WAPs

The Ethernet Radio operates just like other network radios, but includes Ethernet hardware and software capabilities. It can be installed directly onto an Ethernet network using 10-BASE-T cabling. Its Ethernet software capabilities convert RF network packets into a format for transmission over an IP-based network.

Banks of Ethernet Radios are installed at key locations – Wired Access Points (WAPs) – throughout the Ricochet Network. They pass RF signals from radios onto the Metricom wired backbone network. Radios are configured to send their incoming packets through a specific WAP, thereby reducing the number of "hops" an RF packet might take to reach the Metricom backbone network.

Ethernet Radio WAPs consist in part of a rooftop installation of up to 16 Ethernet Radios. Inside the WAP site are the other WAP components – one distribution box for every eight radios, a hub/router and a CSU/DSU unit with connections to a T1-based Frame Relay network.

Radio Network Operation

The following describes the operation of radios and Ricochet modems in the Ricochet Network.

The "Mesh Network"

The Ricochet Network uses radios to form a "mesh network" architecture. This means that the network has a number of strategically installed radios for a given city or town whose combined coverage forms a mesh for routing radio packets over that geographic area.

Physically, Ricochet radios are very small compared to conventional cellular telephone base stations. They can easily be mounted to street light poles or on top of buildings and houses. These radios operate using a "Gaussian" Frequency-Shift Keying (FSK) modulation scheme for baseband modulation. These radios employ 163 hopping channels with each using a different hopping sequence and phase. In the Ricochet Network, data packets have a maximum length of 1189 bytes and radios can forward packets at a data rate of 100 kilobits per second.

Acquisition

Acquisition is the necessary first step for each radio on the network. Each radio or Ricochet modem must receive at least one packet from another Ricochet modem or a radio within range before it can send or receive packets within the network.

When a radio is powered on, it has no knowledge of any of its radio neighbors. It attempts to acquire neighboring radios (radios or Ricochet modems) by sending out packets where neighboring radios might be listening.

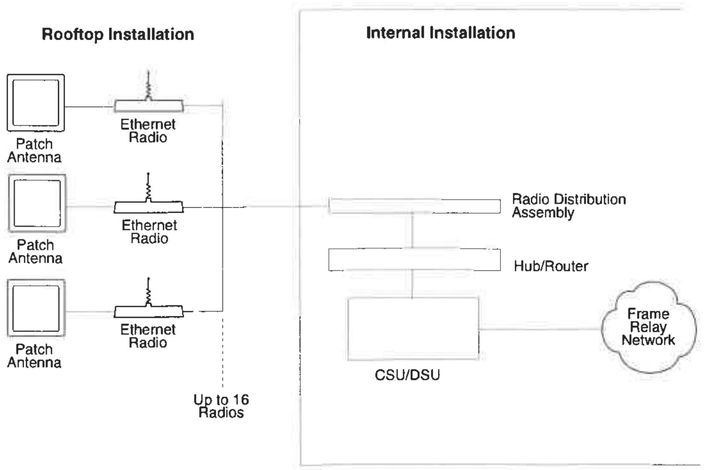


Figure 3 - Typical Ethernet radio WAP site.

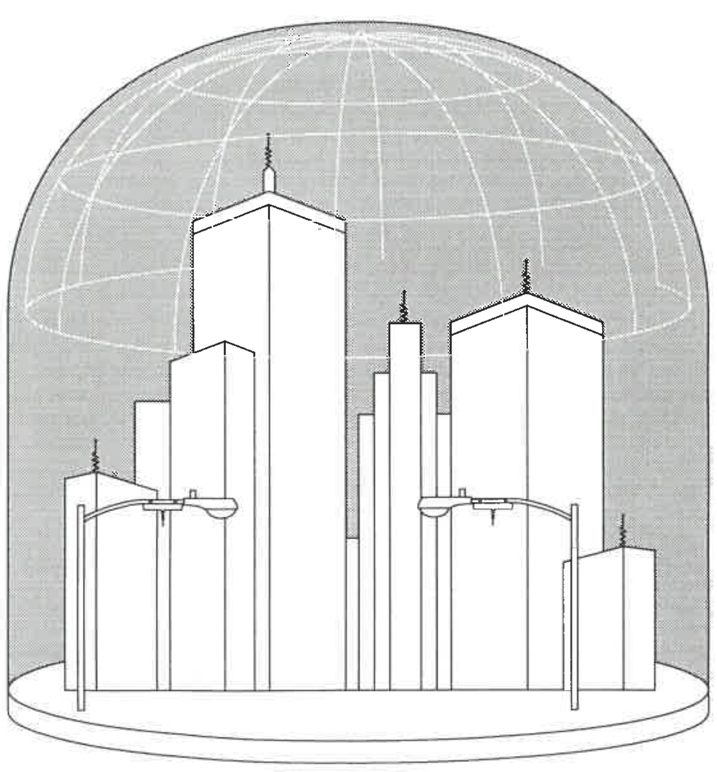


Figure 4 - The Ricochet network mesh.

Once it receives a packet from another radio, the information passed in that packet allows the radio to become a fully functional member of the mesh network.

Through the acquisition packet a radio learns the locations and visibility of its neighboring radios and the location of neighboring Ricochet modems. The Ricochet modem learns the location of the nearest radio it can use to communicate with the network.

When the Ricochet modem's light blinks and sounds a two-tone beep (from low to high), it has acquired the network and verified a user's Ricochet Service subscription. If the Ricochet modem's light blinks and the Ricochet modem does not sound a two-tone low-to-high beep, the Ricochet modem has not acquired the network. However, the Ricochet modem can perform peer-to-peer communication.

Operation

When the Ricochet modem has data from the radio, it uses the radio's routing information to prepare for sending and receiving packets. Upon receiving an "Acknowledge" (ACK) packet from the radio, the Ricochet modem sends its data packets to the radio.

If another Ricochet modem is already communicating with the radio, it will not receive the "ACK" and simply retries at a random time interval.

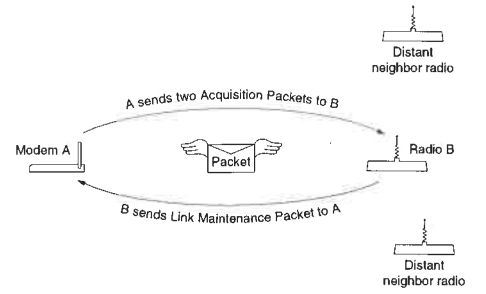


Figure 5 - Sample Ricochet modem network acquisition.

Packet Routing

Radios have unique addresses – Wide Area Network (WAN) addresses – that reflect their geographic coordinates in latitude and longitude. During acquisition, radios provide their neighbors with their WAN addresses. Routing of packets through the network is greatly simplified through this addressing scheme because each radio knows where every other radio is located. In this way, packets are routed to radios progressively closer to the final destination.

The mesh network allows for alternate routing of packets should a given radio be busy or out of operation. Consequently, each of a series of packets might take very different routes through the network.

Except for peer-to-peer Ricochet modem communications, all packets are funneled from the radio network onto the Metricom wired IP network. Here subscriber service registration takes place and packets destined for Ricochet's services, such as Internet and TMA, are routed to the appropriate service connection point. In addition, communication between geographically distant points in the radio network can take advantage of the high-speed packet transmission available on the wired network.

Maintenance

Each radio uses a different free-running clock as a timing reference. Since the radio's hopping sequence and phase information are essential for network operation, this information must be updated periodically, about every half hour. The relative phases of the hopping sequences remain randomized over the network. Along with phase information, additional status data will be sent. Maintenance packets are small, infrequently issued packets that are sent during relatively low network usage periods.

How Network Connections Are Made

Ricochet Network subscribers can use their modems to connect to:

- Ricochet's Internet service
- Telephone Modem Access (TMA) service
- Another Ricochet modem

All but peer-to-peer connections pass through the Metricom private IP networks. Connection requests pass from the radio network to the Metricom IP network and onto the requested service.

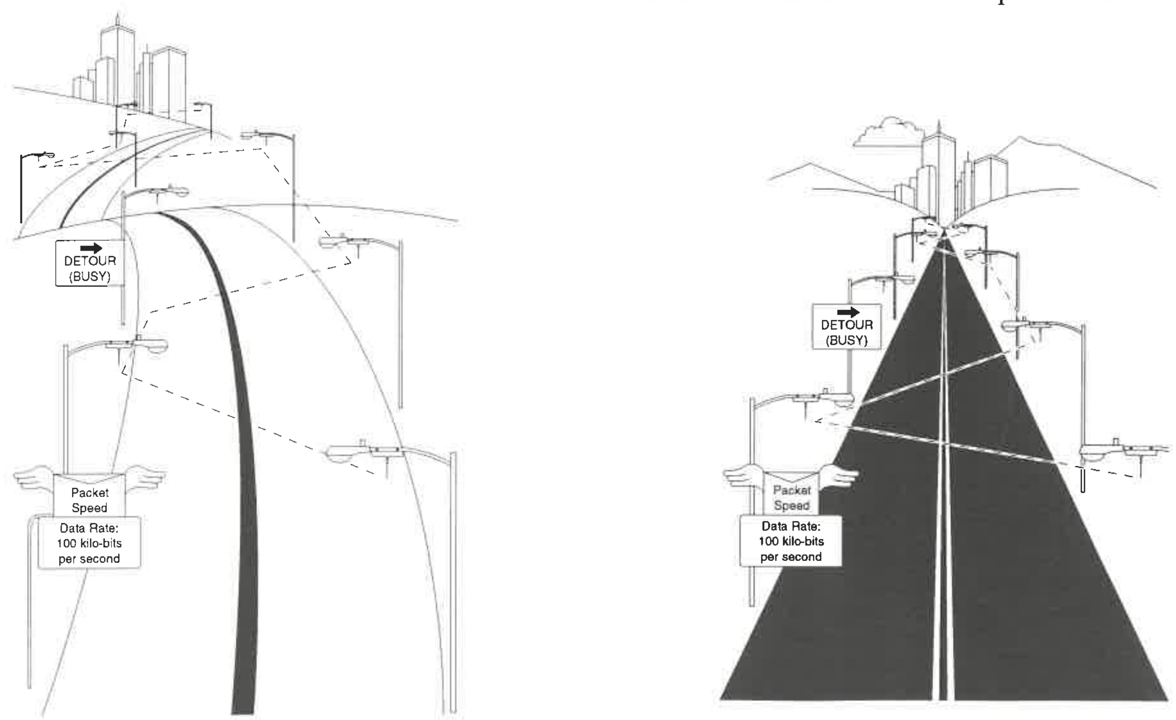


Figure 6 - Ricochet routing scheme.

A Ricochet modem gains access to the network by using the installed radio network to reach a Metricom IP network entry point, the WAP. After Name Server validation, a packet can take one of two paths, one bound for the Internet service and one for the TMA service. If the packet is bound for the Internet service, it is passed to a Ricochet IP Gateway. If the packet is bound for the TMA service, it is passed to a radio used for dial-up services.

Internet Access

Access to the Internet service requires Name Server authorization. The Ricochet modem's RF packet is passed to a radio and onto a WAP over the Metricom IP network then to the Internet. The following diagram shows how a Ricochet modem connects to the Internet: In this case, Internet access consists of the following steps:

1. Before a Ricochet modem sends an Internet connection request, it issues a lookup request to the Name Server.
2. The Ricochet modem lookup request traverses the RF network, entering the Metricom backbone IP network through the WAP site.
3. At the WAP the RF packet is encapsulated for transmission over the Metricom IP network as a UDP packet to the Name Server.
4. The Name Server validates the subscriber and the services purchased by the subscriber. In its response, it also tells the Ricochet modem the path to take to access the Internet service. This path includes the address of the Ricochet IP Gateway to use.
5. The Ricochet modem then issues a request to connect to the IP gateway. Once the gateway responds to the modem's request, a virtual circuit is set up between the modem and the gateway.
6. The computer attached to the Ricochet modem issues a request to connect to an Internet host. The Ricochet modem forwards the packet to the WAP. When it reaches the Ricochet IP Gateway, the gateway will turn the encapsulated UDP packet into a true IP packet. The IP packet is forwarded to a router that serves as a "firewall" to the Metricom IP network and from there the packet is forwarded to the target Internet host.

TMA Service Access

Before a subscriber can access the TMA service, the access must be registered from the Name Server. After access is granted, RF packets from the Ricochet modem pass from radio to a WAP before finally reaching the TMA service bridge to complete the connection to a target on-line service provider (such as AOL, CompuServe, bulletin board). The following shows an example of how a Ricochet modem connects to the TMA service:

Peer-to-Peer Access

Two Ricochet modems can establish connections with each other for peer-to-peer communication. Unlike Internet or TMA service access, peer-to-peer communication does not require Name Server validation. The Ricochet modems can acquire each other directly if within unobstructed quarter-mile RF range of each other using RF packet routing. Peer-to-peer connections may also use RF-to-Metricom IP network paths, but only if modems are subscribed to the Name Server and authorized to use the Metricom network. ✔

For more information about Metricom's technology, products or services, call 408.399.8200 or take a look at their web site at http://www.metricom.com.

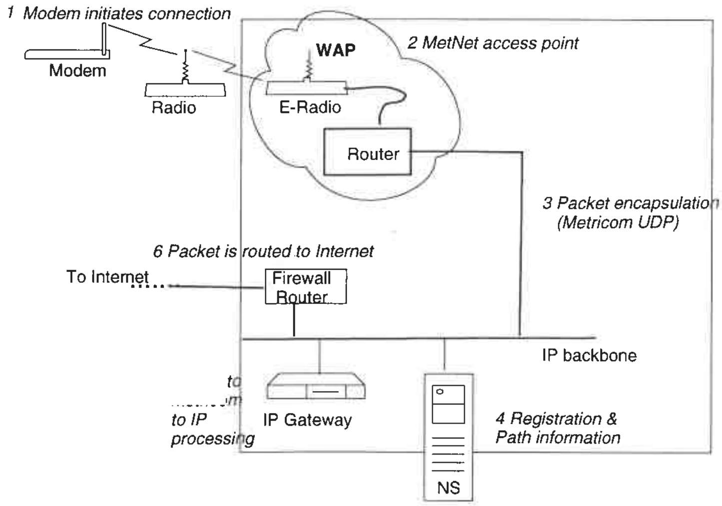


Figure 8 - Internet access service using the Ricochet IP gateway.

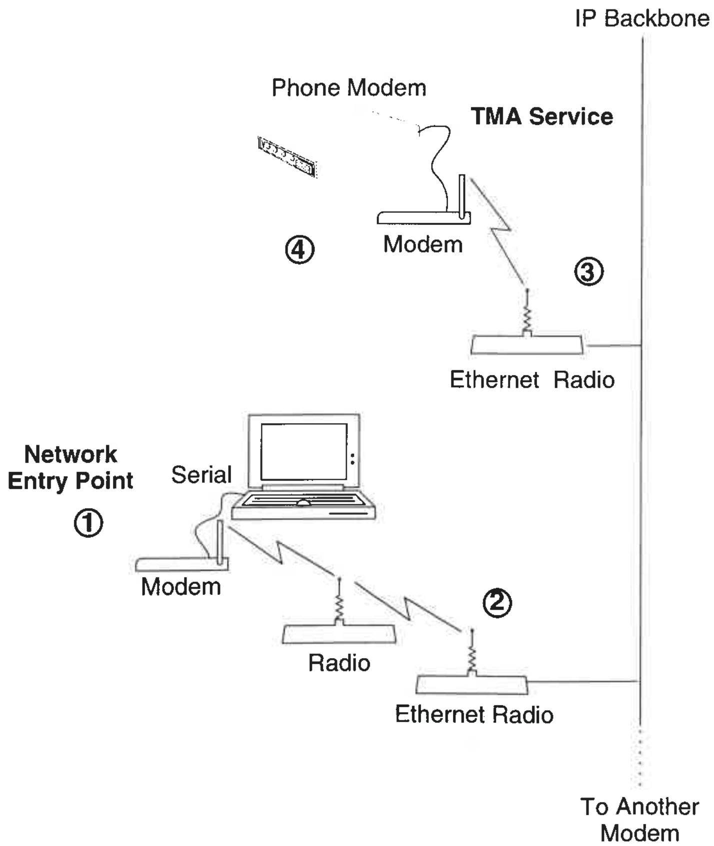


Figure 7 - How TMA service connections are made.

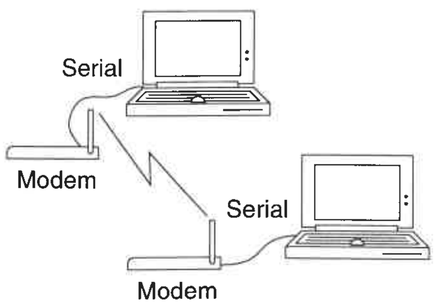


Figure 9 - Peer-to-peer access overview.