User Devices Cooperating to Support Resource Aggregation

Casey Carter Robin Kravets
Department of Computer Science
University of Illinois at Urbana-Champaign
E-mail: `{ccarter,rhk}@uiuc.edu`

Abstract

MOPED (MObile grouPEd Device) is a network model that treats a user's set of personal devices as a single, virtual device. The nodes of the MOPED dynamically aggregate available communication resources to provide the user with the best possible network service. We demonstrate MOPED's resource management capability with a short series of experiments that show how the MOPED paradigm enables effective group mobility.

1. Introduction

Increases in availability and capability of computation and communication technologies have resulted in an embarrassment of riches: a plethora of personal devices ranging from pagers and cell phones to PDAs and webpads, and from familiar devices like laptop and desktop computers to more exotic wearable computers and Web-enabled watches. Many of these devices incorporate some form of wireless communication.

As users collect multiple small computing devices, the available communication resources increase, as does the demand for coordination of these resources. The utility of these devices has been limited by their isolation from each other – each device must be a self-sufficient computational and communications entity. A user is incapable of taking advantage of a new, faster wireless technology without upgrading each of the devices individually.

To help users control and coordinate their constantly evolving set of mobile computing devices, we present the MOPED (MObile grouPEd Device) paradigm. A MOPED is a set of devices that present themselves as a single device. To the rest of the Internet, the MOPED appears to be a normal host. A fixed host within the Internet performs as a proxy for the MOPED, directing individual traffic flows to and from the appropriate device within the MOPED. As they move through the Internet, devices from the same MOPED dynamically form a MOPED *component* when they come into contact with each other, allowing them to share communication resources.

The goal of the MOPED project is to provide service to a user through the cooperation of the devices that is better than the service provided by the devices working individually. Our solution provides three key benefits. First, a user can reach the network via any of the services currently available to the devices. Second, if multiple devices have connectivity, the MOPED can take advantage of the additional resources. Finally, such connectivity smooths handoffs as individual devices gain and lose connectivity, insulating the user from instability.

Previous work focused on the MOPED Routing Architecture (MRA), the basic architecture that supports MOPEDs and enables them to move through the Internet as distributed mobile networks [5]. The MRA enables localized cooperation of devices through the concept of a MOPED component, a locally connected subset of the user's devices. As part of a component, a node's resources are added to the pool of resources available to the component. In this paper we investigate the adaptability of a MOPED to its changing environment. Discussion of the implementation of the MOPED system and some experimental studies of resource discovery and aggregation will serve to illuminate MOPED's ability to assimilate network service of many devices into a coherent whole.

In Section 2, we discuss the motivation behind the MOPED concept. We provide a brief introduction to MOPEDs and MOPED routing in Section 3, to allow the reader to appreciate the results presented in Section 6. Section 4 contains a description of the MOPED self-assembly process, by which a MOPED discovers available resources. We give a detailed description of our implementation of MOPEDs in Section 5. Conclusions are drawn in Section 7, discussing the results and laying out plans for future MOPED work.

This work was supported in part by the National Science Foundation under grant ANI-0081308, and by Hewlett Packard Labs.

2. Motivation & related work

"Convergence" is a watchword for the communications industry: the slow process of merging many pieces of personal technology into a large amalgamated device. The result of this process will be an unwieldy device that supports dozens of communication technologies and is a mediocre pager, cell phone, PDA, laptop, palmtop, scanner and fax. We believe that convergence in the communications industry will have the same effect as convergence in the printer, scanner and fax industries: the creation of devices that are dabblers in many trades, but masters of none. The misconception here is not a new one, but is the familiar assumption that one solution can be all things to all users.

In contrast to the approach of convergence, we embrace diversity in personal technology. Users should have the privilege of picking and choosing from many well-focused, specialized pieces of personal technology to form their interface to cyberspace in a modular way. We refer to this concept, and the set of devices thus formed, as a MObile grouPEd Device (MOPED). MOPED technology forms a fundamental network layer for this set of devices, a common system enabling the devices to coordinate and share resources to provide the user with mobile network service.

Rapid deployment of many new wireless technologies has produced a proliferation of wireless connectivity, with overlapping coverage in some areas. Such diversity in communications is the meat and potatoes of MOPEDs, allowing the aggregation of many communication resources to provide the user with enhanced throughput and wider coverage than that achievable by any single technology. All devices in a MOPED can enjoy the advantages of a new wireless technology simply by integrating a single device supporting that technology into the MOPED.

Other work in mobile networking has addressed user mobility by enabling a user to carry a single mobile device with them through the Internet; many current solutions focus on a single host with one network interface [1, 10, 11]. We believe that the appropriate next step is mobility management for a MOPED, the network of devices associated with one person. All communication traffic for a MOPED user is delivered to the MOPED, where the final disposition of traffic is determined. Since a MOPED is designed to support a single user, communication with any of the devices in the MOPED is considered successful communication with the user. This model enables a group of devices to be mapped into a user's point of presence on the Internet. To the outside world, a MOPED appears as a single device with a single interface or identifier. In reality, the group of devices cooperates to provide the illusion of unity.

In this section, we discuss the factors that motivate the MOPED design, and how they foster the MOPED's ability to aggregate resources. The three key factors are: support of a single point-of-presence for a user, infrastructural support for an overlay network, and mobility of a network with multiple points of Internet attachment. The combination of these factors uniquely positions MOPED as a tool for user management of communication resources.

2.1. User location

We believe that a user should be able to create a representative presence on the Internet. If the presence has a unique network name, a single IP address, the user is built into the network infrastructure, replacing the problem of user location with a more traditional network location problem. This conversion of person-location into network-location allows our solution to interoperate with legacy applications that are not MOPED-aware.

The user-location aspect of MOPEDs is similar to the goal of Stanford's Mobile People Project [9]. Mobile People is an architecture for allowing application-level mobility: it provides a name service to resolve user names to application-specific addresses for a user, a process Mobile People calls "person-level routing." Mobile People provides an intermediary between communicating parties where they may record their current address and learn others' addresses. Both Mobile People and MOPED provide location privacy; they make it possible to communicate through a proxy, hiding the user's actual location. Mobile People does not address the grouping of several devices into a single logical entity.

If a user has only a single device with a single interface (e.g., a cell phone, laptop), the device can be supported via existing techniques such as cellular telephony or MobileIP [11]. If the device has multiple interfaces, each interface can be used as available [13] or simultaneously [15]. If the user wishes to use several devices with many interfaces, the only available solution is to force the correspondent to choose the destination, forcing the correspondent to be aware of the complexity of a user's suite of devices. In a MOPED, correspondent hosts need not, and in fact should not, be aware of how the devices compose to form a user presence.

2.2. Network infrastructure

A MOPED is more than a PAN. Technologies for personal area networks (PANs) such as Bluetooth and low-power IEEE 802.11 have come into vogue in the networking research community, but they are simply mechanisms for physical connectivity among a set of devices. We see MOPEDs as a network-layer entity that is complementary to the datalink-layer concept of a PAN. Although a PAN is useful, it is not necessary to our design; indeed, we allow separate components of connected devices to participate in the same MOPED using external channels.

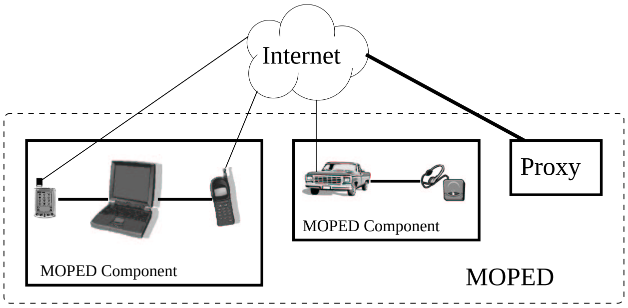

Figure 1. A MOPED with two components and five devices

Some recent work has focused on networks of devices associated with a router that changes its point of attachment to the Internet, a Mobile Router [2, 4]. Although clearly related, a MOPED component differs from a mobile-router network in that a MOPED is a multihop ad hoc network with multiple points of Internet attachment. In the spectrum of network mobility, this places a MOPED component somewhere between a MONET (MObile NETwork) and a MANET (Mobile Ad hoc NETwork). Furthermore, the MOPED is cooperatively supporting the needs of a single user, so some problems of resource utilization may be solved differently, i.e., such as by sacrificing one device to preserve others more important to the user.

2.3. Mobility

A MOPED is a composite of many devices with many network interfaces; a mobile network with multiple points of attachment to the Internet. A single MOPED device might itself have multiple external interfaces. MOPED is thus a mobility solution for one or more devices with zero or more Internet-mobile network interfaces each.

The goal of MobileIP is to make it appear that a mobile host is not mobile, but is static located at “home” in the Internet. Although MobileIP solves the problem of mobility for a single node and interface [11], the multiple-interface, single address nature of MOPED mobility does not align well with MobileIP. MOPEDs must multiplex traffic destined for many devices onto a single IP address. MOPEDs may contain devices without direct Internet connection, and thus cannot use MobileIP.

We will, however, use MobileIP as an internal component, providing mobility (relative to the MOPED proxy) to each individual interface on a MOPED device that can be used for external connectivity. MobileIP is ideally suited to this role in the MOPED architecture and allows a MOPED to take advantage of any deployed MobileIP infrastructure.

2.4. Resource discovery and aggregation

Users with several devices with diverse network resources would like the devices to share resources to achieve better service and wider coverage. Such resource-based cooperative networking is a benefit of MOPED technology.

Consider a MOPED component comprised of a PDA with a cellular modem, a cell phone, and a laptop with no connectivity, all connected by a PAN (the left component in Figure 1). While the user is talking on the phone, traffic for the laptop can be routed through the PDA. To support this, the MOPED proxy routes flows for different endpoints within the MOPED to the appropriate external interface. If the user is participating in a videoconference on the laptop, the audio could be routed through the phone, while the video is routed through the cellular modem, providing more bandwidth to the application than had a single interface been used. Since the endpoint of the communication is the laptop, traditional routing support cannot route the separate flows to the same endpoint along different paths.

We believe that MOPED should support bandwidth aggregation with flow-level granularity, i.e., all packets in a particular flow – identified by IP protocol, and source & destination IP addresses and transport-layer port numbers – should follow the same path. A finer level of granularity, directing packets from the same flow to follow different paths, could cause severe packet reordering which transport layers such as TCP will interpret as loss [12].

If an application has requirements for a single flow which are greater than the bandwidth provided by any of the external connections individually, specialized transport protocols [6, 7, 8] can utilize all available bandwidth by inverse multiplexing the flow across multiple paths. MOPED provides flow-based and packet-based routing to provide transport layer protocols the ability to realize these potential bandwidth improvements.

Aggregation of multiple channels has benefits beyond the obvious enhancement of bandwidth. By using multiple or all available channels simultaneously, the MOPED is freed from the necessity of probing and measuring individual paths and trying to determine which interface is best. Diversity of communication channels also enhances reliability, by making the MOPED robust against the potential failure of a single channel.

When establishing a new connection, a user would like for the system to route that connection over the best interface for that connection’s requirements. What “best” means in this case can be very complex, and unpredictable. The best interface for that connection may vary over time as a result of factors beyond the user’s control, or the system’s knowledge. Often there is no way to prejudge the best interface for a given connection, and so a solution that avoids even the necessity of choosing is ideal.

3. MOPED design

The MOPED Routing Architecture (MRA) is a framework that enables a collection of devices to appear as a virtual distributed device overlaid on the Internet. The nodes that form the MOPED are assisted by a proxy that is fixed within the Internet. The MRA routes packets between nodes of a MOPED and, via the proxy, to correspondent hosts on the Internet, which may not be MOPED-aware or even mobility-aware. The MOPED Routing Architecture is presented in more detail in earlier work [5].

In MOPED nomenclature, a node that can communicate directly with the Internet is termed a *perimeter node*, and a node that can communicate with the Internet only indirectly through others is an *internal node*. A set of nodes that can reach each other using paths that pass only through internal interfaces is a component (from "connected component"). We expect a MOPED frequently to be partitioned into multiple components, and still continue to function normally. An example MOPED is depicted in Figure 1. In the component on the left in Figure 1, the cell phone and PDA are perimeter nodes, and the laptop is an internal node. Due to the potential of multiple external interfaces for each component, there may be multiple paths between each component and the proxy, although only one external interface is necessary to support connectivity to a component.

3.1. MOPED routing

A MOPED has a single official IP address by which it may be reached, and must have a supporting proxy to direct traffic to individual nodes from the home network. From the perspective of a node, traffic for other nodes in its component (internal traffic) and traffic destined outside the component (external traffic) must be handled differently.

There are three aspects of MOPED that prohibit the use of traditional IP routing: addressability, the necessity of addressing each internal device individually although the MOPED itself has only one public IP address; mobility, managing the mobility of the MOPED devices relative to the proxy and each other; and path selection, the ability to selectively utilize many paths from a node to the proxy to enhance throughput and reliability.

Mobility There are two types of mobility we consider in a MOPED: mobility of components with respect to the proxy, i.e., mobility through the Internet proper, and mobility of nodes within a component. Mobility of components through the Internet is well addressed by MobileIP, if each external interface is assigned a home address. Devices with external connectivity use MobileIP to establish a channel to the proxy, forming a link between the node's component and the proxy in the overlay network. Within a component, nodes locate their peers with a simple ad hoc routing protocol; this allows devices to dynamically assemble a component (see Section 4).

Addressability To deliver incoming packets to the proper endpoint in the MOPED, the proxy must maintain a mapping between the public IP address that identifies the MOPED and the internal IP addresses that identify the nodes. Network Address and Port Translation (NAPT), although traditionally used as a solution to the problem of address space pressure in IPv4, in a MOPED solves the problem of addressing specific nodes. In the MOPED Routing Architecture, NAPT multiplexes the entire MOPED on to a single address.

The MRA assigns to each device a static, private IP address in the MOPED overlay network – a node identifier. These addresses need not have significance outside of the MOPED, except as targets for NAT within the proxy. The NAT layer's sole responsibility is to maintain a mapping from correspondent host IP and port number to internal IP.

Path selection The final task necessary for the MRA is path selection. We believe that the bottleneck for communication resources in MOPEDs will be the hop from the perimeter nodes to the infrastructure, and not in the MOPED-internal links; lower bandwidth technologies usually have larger coverage areas than high bandwidth technologies. To enable cooperative resource utilization, traffic flows in the MOPED must load balance in a reasonable way across the interfaces on the perimeter of each component. The path selection mechanism enables a node to intelligently schedule packets through particular perimeter interfaces on their route to the proxy.

The MOPED's utilization of multiple communication channels is at a higher level than traditional cellular handoffs. Via the path selection mechanism, a MOPED uses multiple channels simultaneously to carry different traffic. This is distinctly unlike the failover characteristics of vertical handoffs [13] or the MobileIP error-robustness technique of simultaneous mobility bindings [11].

3.2. Address hierarchy

The MOPED Routing Architecture uses four distinct kinds of network addresses:

1. The MOPED IP address. This is the official, public IP address used to identify the MOPED, and consequently its owner.
2. Internal IP addresses. These addresses are used to identify particular nodes or their interfaces; they have meaning only within the MOPED and its proxy.

3. Perimeter IP addresses. These are the MobileIP home addresses of the external interfaces on the MOPED devices.
4. Care-of-Addresses. These are the IP addresses to which the MOPED external interfaces are currently bound by MobileIP.

3.3. Routing through the MOPED

Intuitively, when a packet arrives from a correspondent host addressed to the MOPED, the proxy must determine:

1. To which node the packet should be delivered—an internal IP address.
2. Through which external interface the packet must be routed to reach the target node's component.
3. Exactly where in the Internet that external interface is.

Upon arrival at that external interface, the receiving perimeter node then uses the internal routing protocol to deliver the packet to the final destination node. The destination node may then record the path taken by the packet, to ensure that return packets follow the same path.

Conversely, when a node transmits a packet to some other host, it must determine if the target is a node in this component; if so, it uses the internal routing protocol to deliver the packet. Otherwise, the target is in another component, or is simply outside of the MOPED—the packet is redirected to the proxy through a selected perimeter node and external interface on that node. At the proxy, the process used for delivery to the MOPED is reversed: the proxy mangles the source address in the packet, so that it appears to come from the official, public MOPED address, and traditional IP routing delivers the packet to the target host.

4. MOPED assembly

The abilities of a MOPED to route internally, aggregate external communication resources, and determine the structure of its own component are closely interrelated. Nodes participate in discovery and an ad hoc routing process that we term *assembly*. Assembly is a two-level hierarchical system that enables the nodes to form components and inform the proxy of the components' composition and location in the Internet.

At system start-up time, a node has no knowledge of its environment or any neighboring nodes. The dynamic nature of a MOPED makes static configuration of such information prohibitive, as it would inhibit the MOPED's usability and preclude the incorporation of MOPED technology into a consumer oriented device. The node is not even aware which of its network interfaces are to be used for internal, and which for external communication. The process of assembling such a collection of isolated devices into a coherent component should be as quick as possible, and require no static configuration (or as little as is feasible).

The dichotomy of internal/external interfaces, although a useful tool for exposition, does not truly exist in the MOPED system – every interface is treated simultaneously as both an internal and external interface. Two separate processes manage the external/internal aspects of an interface so as not to interfere with each other. The "external" interface manager's job is simply to seek out networks to which the interface can attach. Such external connectivity to a network is made possible by the presence of MobileIP foreign agents or the availability of DHCP services that the manager can use to assign a local address to the interface so that it may connect to the network infrastructure, and thus to the proxy. The task assigned the "internal" interface manager is much more complicated.

The internal manager is responsible for finding other nodes participating in the same MOPED and associating them into a component – the first level of the assembly hierarchy. Periodic beacons multicast on the interface inform a node's neighbors of its presence; each node assembles a neighbor list for each of its interfaces. The neighbor lists are subject to continuing maintenance and are used as input to an ad hoc routing protocol.

The ad hoc routing protocol is a special-purpose link-state protocol for MOPED components. Link state protocols distribute complete global information about the component, allowing any node to make optimal decisions about the network state within its component. The high overheads often associated with proactive protocols, especially full-topology link state protocols, are not a problem in the MOPED environment due to the small expected size of a component (on the order of 4-6 nodes). To enable the global dissemination of extensive link, interface, and node state throughout the MOPED, a link state routing protocol is the most viable choice.

Each node assembles a link state packet (LSP) containing the list of its neighboring nodes, and a list of its external interfaces. Flooding of these LSPs throughout the component informs each node of the component's topology and composition, allowing a node to locally make effective routing decisions. The LSPs are maintained by the neighbor discovery process, and re-flooded when the neighbor list – or set of external interfaces – changes. We again make the point that the frequency and volume of LSP updates is negligible for an ad hoc network consisting of only a handful of devices. The flooding process also enables the nodes to participate in an election process to determine a *component leader*. The component leader represents the component in the second level of the MOPED routing hierarchy.

The leader of each component transmits an abbreviated version of its LSP database to the proxy. This message contains only the node IDs that compose the component, and the set of external interfaces associated with each node. With this information, the proxy can determine how to route to each node in the MOPED and when route bindings must be changed due to mobility of devices and component partitioning. Future extensions to the second-level routing protocol will enable the proxy to disseminate the components' LSP databases to each other, enabling optimized component-to-component routing.

5. Implementation

The MRA is implemented atop the Linux 2.4 kernel, running on our MOPED testbed described in Section 6.1. We use the netfilter NAT functionality built-in to Linux as the MRA's NAT layer. The modularity of the architecture allows us to easily combine this component with a MOPED-specific routing agent. The routing agent is realized as a single user-space process – the MOPED Routing Daemon (MRD) at the nodes, and the MOPED Proxy Daemon (MPD) at the proxy. Both daemons are user-space applications that use the Linux kernel's Universal Tunnel driver to intercept packets from and inject packets directly into the kernel network stack.

To realize the complicated MOPED address structure, three network prefixes are necessary:

1. The public IP address that identifies the particular MOPED. (In this case, `192.17.240.48`).
2. The perimeter IP subnet – the set of addresses that contains the home address of each external interface. Our testbed uses the prefix `192.17.240.48/29`. This subnet is large enough to allow for 7 external interfaces, since broadcast functionality is not used.
3. The MOPED-internal IP subnet – the set of addresses from which MOPED internal addresses and node IDs are drawn. Since these addresses are only used within the MOPED and its proxy, this can be any non-allocated IP subnet. We use `10.0.0.0/16` for the example.

The current implementation structures the internal address space so that node number `x` has node ID `10.0.x.0` and internal interfaces numbered from `10.0.x.1 - 10.0.x.255`. This address space structure is convenient in that it allows a node to instantly allocate an internal interface address when it discovers a new interface (such as when a PCMCIA card is inserted). The node ID, network prefixes and proxy address are the only pieces of configuration that must be user-supplied.

5.1. Flow-based routing

Both routing agents participate in flow-based routing; a common module manages flow-specific routes. Specifically, the routing agents manage a route table, whose entries can match any subset of the attributes of a flow-specific route tuple (internal IP, internal port, external IP, external port, protocol). These flow-route specifications are used on demand to generate fully-qualified routes (with no wildcards) for the route cache. Packets which do not match any entries in the route cache or routing table are passed to the multipath routing policy algorithm, which chooses an appropriate path for the flow (based on user-configured routing policy) and inserts that path into the route cache.

Since all route cache entries are fully qualified, route lookups that hit in the cache require only a single hash table lookup. Route cache entries expire if unused for a some period of time, currently empirically chosen as 3 minutes. The stability of these cache entries ensures that packets from the same transport-layer flow follow the same path, and do not experience cross-channel reordering.

5.2. MOPED proxy daemon

The MPD is responsible for:

1. transparently translating the MOPED public address to a specific internal node's ID,
2. intercepting traffic to MOPED internal addresses,
3. encapsulating the packet with a routing header and
4. directing the packet toward the home address of an external interface on the target node's component.

Since the Linux kernel itself handles address translation, the MPD must create a set of netfilter rules that affect translation of MOPED-internal addresses to/from MOPED public address for incoming and outgoing traffic. Similar firewall port-forwarding rules allow nodes to "publish" ports that can be reached by correspondent hosts who initiate connections to the MOPED. An example of a MOPED firewall rule set is shown in Figure 2. Rules are structured so as to interfere as little as possible with non-MOPED network traffic. The proxy creates two chains at startup time, the `mopdvrt` and `mopsnat` chains, which apply to traffic incoming to and leaving from the MOPED, respectively. For each MOPED that a proxy supports, it

- creates an entry on the `mopsnat` chain that instructs the kernel to NAT that MOPED's internal address space onto its public address, and
- creates an entry in the `mopdvrt` chain that passes traffic destined for that MOPED to a new, MOPED-specific chain (`mop0` in the example). The MOPED-specific chain is populated with rules for each service exported from the MOPED.

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
mopdvrt    all  --  0.0.0.0/0            0.0.0.0/0

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
mopsnat    all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
mopdvrt    all  --  0.0.0.0/0            0.0.0.0/0

Chain mop0 (1 references)
target     prot opt source               destination
DNAT       tcp  --  0.0.0.0/0            0.0.0.0/0
          tcp dpt:22 to:10.0.3.0:22

Chain mopdvrt (2 references)
target     prot opt source               destination
mop0       all  --  0.0.0.0/0            192.17.240.48

Chain mopsnat (1 references)
target     prot opt source               destination
SNAT       all  --  10.0.0.0/16          0.0.0.0/0
          to:192.17.240.48
```

Figure 2. Sample `netfilter` **rules for a MOPED; output from** `iptables -t nat -L -n`**. The MOPED is running an** `ssh` **server on TCP port 22**

The use of separate `mopdvrt` and `mopsnat` chains minimizes the number of rules in the global `PREROUTING`, `POSTROUTING`, and `OUTPUT` chains, reducing the potential for interference with other netfilter clients, such as firewall managers.

Once the addresses are translated to node IDs, they must be captured by the MPD for flow-specific routing (a feature the Linux kernel does not provide). The MPD affects this capture by making a route for the MOPED internal subnet prefix that points to its universal tunnel interface. This route forces the kernel to queue packets with matching destinations to a device file from which the MPD can read them. The packets are then encapsulated, and injected back into the kernel where MobileIP's routing handles delivery to the component. The encapsulation used is MOPED Routing Encapsulation (MRCAP), a very lightweight encapsulation devised specifically for MOPED routing.

The reverse path, from the MOPED to correspondent hosts, is simple. The MPD decapsulates packets (after recording any flow binding contained in the encapsulation header) and relays the payload packet into the kernel network stack for address translation.

5.3. MOPED routing daemon

The MRD is a much more complex entity than the MPD. Its responsibilities include:

1. participation in assembly,
2. component-internal route management,
3. capture of and multipath selection for external traffic,
4. interfacing transports with the MOPED routing system.

The MRD's role in assembly is detailed in Section 4. Internal interface addresses are installed with a full netmask of `255.255.255.255`, so that the kernel does not forward *any* packets on this interface unless the MRD itself installs routes allowing it to do so.

Routes to other nodes in the same component (and their associated external interfaces) are disseminated via the link state routing protocol, and installed directly in the kernel routing table as host-specific routes. Component-local traffic is thus handled directly by standard IP hop-by-hop routing, with no per-packet processing.

The MRD manipulates kernel routing tables to direct externally destined packets into the tunnel device, so that they may be captured and have their path to the proxy selected appropriately. All routes installed by the MRD are kept in a separate kernel route table, so that routes installed by interface configuration tools and MobileIP in the main route table can still be used without interfering with the MRD. The MRD can then delegate specific routes to the `main` table by installing *throw* rules in the `moped` route table that instruct the kernel to lookup routes for matching destinations in the `main` route table. A representative set of route tables is shown in Figure 3. In this example, `mrd0` is the name of the MRD's universal tunnel interface. This node has ID `10.0.3.0` – the source address of the default route into the tunnel. The node has installed routes to its peers `10.0.4.0` and `10.0.5.0`, specifically from its interface `10.0.3.1` to their interfaces `10.0.4.1` and `10.0.5.1`. The MRD makes this

```
throw 255.255.255.255  table moped
10.0.4.0 via 10.0.4.1 dev eth0  table moped  proto 169
    src 10.0.3.0
10.0.4.1 dev eth0  table moped  proto 169  scope host
    src 10.0.3.1
192.17.240.55 via 10.0.5.1 dev eth0  table moped
    proto 169 src 10.0.3.0
10.0.5.0 via 10.0.5.1 dev eth0  table moped  proto 169
    src 10.0.3.0
10.0.5.1 dev eth0  table moped  proto 169  scope host
    src 10.0.3.1
throw 192.17.240.48/29  table moped
default dev mrd0  table moped  scope link  src 10.0.3.0

192.17.240.56 dev ppp0  proto kernel  scope link
    src 192.17.240.51
192.17.240.32/27 dev eth1  proto kernel  scope link
    src 192.17.240.50
127.0.0.0/8 dev lo  scope link
default via 192.17.240.33 dev eth1
```

Figure 3. Sample kernel route tables for a MOPED; output from `ip route list table all`

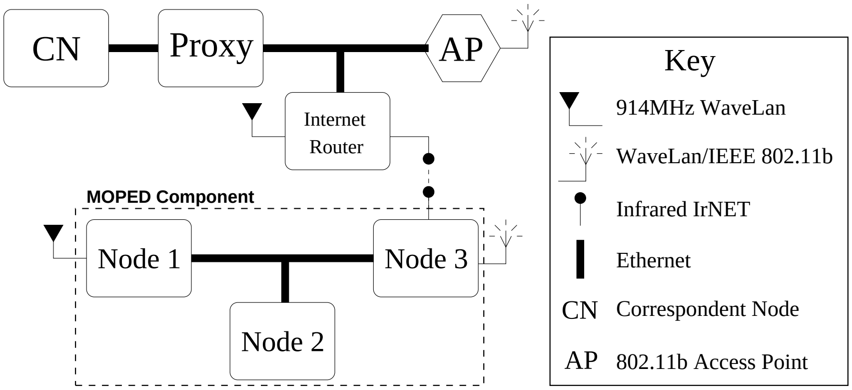

Figure 4. The MOPED testbed network

multiplicity of routes – a separate route to each interface – available for future in-component multipath route optimizations. The default route in the `moped` route table forces all externally destined packets into the tunnel, so that the MRD may perform flow-specific routing. After choosing an appropriate path to the proxy, the MRD encapsulates the packet and delivers it to the chosen perimeter node with standard IP routing. This node has a route to external interface `192.17.240.55`, and is attached to interfaces `192.17.240.50` and `192.17.240.51`. This route table corresponds to the node marked “Node 3” in Figure 4.

At perimeter nodes, the Linux `SO_BINDTODEVICE` socket option [15] is used to force packets through raw sockets bound to the proper external interface. Such external interfaces may, in fact, be virtual interfaces representing the end of a MobileIP established tunnel, which will then encapsulate the packets. and deliver them through the Internet to the proxy. A future integration of the MRD with a MobileIP agent will merge the two layers of encapsulation, reducing overhead for bidirectional tunneled traffic. Some possible routing optimizations to eliminate “triangle routing” are presented in our previous work [5].

5.4. MOPED routing agent coordination

There is a simple exchange of control messages between the MRDs and the MPD. The control messages:

- remotely establish flow-route table entries,
- publish ports to the outside world (i.e., export services from the MOPED), and
- transfer component topology from the component leaders to the MPD.

Control messages are piggy-backed on data packets (as an extension to the encapsulation header) when possible. Since control messages are usually small – on the order of the size of the IP packet header – and sent infrequently, piggybacking can save overhead.

6. Experimental results

We describe the experiments that illustrate MOPED’s ability to utilize multiple channels to enhance the user’s perceived level of network service. The first set of experiments in Section 6.2 shows the aggregation of bandwidth from multiple channels to improve throughput, and the second set in Section 6.3 shows how diverse channels can be used to smooth handoffs due to mobility.

6.1. Experimental setup

All tests were performed on the single-component MOPED test network depicted in Figure 4. Since our assumption is that the perimeter links will be the bottleneck for throughput to the component, we used wired Ethernet for the component-internal link technology. The bandwidth of the internal link is greater than the sum of all external link bandwidths, so that the internal subnet will never constrict throughput for external traffic.

Three different, non-conflicting wireless technologies are used to connect the component to the Internet: 914MHz Lucent WaveLAN, IEEE 802.11b WaveLAN/IEEE, and 4Mb Fast InfraRed (FIR). There are no distinguishing characteristics that influenced the choice of these wireless technologies – they were chosen for quality of driver support and for the fact that none of the three will conflict and reduce the bandwidth perceived by another link. No other traffic was on the network during testing, to avoid congestive interference.

The three nodes are a variety of x86 laptops all running a Linux 2.4 kernel and the MOPED routing daemon. The Proxy is a desktop x86 machine also running a 2.4 Linux

	802.11b (1Mb/s)	IR	914MHz	Aggregate (1Mb/s)	802.11b (11Mb/s)	Aggregate (11Mb/s)
Goodput	92 KB/s	122 KB/s	126 KB/s	295 KB/s	525 KB/s	711 KB/s
ETA	356 s	269 s	260 s	111 s	62 s	46 s

Table 1. Results for the 32MB transfer

kernel. We do not describe details of machine configuration since we have not yet focused on performance of the MOPED software. The tests were designed so that all processors were capable of forwarding packets at wireline speed, so there were no congestive losses due to inability of the processors to maintain full speed routing. The network node labeled “Internet Router” in the diagram is simply an access point for the IR and 914MHz links; it contains no special support and is not affiliated with the MOPED. All tests were run after the assembly process had reached a stable state, a process which happens very quickly in such a small network.

All transport tests transferred data from the Correspondent Node (CN) into the MOPED. We used a transport protocol influenced by prior work on inverse multiplexing [6, 7, 8], the MOPED inverse multiplexing transport protocol (MIXTP). MIXTP is a self-clocking, window-based protocol that mimics TCP NewReno [3] on a per-channel basis, with extensions to enable inverse multiplexing of data packets across several channels. For single channel tests, MIXTP (with these extensions disabled) was used as a very close approximation to the behavior of TCP NewReno. MIXTP also forgoes the wireless-specific loss discrimination techniques used by prior protocols, pessimizing the throughput achievable in a wireless environment. The goal of these experiments is to purely isolate the effects of resource aggregation on transport service.

6.2. Aggregation

In this experiment, the MOPED user wishes to download a significant amount of bulk data (32MB) from the CN to Node 2 in the MOPED. The perceptive reader will note from the diagram that Node 2 has no external connectivity, and is thus dependent on the MOPED system to provide connectivity. We contrast the network service perceived when using a single-channel transport over the best available channel, and that of a multiplexing transport over all available channels. All results reported are averaged over ten runs to eliminate variations.

In a real-world environment, choosing the best available channel is a very tricky proposition. While the peak raw bandwidth of each attached interface may be known, that figure does not necessarily indicate the amount of available bandwidth from that interface, which may be affected by factors such as noise, interference from other senders, or congestion on some link between a sender and receiver. We simulate this uncertainty in this experiment by restricting the bandwidth of the 802.11b interface to 1 Megabit/second, using the Linux `iwconfig` [14] tool to constrain the interface driver to use only the basic 1 Mb/s 802.11 modulation scheme. This approximates the effect of locating the user at the edge of an 802.11 cell, where the radio is only able to sustain low transmission rates.

Table 1 shows the results achieved by this transfer for various choices of transport channel. If the choice of the “best” channel is made statically, the 802.11b link is the clear winner at 525 KB/s. At this rate, the entire transfer would complete in about 62 seconds. Under adverse channel conditions due to e.g. interference or multipath interference, the 1 Mb/s restricted 802.11b link has the *worst* data rate, stretching the transfer time to 356 seconds. Ironically, the best channel in this case is the channel with the *lowest* raw data rate, the 914MHz WaveLAN. If the MRD on Node 2 is somehow wise (or just lucky) enough to choose to route the transfer over the 914MHz link, the transfer will complete in 260 seconds.

In this experiment, the ability of MOPED/MIXTP to utilize multiple channels enhances throughput while avoiding the decision of which channel to use. Regardless of which channel is currently optimal, aggregation can achieve good throughput without even probing for channel characteristics. In comparison to the 1Mb/s channel, aggregation triples throughput, diminishing the transfer time by 245 seconds—a 69% reduction. When the 802.11b channel is in a good state, the aggregation is less impressive: from 62 to 46 seconds, still an improvement, but only by 27%.

The MOPED enables the combination of all channels to the component into a single effective channel for any node to use. This sharing of available communication resources satisfies one of the major goals of cooperative mobility.

6.3. Handoff

The benefit of resource aggregation in terms of raw bandwidth is clear enough, but aggregation can also be of benefit to mobility-induced handoffs in areas of overlapping coverage. As is the case for bandwidth in the bulk transfer experiments above, aggregation has both a quantitative and a qualitative effect on handoffs.

In these experiments, we simulate the effect of passing from the coverage area of one wireless technology, through an area of overlap, and into the coverage of another wireless technology. For these experiments, the IR link is disabled. Initially, the 802.11b link is running at full bandwidth, and the 914MHz is disabled – this first phase lasts for 15 seconds. We simulate moving from an 802.11b cell into a 914MHz cell and back again. We introduce a smooth degradation in bandwidth on the 802.11b interface by changing the data modulation with `iwconfig` every 5 seconds in steps from 11Mb/s to 5.5Mb/s, 2Mb/s, 1Mb/s and disabling the interface completely. After a 15 second period, reentry into the 802.11b cell is modeled by reversing this sequence of steps. The overall effect approximates a transition out of, and back into, the coverage of the 802.11 link. The 914MHz link comes up as the 802.11b transitions down to 2Mb/s, and is disabled as the 802.11b comes up to 11Mb/s. The sequence of events is depicted in the plot in Figure 5.

Handoff performance is dependent on when and how a transport layer connection shifts its usage of the two link layers. In a smooth handoff, the best performance that any handoff approach can hope to achieve is to switch at the ideal time, when the available bandwidth from the new link surpasses that of the old link. Figure 6 illustrates a hypothetical handoff from link A to link B. When aggregation is used, simultaneous utilization of both links during the overlap period can result in a brief *increase* in throughput, instead if a decrease. If the two links are of approximately equal capability, aggregate handoff can maintain the integrity of the flow without a reduction in rate.

To examine this hypothesis, we ran the simulated handoff scenario using a) MIXTP aggregating both channels, and b) single-channel MIXTP. To approximate the behavior of a naïve vertical handoff, the single-channel approach performs handoff after the link in use fails. This is often referred to as late handoff, since the handoff is performed after the ideal transition point. We illustrate the performance of the two approaches by presenting a plot of the sequence numbers of packets originated at the sender in Figure 7.

The rate changes imposed on the 802.11b link by `iwconfig` are not instantaneous, and do interfere with packet transmission. The results, however, effectively illustrate the advantage of aggregation for handoffs. The interesting regions of the plots are from 15-30 seconds and 45-60 seconds, all rate transitions take place in these regions. The higher slope of the aggregate plot during these transitional regions shows its advantage in bandwidth. The aggregate transfer finishes almost 20 seconds earlier. The single channel transfer flat lines briefly at 30 and 60 seconds, the points at which the link in use fails and handoff occurs. In the period from 55-60 seconds, we see that the aggregate transfer briefly achieves aggregated bandwidth greater than what is available from the 802.11b link alone (i.e., the slope of the curve is temporarily greater in the 55-60 range than after 60 seconds, when the 802.11b is back at full speed).

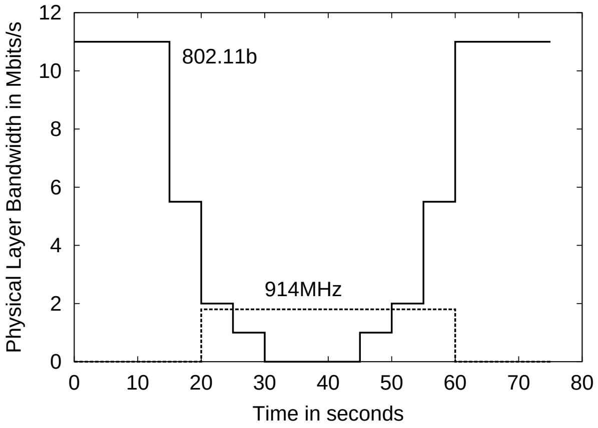

Figure 5. Raw link bandwidths during handoff experiment

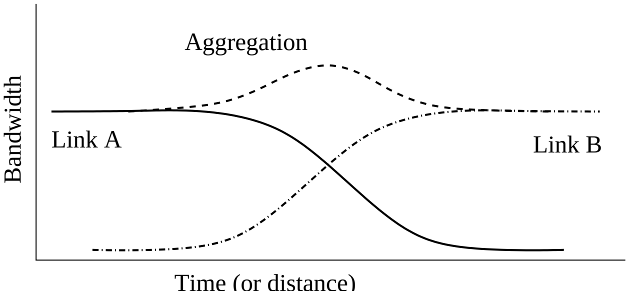

Figure 6. Advantage of aggregation for handoff

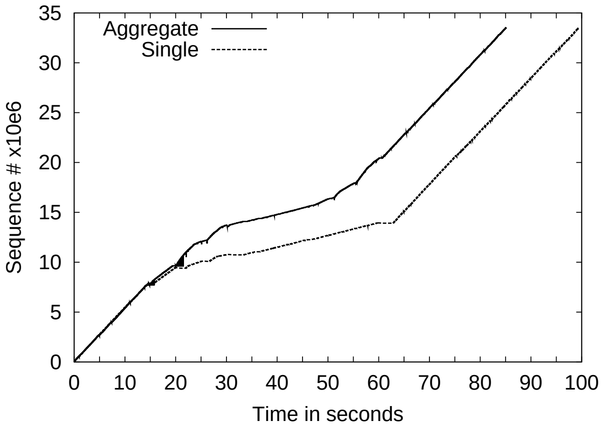

Figure 7. Sequence numbers sent during handoff

7. Conclusions and future work

We have demonstrated the effectiveness of a MObile grouPEd Device (MOPED) in providing enhanced network services to a mobile user. By aggregating available communication resources, the MOPED architecture enables a group of mobile devices to cooperate to provide better coverage and throughput, with less variation in channel quality.

Although we have proved the validity of the basic MOPED concept as a paradigm for group mobility, much work still remains to be done to achieve the full goals of the project. Our near-term interests are to study routing interactions within a MOPED. We are researching a framework for resource-aware routing with the goal of optimizing routing within the components, and managing traffic flows from the proxy into each component. We also plan to investigate mechanisms that will probe paths between components for performing optimal inter-component routing. The MOPED has an active routing structure: since one of the key resources to conserve is power, the MOPED routing fabric can actively bring links up and down.

Other topics for future MOPED investigation are:

- Support for non-MOPED-enabled hosts; sharing the MOPED's rich resources with legacy devices.
- Integrating transparent support for aggregating, wireless-aware transport protocols; allow TCP to be replaced with a MOPED-specific transport protocol, using a split connection model at the proxy.
- Develop a user-interaction model for the MOPED. How should a user configure and manage the MOPED? A sub-problem to this issue is the application interface to MOPED functionality.

The area of resource management in a cooperative mobile environment is a rich one for the MOPED to explore.

Acknowledgements

We would like to acknowledge the large contribution of Luiz Magalhães's inverse multiplexing protocols, which formed the basis for our multi-channel aggregation. We would also like to thank the Mobuis group for their continual input and support of the MOPED project.

References

[1] M. Baker, Z. Zhao, S. Cheshire, and J. Stone. Supporting mobility in mosquitonet. In *USENIX Winter Conference*, 1996.

[2] L. Bellier and C. Casteluccia. Mobile networks support in mobile IPv6. Internet Draft (Work in Progress) draft-ernst-mobileip-v6-network-01.txt, Internet Engineering Task Force, November 2000.

[3] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno, and SACK TCP. *Computer Communications Review*, 1996.

[4] T. J. Kniveton, J. T. Malinen, V. Devarapalli, and C. E. Perkins. Mobile router support with mobile IP. Internet Draft (Work in Progress) draft-kniveton-mobrtr-00.txt, Internet Engineering Task Force, November 2001.

[5] R. Kravets and C. Carter. A cooperative approach to user mobility. *ACM Computer Communications Review*, 31(5):57–69, October 2001.

[6] L. Magalhães and R. Kravets. End-to-end inverse multiplexing for mobile hosts. In *19th Brazilian Symposium on Computer Networks (SBRC'01)*, 2001.

[7] L. Magalhães and R. Kravets. MMTP: Multimedia multiplexing transport protocol. In *The First Workshop on Data Communications in Latin America and the Caribbean (SIGCOMM-LA 2001)*, 2001.

[8] L. Magalhães and R. Kravets. Transport level mechanisms for bandwidth aggregation on mobile hosts. In *9th International Conference on Network Protocols ICNP 2001*, 2001.

[9] P. Maniatis, M. Roussopoulos, E. Swierk, K. Lai, G. Appenzeller, X. Zhao, and M. Baker. The mobile people architecture. *ACM Mobile Computing and Communications Review*, 3(3), July 1999.

[10] A. Myles, D. Johnson, and C. Perkins. A mobile host protocol supporting route optimization and authentication. *IEEE Journal on Selected Areas in Communications*, 13(5):839–849, 1995.

[11] C. Perkins. IP mobility support. Request for Comments (Proposed Standard) RFC 2002, Internet Engineering Task Force, October 1996.

[12] J. Postel. Transmission control protocol. Request for Comments (Standard) RFC 793, Internet Engineering Task Force, September 1981.

[13] M. Stemm and R. H. Katz. Vertical handoffs in wireless overlay networks. *ACM Mobile Networking (MONET), Special Issue on Mobile Networking in the Internet*, 1998.

[14] J. Tourrilhes. Wireless tools for linux, March 2002. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/-Tools.html.

[15] X. Zhao, C. Castelluccia, and M. Baker. Flexible network support for mobility. In *Fourth ACM International Conference on Mobile Computing and Networking (MOBICOM'98)*, 1998.