Mobile Connectivity Protocols and Throughput Measurements in the Ricochet MicroCellular Data Network (MCDN) System

Mike Ritter, Robert J. Friday, Rodrigo Garces, Will San Filippo, and Cuong-Thinh Nguyen, Arty Srivastava
Metricom, Inc.
333 West Julian Street
San Jose, CA 95110
{mwritter, friday, rgarces, will, ctnguyen, asrivastava}@metricom.com

ABSTRACT
We describe the protocols implemented in the Ricochet MCDN system to provide continuous connectivity to mobile users traveling up to 70 mph. These protocols are general in nature for any frequency-hopping microcell-based system, particularly those that follow the FCC part 15.247 rules [9] and operate in unlicensed spectrum. We also present throughput measurements as a function of velocity and describe a model to predict those numbers based upon the protocols implemented. The MCDN system is a mesh-based system of microcells that are connected wirelessly to an interspersed mesh of wired access points (WAPs) that cover approximately 12 square miles on average [7]. The average microcell density is approximately 5-6 per square mile, with 3-8 overlapping cells at each point. Since the system is entirely packet-based, we have instantaneous hand-off between microcells as there are no complicated cellular-type negotiations for circuits required as all of the information needed to route the packet through the system is included in the header; however, when traveling through the mesh of microcells at a high rate of speed, the mobile unit must acquire new microcells fast enough to ensure continuous connectivity. The system must also know how to route packets to the mobile unit as it drops old cells and acquires new ones, as well as being able to contact a moving mobile unit. This paper discusses the acquisition, registration, and routing protocols that make this possible and reviews performance data of typical mobile users.

Keywords
Mobility, Wireless Networks, Wireless Protocols, Wireless Routing, MCDN System Architecture.

1. INTRODUCTION
A user of the MCDN system has a radio modem, also known as a subscriber device or mobile unit, attached to or embedded into some computing device that requires network connectivity. This computing device (or in some implementations, the modem) must support PPP in order to negotiate its way onto the network. The radio modem typically appears to the user's computing device as a modem supporting the AT command set. This mobile unit can appear at any place in the MCDN system and can be mobile at speeds up to 70 MPH while sending and receiving packets. The goal of the MCDN mobile architecture is to make this possible.

There has been much work done lately on ad-hoc routing for mobile nodes [1], [3], [4]. This total solution is not required for the MCDN architecture as the infrastructure, which consists of microcells either wired or wirelessly connected to a high-speed backbone, is fixed in place, as will the vast majority of commercial deployments of any microcell system. Only the subscriber devices themselves change places on a near real-time basis. This allows us to split the routing system into three different links: 1) between the gateways and the wired microcells over the wired backbone, 2) the wireless links between the mesh of fixed microcells and wired microcells, and 3) the links between the microcells and the mobile units. The wired backbone uses standard IP routing by encapsulating the wireless packets in UDP and forwarding them to a gateway. The wireless mesh uses a proprietary Bellman-Ford type of routing system based on throughput estimates when moving packets towards the wired backbone, and a proprietary loose-source routing system when moving packets towards the mobile units. The link between the mesh of microcells and the mobile units is a pure broadcast system at the data link level based upon acquisition tables kept in the microcells. This architectural separation allows the wireless routing system to scale independent of the number of subscriber devices, thus making it feasible to support hundreds of millions of mobile units in its current incarnation.

Another method of supporting mobility in IP is the Mobile IP specification [6]. This specification was not optimized for the high degree of mobility required by the MCDN system, and the overhead of maintaining and configuring all of the microcells and mobile units with fixed IP addresses and IP subnets is undesirable. However, the architecture upon which Mobile IP is based, that of a care-of-address server, is used in the network in the form of MCDN gateways which funnel all of the user's traffic

ACM SIGMOBILE 7/01 Rome, Italy

to the Internet or an intranet. When a user's device connects to the network it typically dials the radio modem with an ATDT command using a logical name or phone number. Upon receiving that command, the radio modem asks the Name Service to resolve that logical name into an MCDN path (in an analogous manner to DNS.) The key to any addressing architecture is the ability to name and route between entities in the system. The MCDN Name Service provides this by mapping any pair of names of entities to a list of addresses that can be used to route packets between the two entities. This route is referred to as an "MCDN Path." The MCDN Path consists of a list of addresses of waypoints between the two entities. A waypoint can be, for example, an IP address, a phone number, a geographic area describing a list of microcells, or a microcell identifier. The routing system ensures that the packet visits each waypoint and thereby delivers the packet to its final destination.

Once that resolution is done, the mobile unit can append the MCDN path to any packet, send it to any microcell, and the packet will be automatically routed to its destination, the correct logical device. When the user's computing device is attempting to negotiate PPP to connect to the network, the radio modem establishes a virtual connection, analogous to TCP, to the MCDN gateway. The MCDN gateway ensures that all of the packets from the user's computing device get routed to the Internet, analogously to Mobile IP. When the virtual connection is complete, the radio modem returns a command response of 'OK' to the user's device. The user's device then continues to negotiate a PPP connection so that it can be assigned an IP address, an IP router, and a DNS server and become a full-fledged communicating member of the global Internet. The PPP session can either be terminated at the MCDN gateway or be forwarded, using L2TP (Layer Two Tunneling Protocol [8]), to be terminated at another gateway (typically owned by a reseller providing Internet connectivity or a corporation wishing to provide remote access.) The user's device then appears to the rest of the Internet as if it is physically located at the PPP termination point.

Finally, another method of providing mobile connectivity would be analogous to a cellular phone system. In this type of system the mobile unit is allowed to roam and stays connected to a cellular base station continuously, never dropping one until it can connect to another and negotiating a handoff to another base station. When a call is made to the mobile unit a connection is set up that completely defines how the signal must be routed through the system. When the mobile unit moves, it must renegotiate a new connection before switching the signal path through the network. This system ensures that the latency and throughput of a connection is guaranteed, but can be very wasteful of bandwidth, as the connection must stay up whether data is traversing the connection or not. In addition, there is no facility to change the parameters of the connections, thus when the system can no longer handle another connection the users get busy signals. In a packet-based system, which is not as sensitive to latency and has a much burstier receipt of signals, this architecture will cause excessive overhead and is unnecessary. Cellular architects have realized this and are in the process of designing systems that can overlay packet data upon their voice networks. However, they must still support voice, thus the overhead for hand-offs and negotiations do not disappear, but becomes more efficient. For instance, GPRS, the packet architecture designed to be overlain upon the GSM voice system, doubles the efficiency of the network for typical Web browsing when compared to the original circuit architecture [2]. However, in a system designed entirely for data, such as Ricochet, it is possible to further increase the efficiency by going to a true packet-based routing system and using virtual connections. Of course, the QOS required for voice cannot be easily guaranteed in this type of system and how to provide high-speed mobility is in question.

This paper describes the virtual connection between the mobile unit and the MCDN gateway and all of the services that the MCDN architecture provides to keep packets flowing over this virtual connection regardless of where the radio modem is or even if it is in motion. To keep packets flowing over this virtual connection requires several services provided by the network above and beyond the Name Service and the virtual connection protocol mentioned above. In addition, the mobile units use the Acquisition Service to acquire the microcells, they then use the Maintenance Service to maintain the relative timing of the microcells, (the physical layer is a slow frequency hopping system that requires the relative timing between radios to be accurate to within approximately 1 ms [6]), the mesh of microcells provides a Wireless Routing Service for packets to traverse between a WAP and a microcell nearby the mobile unit, a Forwarding Service provides the capability for the microcells to forward the packet through the mesh if the mobile unit has moved, and for efficiency, the microcells will use the WAP Hand-off Service to route the packets to new, nearby WAPs if the mobile unit has moved out of range of its original WAP.

The paper is split into five further sections after this introduction: 1) a description of the MCDN architecture for a static radio modem, 2) a discussion of the changes made to support a mobile radio modem, 3) a foundation for understanding the throughput provided by the system to a mobile radio modem, 4) a discussion of several measurements made to show the improvements that resulted from the modifications to the architecture to support mobility, and 5) a summary of the paper and suggestions for further research.

2. MCDN System Architecture for Fixed Point Connections

The MCDN system architecture is based on a mesh of microcell deployed throughout a metropolitan area following the part 15.247 FCC rules and regulations for the ISM band [8]. Each microcell consists of two transceivers, one that transmits to the subscriber's radio modems or mobile units, typically connected to a laptop, and another transceiver that transmits to another microcell or a Wired Access Point (WAP.) The WAPs place the wireless packets onto a high bandwidth wired backbone and route them to a central collection point, the Network Interface Facility (NIF.) At the NIF, the packets are routed to MCDN gateways that decapsulate the MCDN protocol headers and forward the packets using standard IETF protocols to resellers or corporate LANs using L2TP (Layer Two Transport Protocol.) This tunneling process, which begins at the radio modem and ends at the final PPP gateway, is shown in figure 1.

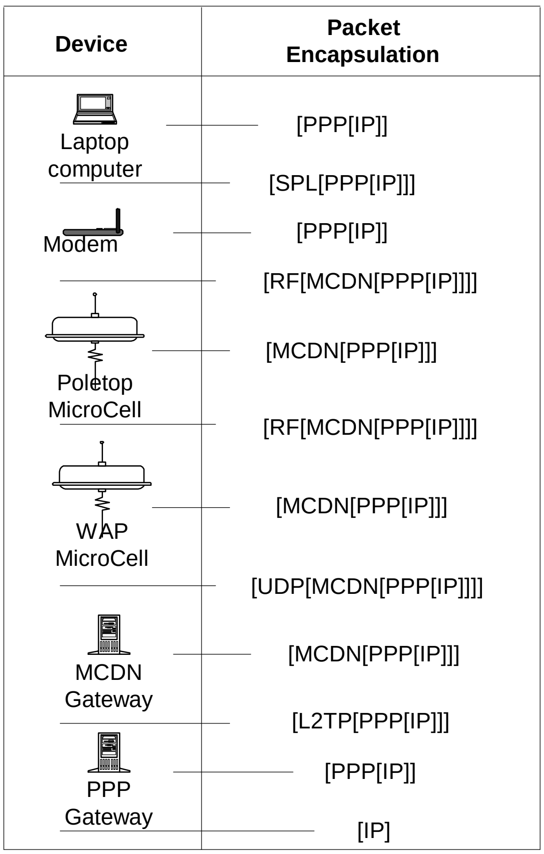

Figure 1. Packet encapsulation for an IP connection. IP is the Internet Protocol, PPP is the Point-to-point protocol, SPL is some serial protocol, RF is the physical and data link layer proprietary to MCDN (described elsewhere), MCDN is the proprietary routing layer, UDP is the standard User Datagram Protocol, L2TP is the standard Layer Two Tunneling Protocol.

Typically, the radio modem presents an AT command interface to the laptop. The laptop 'dials' an MCDN gateway by name or phone number, when the modem receives the ATDT command, it sends a request to the MCDN Name Server asking for a logical address translation of the gateway name to an MCDN path.

The MCDN path contains all of the information necessary to route any packet between the modem and the gateway through the network. The modem inserts the MCDN path in front of every packet it transmits. Unlike a typical cellular network, the MCDN path allows hand-offs to occur instantaneously. There is no need for the modem to negotiate a hand-off from cell to cell, as all of the routing information required to guide the packet through the network is contained in the header of each packet. This allows the system to support much faster hand-offs than a typical cellular system and is crucial in making a microcell based system work.

The physical instantiation of the MCDN architecture is shown in figure 2. The MCDN system is based on frequency hopping spread spectrum. Each radio in the network is hopping its receiver to a different frequency every 10-25 ms. Every radio hops on a different pseudo-random pattern so that if collisions occur at one point in time the radios will be on different frequencies on the next hop and no longer be colliding. The radios must acquire accurate relative timings from each other, MAC addresses, and the hopping pattern before they can establish data link connectivity. All of this information is traded between nodes during the acquisition process. Since the system is based on slow frequency hopping the radios must maintain relative timing accurate to within approximately 1 ms, and after acquisition they do this by listening to periodic heartbeats that advertise the local time of a radio. The heartbeats occur approximately every 30 seconds. This process is called maintenance. To acquire neighboring radios, a burst of approximately 60 small acquisition packets (approximately 100 bytes each) is transmitted on random frequencies; each packet contains all of the information required for a radio to send a packet back to the originator. This burst takes about 600 ms and is repeated, with a few seconds between them, for 30 s to ensure that the radio modem learns of all its neighbors. This information includes relative timing, hopping sequence, the MAC address of the originating device, and the time that the originator will be listening for responses. After the acquisition is started with the broadcast acquisition packet, the radios talk directly to each other to acquire information needed by higher layers to become fully functioning members of the network, such as routing services and more advanced data link services (different modulations supported, for instance.) In addition, the radio modem uses 'third-party queries' to ask its neighbors for information on any of their neighbors. This way the radio modem can acquire radios that it doesn't happen to reach using the broadcast acquisition process.

After the radios have acquired each other, they then register with the Name Service. The registration process allows the Name Service to store the information required to route from the Name Server to the registering entity. Every device in the network must register. When any device wishes to find another device in the network, it looks up the other device by logical name in the Name Service. The Name Server can then construct a route between the two devices and return an MCDN path with enough information to allow packets to be routed between the two entities. Each device appends the MCDN path to any packet it wishes to route through the network.

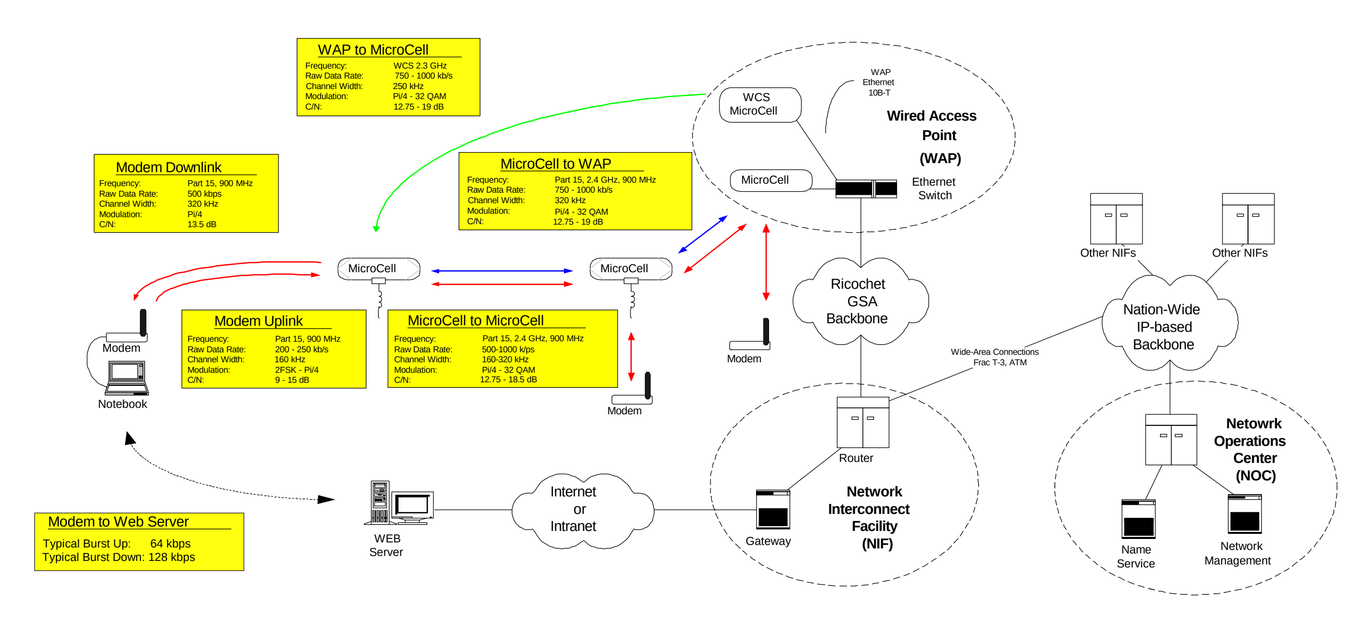

Figure 2. The MCDN (3G) Network Architecture (Physical).

An MCDN path consists of a list of waypoints, each waypoint is a TLV (Type, Length, Value) which has a one byte type identifying the routing system to be used, a one byte length, in words, of the address of the entity to route to, and the address stored in native format. The header of the path consists of a version, a bit to designate which end of the path is the source of the packet, a counter that designates which is the next entity to be routed to, and the total length of the path. When a node receives a packet it examines the next entity to be routed to. If that entity is itself, it then looks at the next address in the path and routes the packet towards that destination. If that address is not itself, it continues to forward the packet towards the original destination. In this way, the packet is eventually routed to the final destination.

The customer's radio modem typically appears as a standard AT command-based modem and is thus compatible with PPP and TCP/IP stacks supported on most computing devices. When the user attempts to connect, the computing device sends a connection request to the radio modem in the form of 'ATDT name', where name is the logical name of the device (such as a phone number, a modem name, or an MCDN gateway name.) The modem sends a LookUp packet to the Name Service asking for the MCDN path to the logical name, typically an MCDN gateway to the Internet or an intranet. When the MCDN path is returned, the radio modem uses this path to establish a virtual, end-to-end connection to the MCDN gateway. This connection can come in three flavors: 1) star mode, which provides a UDP-like service, best effort delivery of the packet to the endpoint; 2) the lightweight transport protocol, which provides in-order delivery of packets, and is a windowed protocol with flow-control, but does not attempt end-to-end retries; and 3) the heavyweight transport protocol, analogous to TCP/IP, which guarantees in-order delivery of every byte and includes flow-control and end-to-end retries. Typically, the best performance when running over the Ricochet system is to use the lightweight transport protocol; the heavyweight protocol usually results in excessive retries at multiple levels of the stack and uses up more bandwidth than required.

3. Modifications to the MCDN Architecture to Support Mobility

The entire system described above works well when the two endpoints are stationary as the MCDN path between the two devices does not change. However, when the devices are moving, the path can actually be different from one packet to the next. The MCDN architecture is designed to spread the intelligence required to support mobility throughout the system: Each piece of the system knows how to best handle its decision using local information only; this reduces the need for storage and state throughout the network. In the MCDN system there are three additional mechanisms that ensure packets get delivered to the right endpoints even while the devices are moving. We use 'best-node' insertion to guarantee that a packet comes back to the right place, forwarding to redirect packets in flight during a switch of microcells, and WAP redirection to make sure the packet gets to the backbone as quickly as possible. Finally, many of the other processes already described are modified to decrease the time they take while paying as little penalty as possible in performance.

3.1 Switching Between the *Static* and *Mobile* State

A subscriber device can be in two distinct states: ***static*** (not moving) or ***mobile*** (moving). There is a transition mechanism in place to switch from one state to the other by detecting if the subscriber device is currently moving or if it is static. Subscriber devices are by default in the *static* state. See Table 1 for more details as described below in the text.

During acquisition, each radio modem remembers all of the band independent information for all of the MCDN devices that it acquires. The eight microcell links acquired that provide the best signal strength are remembered; the other microcell links acquired are dropped. The radio modem also increments a counter, called the ***try counter*** for each link whenever there is a failed transmission and resets it to zero whenever there is a successful transmission. The radio modems also pick the two nodes with the strongest RSSI and stores them in an array called the ***strongest-node*** array. The mobile unit also

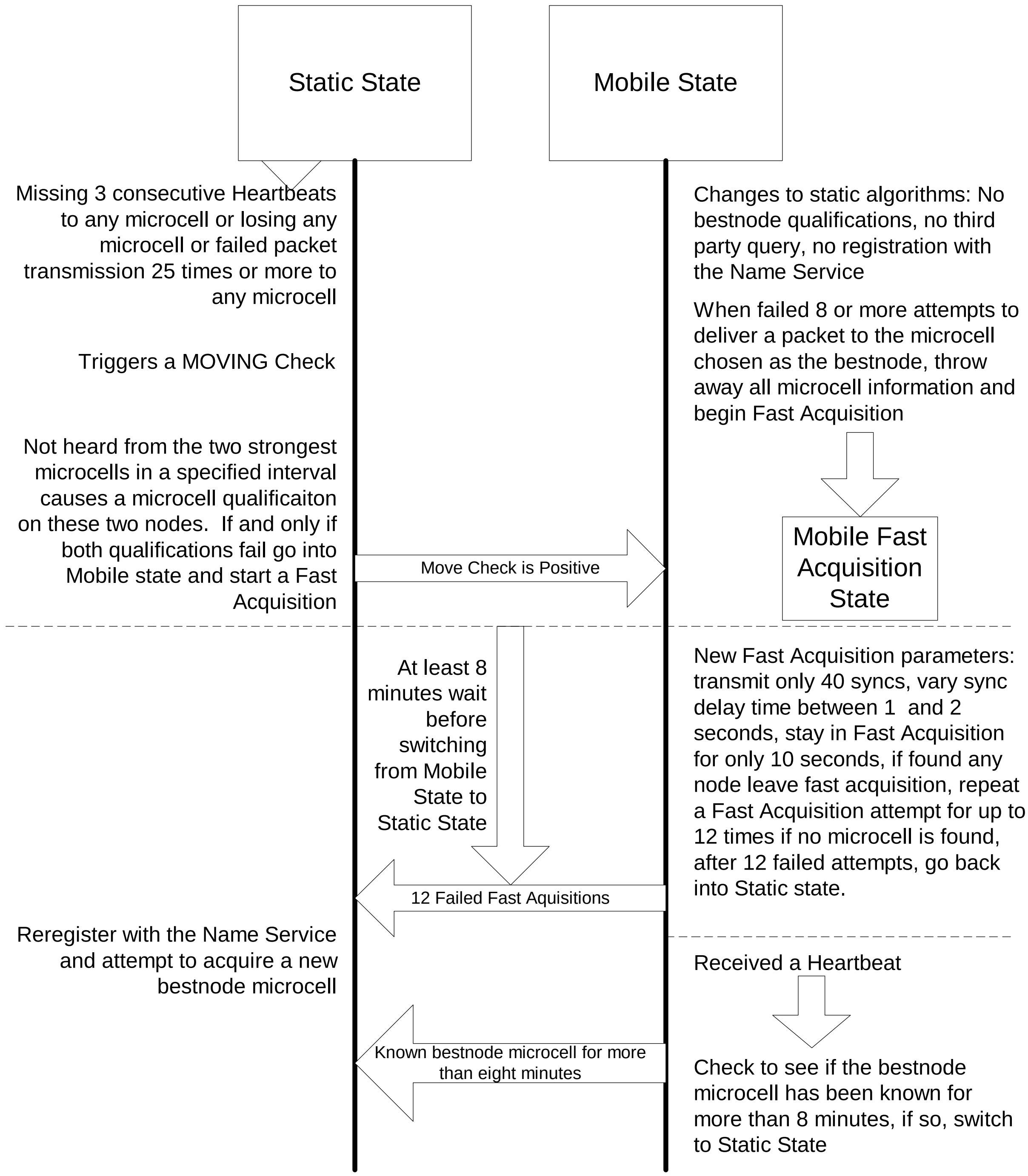

Table 1. Transition state diagram for modem with mobility enhancements

selects the best microcell link among the eight, designating that microcell as the 'best-node' and uses it to route all of its packets.

If a subscriber device in the *static* state fails 25 consecutive packets over any microcell link, or misses 3 consecutive heartbeats from any of those microcells, or if it loses any of its microcells (which can happen during the course of normal maintenance), it performs a test on the two strongest nodes to see if it is moving or not. If the subscriber device has heard from either of the two strongest nodes, i.e., a heartbeat or a packet, within a specific interval, depending upon the setting of the try counter, it will remain in the *static* state. If not, then the subscriber device will transition into the *mobile* state, remove all microcell links from memory, and begin a fast acquisition burst to acquire a new set of microcell links.

Whenever a heartbeat is received, a subscriber device will check to see if its best-node has been known for more than 8 minutes. If this is the case, it will switch back to the *static* state. Otherwise, it will stay in the *mobile* state.

3.2 Best-node Insertion

The first method of mobility tracking is called best-node insertion. The originating modem always places the address of the microcell that maximizes its throughput to the wired backbone, its 'best-node', as the return address of the path. When a modem chooses another best-node it changes the MCDN path on the next packet it sends out to direct the return packets back to its current best-node; thus, the gateway only needs to return the packet to its source to ensure delivery. In the *static* state, subscriber devices try to stick with their best-node as long as possible and go to great lengths to make sure that the measurements to determine a best-node are accurate. In the *mobile* state, the subscriber devices switch best-nodes whenever it determines a new node has a better score, with no regard for accuracy of the measurements.

3.3 Forwarding

However, even after best-node insertion, a packet in flight may reach an old best-node, which can no longer talk directly to the radio modem. Without any further modification to the protocol, the packet would eventually time out and be discarded. To ensure the delivery of the packet, the modem sends a forwarding address to its old best-node when it picks a new one. This forwarding address is kept in a cache, which slowly times out with a lifetime much longer than the typical in-flight time of a packet. The old best-node uses this address to forward to the new best-node all packets destined to that modem. This ensures that packets will get forwarded to the right best-node and reach the modem even if it has moved to a new best-node while the packet was in flight.

3.4 Wired Access Point Redirection

Finally, the MCDN path delineates which Wired Access Point the packet should use to enter the wired backbone. If the modem has moved, this may no longer be the best WAP to use. In early versions of the MCDN system, this WAP address was fixed in the path, thus as the modem moved throughout a city, the packet would always try to route itself back to the original WAP, even when a better WAP was closer. The routing system has been modified so this no longer happens. The microcell receiving the packet is continually evaluating the best WAPs available to use via the wireless routing system. When it receives a packet destined to a WAP, it determines if it has a better choice, if so, it modifies the path directly to take that into account and forwards the packet to the best WAP, regardless of what the path was originally.

3.5 Modifications of Existing Mechanisms

As the mobile unit moves faster it stays near any particular microcell for a shorter and shorter time, thus it has less time to determine the correct best-node or determine details of all the nearby microcells. To provide good connectivity, it should take advantage of any microcell it can find that provides adequate throughput. Thus, the mobile unit goes into the *mobility* state when it determines it is moving and changes the parameters of the various services. For instance, the mobile unit sends out more acquisition packets, does no third-party querying, does less maintenance, and accepts a worse link for a best-node.

Whenever the modem fails 8 consecutive packets to its best-node, which takes around 800 ms, it forgets all of the microcell information it has and begins a new fast acquisition process. The acquisition process is modified by increasing the number of acquisition packets sent out in a burst to 40 packets, with a variable delay of 1 to 2 s between bursts. It will continue like this for 15 seconds. However, the radio modem stops sending out acquisition bursts after it acquires at least one microcell that it can use to provide reasonable service. This causes the radio modem to use links that provide poorer throughput on average, but it reduces the time spent picking a best-node and causes the modem to switch to a new one faster, thus increasing the amount of time the modem spends talking to a good microcell. At 70 miles/hr a radio modem will spend only about 10-20 seconds in the range of a microcell, thus it is critical to use this time as efficiently as possible.

Whenever the radio modem changes its best-node, it must perform some actions to ensure that connectivity can be maintained. The modem will send forwarding packets to the old best-node, update its new path to the Gateway, and send a re-registration to the Name Server. However, during the mobile state, the modem will not re-register with the Name Service when it changes its best-node. The only way to force the modem to re-register is to restart the modem or to force a best-node qualification event, typically by pressing and holding a button on the modem to get a signal strength reading. This reduces the number of packets sent out during mobility, thus increasing the time that can be used to send data; however, this is at the cost of not being able to contact the modem until it goes back into the static state. Since the radio modem is almost certainly the originator of all calls, this penalty is small. If for some reason we are required to connect to moving radios, they could re-register on a time-period less than the clearing of the forwarding cache and packets could always be routed to the modem so that connections could be made.

4. Theoretical Model of Throughput

We will derive the average throughput of a mobile link by splitting the throughput, S_m, into two pieces that reduce the throughput, S, from that available on a perfect link: the first factor, R_p, due to lost packets and the second factor, R_l, due to additional latency at the link level due to retries.

$$S_m \propto S \times R_p(v) R_l(RSSI)$$

The rough model for R_p is that throughput loss from a perfect link is proportional to the square root of the percentage of lost packets [4]. With our acquisition model we can relate packet loss to velocity, as shown in the next section. The rough model for R_l is that as the signal strength decreases linearly, the throughput decreases exponentially as retries, and hence latency, increase exponentially. The average throughput of a link will be less in the *mobile* state as the radio modem does not spend as much time determining the best best-node, but just uses any microcell that is good enough, thus the average RSSI seen by a mobile node is less than that seen by a static mode. The FTP throughput, S, of 200 Kbytes files versus the RSSI (Received Signal Strength Indicator) has been empirically fit from field data to the following equation, linear in dB:

$$S(\text{kbits/s}) = 5.5 * \text{RSSI(dBm)} + 550,$$

where S is railed to 0 and 220 kbits/s at RSSI values of -110 to -60 dBm. When the radio modem is static the average FTP throughput is 176 kb/s, which implies that on average there is a microcell with an RSSI of around -70 dBm nearby the modem. This has been verified by measurements in the field. The distribution of RSSI at the modem from microcells in our typical deployment is spread at random around –80 dBm, when you allow the modem to pick the best microcell out of that distribution, the RSSI average increases to –70 dBm. However, when in the mobile state the modem does not get to choose the best microcell, but picks one at random, thus the distribution of RSSI it sees is the same as the distribution measured, with an average RSSI of approximately –80 dBm or an FTP throughput of around 14 kB/s while not moving (see 0 speed data points on figure 6.) This explains the difference between figure 4, taken while the modem is in the *static* state and the 0 velocity histogram of figure 7, where the modem is in the *mobile* state, but not actually moving. Thus there is a penalty to pay for going into the mobile state; however, the improvement at higher velocities more than makes up for this penalty. The average of the RSSI sensed by a modem is only dependent on the *state* (*mobile* or *static*) of the modem, in the static state the modem finds the best RSSI available, in the mobile state, the modem uses any microcell at random, thus has a lower RSSI on average. Since the average throughput is only dependent on state, R_l, is simply a factor, approximately, 0.7 (14 kB/s / 20 kB/s from figures 4 and 7.)

Without the *mobile* state, the average throughput of the modem while moving is reduced even more as it spends a significant amount of time making measurements and finding new microcells. Determining a new best-node can take 10-20 s, during which time the throughput of the modem is significantly reduced as it is busy making measurements of RSSI to determine the best-node and making third-party queries to determine what the network looks like around it. Since it may only be in contact with a node for 10-20 s, this causes a severe degradation in average throughput because of packet loss during these times. This can be seen by looking at figure 5, where it can be seen that without the modifications made in the *mobile* state, the modem transmits little or no data most of the time.

Assuming that a mobile modem can lose connectivity with a microcell at any random time, the model for R_p, is reasonable. We assume that the packet loss is proportional to the average time the modem spends acquiring new. These periods are long enough that the packets sent during this time, even if forwarded, are likely to exceed the retransmit timer and look to TCP as if they have been dropped. The percentage of time when packets appear to be dropped increases as the velocity of the modem increases since it spends less time within the range of any microcell. The following model tries to predict what the change in throughput as a function of velocity should be.

When a modem is in the mobile state and loses its best-node, it will attempt to acquire a new best-node by going into fast acquisition, as described above. The probability that a burst will result in at least one neighboring radio acquiring the mobile unit can be calculated by first determining what the probability of having a microcell hear a particular sync packet, P_h, is

$$P_h = P_c * P_l * (1-P_n),$$

where P_c is the probability that a node is listening on a particular channel and is equal to one over the number of channels (~0.01), P_l is the probability that a node is listening (approximately 80% under typical load), and P_n is the probability that the sync packet is lost due to noise, approximately 20%. Then, the probability that at least one node gets a sync packet, P_A, is a function of the number of nodes a modem can see on average and the number of sync packets transmitted. If we send n synch packets and we see m nodes, then we can write the probability that a particular node sees at least one synch packet, P_s, as

$$P_s = 1 - (1-P_h)^n.$$

And thus we can write the probability that at least one node of m nodes hears at least one (of n) sync packet, P_A, as

$$P_A = 1 - (1-P_s)^m.$$

On average a mobile unit can hear 5 reasonable neighbors at any time; thus P_A ~ 60% for each burst. Each burst takes approximately 400 ms to send and the radio then waits for about 1500 ms to receive an answer. Thus, a microcell is usually acquired after the first burst. However, the average amount of time to acquire at least one radio, T_A, is approximately,

$$T_A = \sum_{n=1}^{\infty} (T_B + (n-1)T_S) P_A (1-P_A)^{n-1}$$

where T_B is the time spent sending acquisition packets, and T_S is the time from the beginning of one burst to the beginning of the next burst, approximately 1.9 s. T_A is approximately 3.4 s. It takes the modem approximately 800 ms to determine that is has lost the current best-node and should find another one. Thus, on average it takes the radio modem 4.2 s after it begins to fail to its best-node before it can begin receiving traffic from another best-node. These values have been reduced significantly in new versions of the radio code to improve the performance further. A radio modem when traveling through a mesh cell of networks spends T_G seconds using its best-node before it goes out of range,

$$T_G = d / (v \times \sqrt{2} / 2),$$

where v is the velocity or speed or the radio modem, d is the average range of a microcell where it will fail 8 consecutive packets, approximately ½ mile, and the numerical factor comes from the fact that the modem is traveling in a random direction from the microcell. Thus, the modem will be out of communication with the network P_c percent of the time, where,

$$P_c = T_A / (T_G + T_A).$$

We can derive the throughput in this case by assuming that P_s is small and randomly distributed over the TCP transfer as for small percentages of dropped packets the TCP performance is proportional to 1 – square root of the percentage of dropped packets, which is proportional to P_c; thus, S, the throughput as a function of velocity is shown in figure 3 and can be written as

$$S(v) \propto S(0)\left(1 - \sqrt{\frac{T_A}{\frac{d\sqrt{2}}{v} + T_A}}\right).$$

This prediction can be compared to the throughput distribution measured and shown in figure 6. The faster the mobile unit is moving the more time it spends acquiring and the less time it has to provide throughput, thus throughput decreases as the mobile units speed increases.

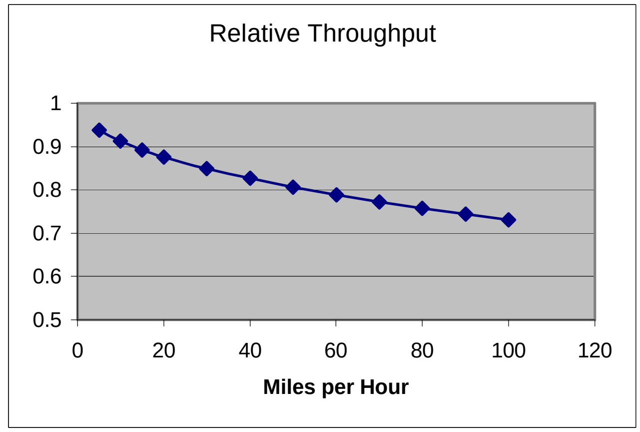

Figure 3. Predicted relative throughput as a function of speed.

5. Field Measurements in the Ricochet Network

We conducted several experiments to evaluate the performance of our modifications to the mobile acquisition protocol compared to the non-modified protocol. Our aim was to compare the performance of mobile to non-mobile users. Furthermore, we wanted to establish the gain in performance by modifying the mobile acquisition protocol in the subscriber devices. We first

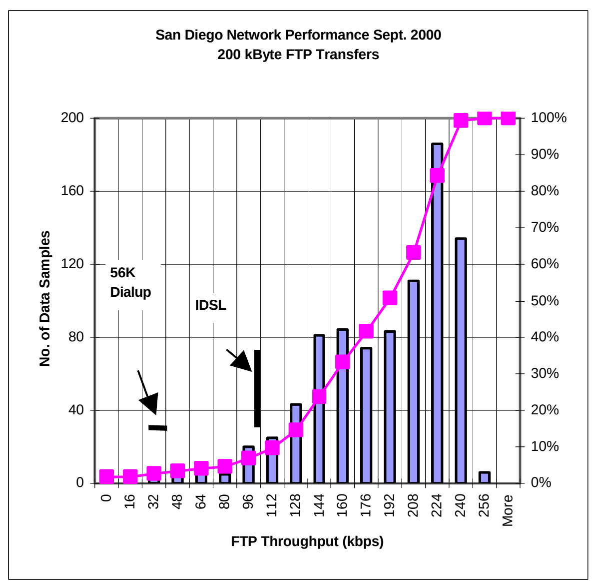

Figure 4. Throughput distribution while the mobile is in the static state. It can be seen that picking the best-node carefully instead of latching on to the first microcell acquired causes the performance to increase. In the static case the 80% performance point is 128 kbps, whereas in the mobile case the 80% case is 28.8 kbps.

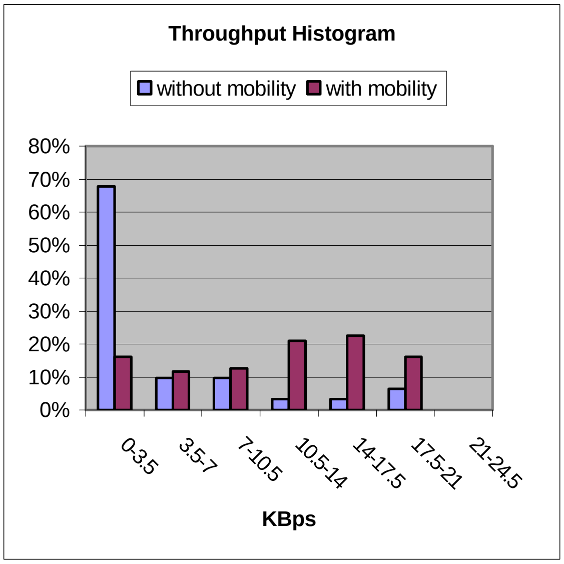

Figure 5. Without the mobility state, the modem spends too much time running the best-node algorithms as well as doing third party querying, thus severely degrading the performance of the network. This test is performed in the same manner as the one in figure 3.

describe the test setup used to conduct the experiments, than we discuss the results obtained.

5.1 Experimental Setup

To establish the performance of a mobile user in the Ricochet Network, with and without the protocol modifications, a driving test trail was carefully selected. A portion of the trail was on the highway, allowing us to reach speeds up to 70 mph. The rest of the path was on standard urban streets. The duration of the test depended on the number of loops around the trail, each taking around 1 hour. We used a laptop running Windows 98 with a Ricochet GS Modem connected to the laptop through the USB port and a GPS system connected to the serial port to keep track of the current position. Every 2 minutes the modem initiates an FTP session and downloads a 200 KB file from the server, recording the FTP download rate and terminating the FTP session. Simultaneously the GPS system kept track of the current position at the beginning and at the end of each FTP session. This allowed us to estimate the speed at which we were traveling.

5.2 Performance Results for non-mobile users

Initially we ran a separate experiment to base-line the performance of the network. A 200 KB file was down loaded at random spots in the network and the FTP value was recorded. A statistically significant number of samples were collected (on the order of 2000 points). The modem used for this experiment did not have any modification for mobility. Figure 4 shows the FTP histogram obtained for this experiment. Notice, that if a user is not moving 80% of the points are above 128 kbps.

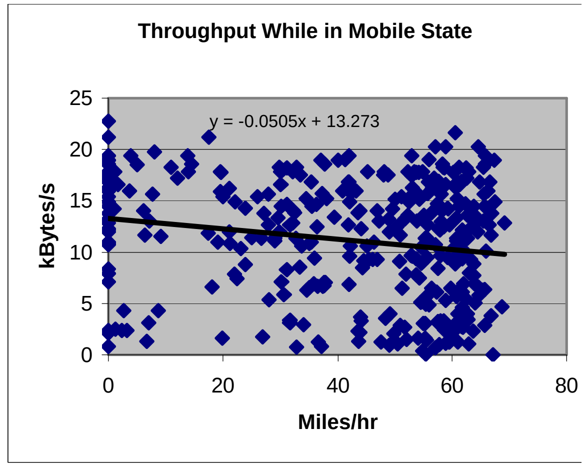

Figure 6. A scatter plot of FTP throughput versus speed. The data was taken by driving through an area covered by microcells and continuously attempting to transfer 200 kB files using FTP. The equation is the best linear fit to the data.

5.3 Mobility Test

To establish the performance gains obtain by modifying the code, we collected FTP values while moving with a modem with no modifications and one with the mobility modifications. In Figure 5 the histogram for both collected data are plotted. The performance of the modem with the mobility mofications outperforms the results obtained by the non-modified code. In the case of the non-modified code, many of the FTP values are cluster in the 0 KB bucket. The reason for the low performance of the non-modified modem was that it failed to acquire a new best-node fast enough to complete the FTP transfer and in many cases the connection was lost. The modified modem was able to acquire a new best-node much faster and complete the majority of its transfers. Even though the best-node selected by the modem was not the optimum node, it was sufficient to transfer data at acceptable rates and to keep the FTP transfer going until it finished.

Changing the acquisition protocol significantly enhanced the performance while mobile. If we compare mobile users with the acquisition-modified protocol with non-mobile users, we observe that our 80% throughput point is now approximately 28.8 kbps range as shown in figure 5 and figure 7. While mobile users will experience performance degradation as compared to static users, the degradation of mobile users with the mobility modifications is much less than without the mobility modifications, as shown in figure 5.

In figure 6 each FTP value collected during the mobility test with the modified code is associated with a speed value. The equation inside the plot is the best linear fit for the experimental data. The agreement in direction and magnitude with the theoretical curve

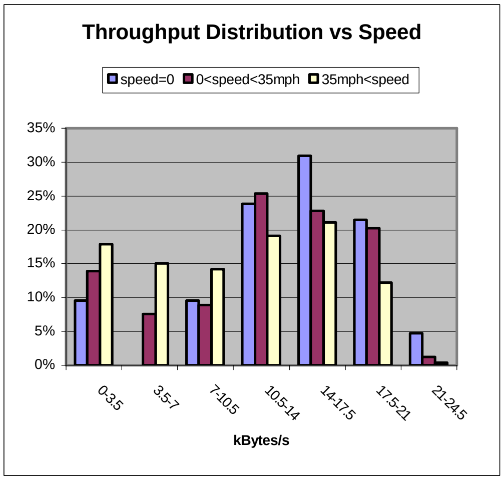

Figure 7. The throughput as a function of speed with the modem in the mobile state. Throughput is measured by continuously sending 200 kB files using FTP from a windows 98 client to a Sun Server placed at a NIF.

shown in figure 3 shows that the model proposed covers the gross features of a mobile modem..

The data points in figure 6 are plotted in a histogram in figure 7. We group the data into three groups according to speed; 1) speed = 0 mph, 2) speed greater than 0 mph and smaller than 35 mph, and 3) speeds higher than 35 mph. The three groups represent typical speed for the non-moving users, urban speeds, and highway speeds. The FTP histogram indicates that speed does have an effect on the performance as predicted by the theoretical model. Notice that the average of the histogram shifts to the left as speed increases as the model predicts.

6. Conclusion

This paper has presented a wireless packet-based microcell architecture, concentrating on providing the motivation for and describing those pieces that support mobility. It is noted that the most difficult problem to solve to provide for high-speed mobility is not routing, but actually acquiring microcells to route through. A compromise has been made to decrease the time required to acquire microcells rather than trying to find the best microcell, this results in a tripling of the average throughput for a mobile node in the network (from 4.7 kB/s to 11.2 kB/s.) In addition, it raises the point that 80% or more of the throughput measurements exceed from approximately 1 kB/s without the modifications to 4.6 kB/s with the modifications.

This paper points out that it can be more important to decrease the time required to acquire neighboring nodes in a highly mobile system rather than ensure that the ideal route is propagated through the system. The authors would be very interested in further work on wireless acquisition, which we think will be crucial to making high-speed mobile ad-hoc networks usable, especially those using unlicensed spectrum. This problem has many different solutions, but finding one that minimizes the bandwidth required for this network service while still keeping the acquisition times reasonable is a difficult challenge.

7. REFERENCES

[1] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva,. "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", in Proc. of the ACM/IEEE MobiCom, October 1998.

[2] R. Kalden, I. Meirick, and M. Meyer, "Wireless Internet Access based on GPRS", IEEE Personal Communications, April 2000, p. 8.

[3] Y.B. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom'98), October 1998.

[4] J. Padhye et. al., "Modeling TCP Throughput: A simple Model and its Empirical Validation", Pro. SIGCOMM'98, AM 1998.

[5] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In Proc. of the ACM SIGCOMM, October 1994.

[6] C. Perkins, " IP Mobility Support for IPv4, revised", draft-ietf-mobileip-rfc2002-bis-03.txt ,RFC 2002, 20 September 2000 [RFC2661].

[7] M.W. Ritter, R.J. Friday, M. Cunningham. The Architecture of Metricom's MicroCellular Data Network? (MCDN) and Details of its Implementation as the Second and Third Generation Ricochet? Wide-Area Mobile Data Service. In Networld + Interop 2001 Broadband Wireless Summit Conference Proceedings, May 2001

[8] W.M. Townsley, A. Valencia, A. Rubens, G. Singh Pall, G. Zorn, B. Palter, "Layer Two Tunneling Protocol (L2TP)", RFC 2661, August 1999.

[9] FCC Rules and Regulations, Part 15, October 1997