

Moving between infrastructure and ad hoc wireless networks: 'opportunistic' mobile middleware

Ling-Jyh Chen, Shanky Das, Mario Gerla, Alok Nandan

Computer Science Department, UCLA

1. Introduction

During the past two decades we have witnessed a major shift from fixed to mobile phones all over the world. The number of mobile phones has surpassed that of fixed phones in most countries. This is no surprise since telephony is an inherently “mobile” application. The same shift has been happening in personal computing platforms, from desktops to laptops and PDAs. The proliferation of mobile computing platforms has coincided with the emergence of mobile data communications, namely, wireless LANs 2.5G and 3G cellular data services. This phenomenon has caused a major paradigm shift in the way we access the Internet, from fixed to wireless and mobile. Already wireless Internet access has exceeded wired access, and in fact one expects that within a few years the great majority of client will be not only wireless and portable, but also equipped with multiple wireless interfaces. Today, the majority of Internet applications are still “stationary” in nature; ie, we do e-mail, web browsing, file downloads, and play internet games from our homes or offices. We exploit the wireless interface mainly to avoid cables. However, there is an emergence of truly mobile access scenarios (from cars or public transport vehicles or while walking in shopping malls). In parallel, there is an emergence of new, mobile applications and services, such as location based services, car navigation services, dynamic workgroups, pervasive computing and interaction with the environment.

Clearly, several concurrent factors are favoring the “mobilization” of computing and communications, namely: the miniaturization of devices (which have become portable); the availability of long life, light load batteries; the availability of efficient wireless access media (from cellular to wireless LAN, personal wireless media and ad hoc networks), and; the emergence of mobile, nomadic applications. As these mobile scenarios are emerging, there is another important “layer” of services that must be implemented to make this happen: namely, mobile middleware. The mobile middleware is essential both for “mobilizing” legacy infrastructure applications (eg, e-mail, web browsing, etc) and for enabling applications that are purely mobile (eg, car navigation safety).

In this chapter we will focus on different wireless media access schemes and on the mobile middleware needs of each. Three major wireless access and network techniques exist today: cellular, wireless (infrastructure) LAN and ad hoc wireless networks. Since cellular and wireless LAN technologies are well understood and have been extensively covered in the open literature, we will focus mainly on the emerging ad hoc wireless and

personal networks. In the process, we will identify key mobile applications and will discuss emerging mobile middleware functions required to support them.

The book chapter is organized as follows. In section 2 we study mobility management in last hop wireless networks and discuss various handoff solutions that manage client movements. In section 3 we review the MANET architecture and study its evolution from battlefield to commercial applications. In section 4 we study scalable routing in presence of mobility. In section 5 we examine the need of P2P overlays in MANETs, with specific application to commercial scenarios. In section 6 we study an emerging commercial application, file sharing in the vehicular network; we propose P2P swarming middleware for this application. Sect 7 concludes the chapter.

2. Mobility management in the “last wireless hop”: horizontal handoff and Mobile IP

We start with the overview of mobile services required in “last hop” wireless networks. The scenario of interest is extremely broad. It includes cellular networks (from GSM, CDMA to 2.5G and 3G); indoor wireless LANs (eg, 802.11abg for basic Internet access; 802.11n for high speed; 802.11e for QoS oriented multimedia access); outdoor, urban wireless access networks (eg, 802.11s for urban Mesh Network access; 802.16 for urban high speed distribution); short range, low data rate access networks (eg Bluetooth and ZigBee for personal and pervasive access). All of the above scenarios are examples of “infrastructure” type networks. The physical environment is partitioned into cells. Each cell is controlled by an access point which acts as gateway to the wired Internet (in some of the above schemes – eg, 802.11s and 802.16 - there is in fact a wireless fixed backbone between the access point and the Internet gateway; but, the following considerations apply to those cases as well). A user may move from one cell to another, either within the same technology (eg, UMTS), or; across technologies. In fact, a mobile client is often equipped with multiple radio interfaces and can roam across technologies. When this happens, the user must “re-register” with the new access point/base station. This registration may happen at one or more layers of the protocol stack. Often, it happens at the middleware layer, thus requiring “mobile middleware” services. This registration procedure goes under the name of “handoff”. The next section describes various handoff modes.

2.1. Handoffs

Handoff occurs when the user switches between different network access points. Handoff techniques have been well studied and deployed in the domain of cellular system and are gaining a great deal of momentum in the wireless computer networks, as IP-based wireless networking increases in popularity.

Differing in the number of network interfaces involved during the process, handoff can be characterized into either *vertical* or *horizontal* [1], as depicted in Figure 1. A vertical handoff involves two different network interfaces, which usually represent different technologies. For example, when a mobile device moves out of an 802.11b network and into a 1xRTT network, a *vertical* handoff occurs. A *horizontal* handoff

occurs between two network access points that use the same technology and device interface. For example, when a mobile device moves between 802.11b network domains, the handoff event would be considered as horizontal since the connection is disrupted solely by the change of 802.11b domain (ie, different frequency channel) but not by the change of wireless technology.

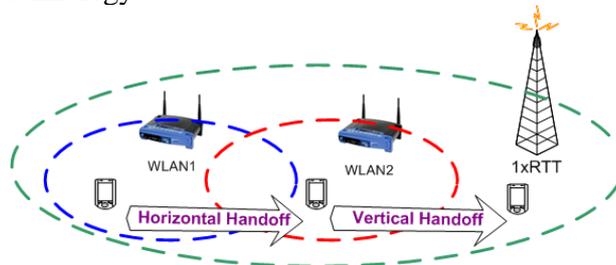


Figure 1: Horizontal and Vertical Handoff

A handoff is defined to be *seamless* if it maintains the connectivity of all applications running on the mobile device. Seamless handoffs aim to provide continuous end-to-end data service in the face of any link outages that occur during switchover. Low latency and minimal packet loss are the two critical design goals. Low latency requires that the switch from one path to the other be completed almost instantaneously; service interruptions must be minimized.

Various seamless handoff techniques [2][3][4][5] have been proposed. These proposals can be classified into two categories: network layer approaches and upper layer approaches. Network layer approaches are based on IP address “indirection” through a home agent and a foreign agent. They can be accomplished using IPv6 [6] or Mobile IPv4 [7] standards. These network layer approaches, however, are costly to implement. They require the deployment of several agents on the Internet for relaying and/or redirecting the data to the moving host (MH). Because of these reasons, the upper layer approaches are becoming increasingly popular. These approaches implement a session layer (in fact, a “mobile middleware” layer) above the transport layer that “hides” any connection changes at the underlying layers and makes them transparent to the application [8][9][10][11][12]. There are also upper layer approaches that implement mobility support at the transport layer, in fact requiring the development of new transport layer protocols such as SCTP [13] and TCP-MH [14].

2.2. Smart vertical handoff

As mentioned earlier, several devices today have multiple radio interfaces. The opportunity then arises to select the “best” radio option, leading to a “user initiated” (as opposed to infrastructure driven) vertical handoff. Basically, smart handoff has all the ingredients of soft handoff; in addition, it includes mobile middleware software to select the best alternative. In this section, we present an example of smart handoff, the *Smart Decision Model* [15] to support flexible configuration in executing vertical handoffs. Figure 2 depicts the Smart Decision Model. In this figure, a Handoff Control Center (HCC) provides the connection between the network interfaces and the upper layer applications. HCC is composed of four components: Device Monitor (DM), System Monitor (SM), Smart Decision (SD), and Handoff Executor (HE). DM is responsible for

monitoring and reporting the status of each network interface (i.e. the signal strength, link capacity and power consumption of each interface). SM monitors and reports system information (e.g. current remaining battery). SD integrates user preferences (obtained from user set default values) and all other available information provided by DM, SM to achieve a “Smart Decision”, to identify the “best” network interface to use at that moment. HE then performs the device handoff if the current network interface is different from the “best” network interface.

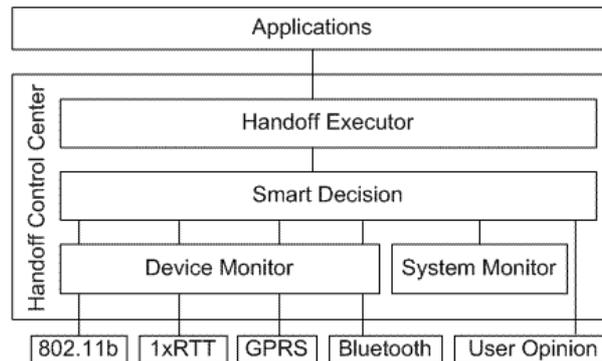


Figure 2: Smart Decision Model

A Handoff Control Center (HCC) in accordance to above is implemented in the vertical handoff testbed to perform automatic handoffs to the “best” network interface. In our design, there are two phases in SD: the *priority* phase and the *normal* phase. The SD algorithm is described in Figure 3.

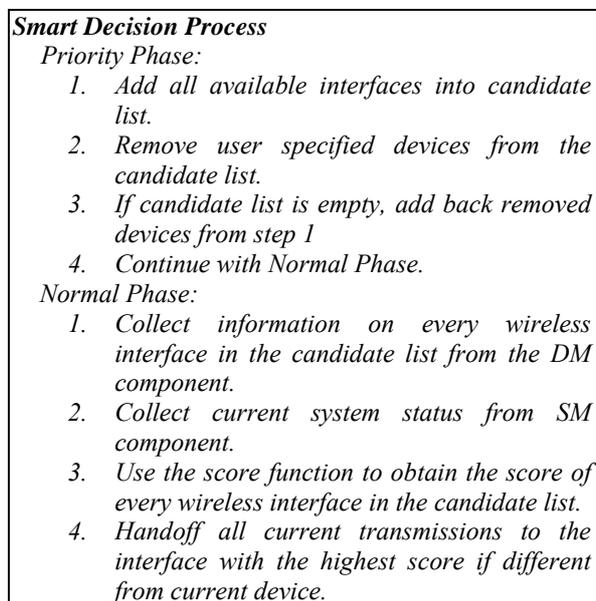


Figure 3: Algorithm for making Smart Decisions on HCC

Priority and normal phases are necessary in SD to accommodate user-specific preferences regarding the usage of network interfaces. For instance a user may decide not

to use a device when the device may cause undesirable interferences to other devices (e.g. 802.11b and 2.4GHz cordless phones). With priority and normal phases in place, the SD module provides flexibility in controlling the desired network interface to the user. Additionally, SD deploys a *score function* to calculate a score for every wireless interface; the handoff target device is the network interface with the highest score. More specifically, suppose there are k factors to consider in calculating the score, the final score of interface i will be a sum of k weighted functions. The *score function* used is the following:

$$S_i = \sum_{j=1}^k w_j f_{j,i} \quad 0 < S_i < 1, \quad \sum_{j=1}^k w_j = 1 \quad (1)$$

In the equation, w_j stands for the weight of factor k , and $f_{j,i}$ represents the normalized score of interface i of factor j . The “best” target connection interface at any given moment is then derived as the one which achieves the highest score among all candidate interfaces. We further break down the score function to three components where each accounts for usage expense (E), link capacity (C), and power consumption (P), respectively. Therefore Eq. 1 becomes:

$$S_i = w_e f_{e,i} + w_c f_{c,i} + w_p f_{p,i} \quad (2)$$

Additionally, there is a corresponding function for each term $f_{e,i}$, $f_{c,i}$, and $f_{p,i}$, and the ranges of the functions are bounded from 0 to 1. The functions are illustrated below:

$$f_{e,i} = \frac{1}{e^{\alpha_i}} \quad f_{c,i} = \frac{e^{\beta_i}}{e^M} \quad f_{p,i} = \frac{1}{e^{\gamma_i}} \quad \text{where } \alpha_i \geq 0, M \geq \beta_i \geq 0, \text{ and } \gamma_i \geq 0 \quad (3)$$

The coefficients α_i , β_i , γ_i can be obtained via a lookup table or a well-tuned function. In Eq. 3, we used the inversed exponential equation for $f_{e,i}$ and $f_{p,i}$ to bound the result to between zero and one (i.e. these functions are normalized), and properly model users preferences. For $f_{c,i}$, a new term M is introduced as the denominator to normalize the function, where M is the maximum bandwidth requirement demanded by the user. Without specified by the user, the default value of M is defined as the maximum link capacity among all available interfaces. Note that, the properties of bandwidth and usage cost/power consumption are opposite (i.e. the more bandwidth the better, whereas lower cost/power consumption is preferred).

2.3. Managing server rate and content during handoff: CapProbe

So far we have considered the “client side” of the handoff. Namely, the client selects the best option etc. Suppose now that the client is a thin client, say a smart phone. It is receiving a soccer game video stream from the server. The client moves from indoor 802.11 at 5Mbps to outdoor 1xRTT at 100Kbps. Smooth handoff guarantees that the connection is maintained. But, it is obvious that havoc will occur unless the server (or the transcoding capable proxy that caches the server stream) detects the change in client

Internet access capacity and adjusts its rate and content accordingly (say from full motion video to highly compressed MPEG4 video or even still frames).

A new mobile middleware software is required, server adaptation middleware, to make the server (or proxy) immediately aware of client changes and to select the best server delivery strategy. We have recently implemented such middleware service using basic a basic capacity estimation technique called CapProbe [17]. CapProbe is a “packet pair” method that measures the capacity of the narrow link on the path (in our case invariably the last wireless hop) with extreme speed (order of seconds). The concept is illustrated in Figure 4. A packet pair is launched by the source. The packets get separated along the path due to varying link capacities. The ratio of packet size over inter packet interval at destination yields the narrow link capacity. Details on the actual CapProbe tool are found in [17]. Referring to Figure 5 we see an internet path ending with an 802.11 link. This was the actual setup of an experiment carried out at UCLA in the Network Research Lab [18]. The 802.11 client is exposed to interference from a Bluetooth user operating in the same frequency. Interference notwithstanding, CapProbe manages to evaluate the exact capacity of the 802.11 channel.

The server “mobile middleware” embeds periodic packet pairs in the multimedia stream (by transmitting some of the video packets back to back) and is constantly informed (by client feedback) of the last hop capacity. It can then adjust rate/content dynamically to client capacity. Using the same principle of client feedback, the server middleware can also adjust to device “form and type” change (say, the user switches from laptop to smart phone as he steps out of the car, yet maintaining UMTS connectivity, say).

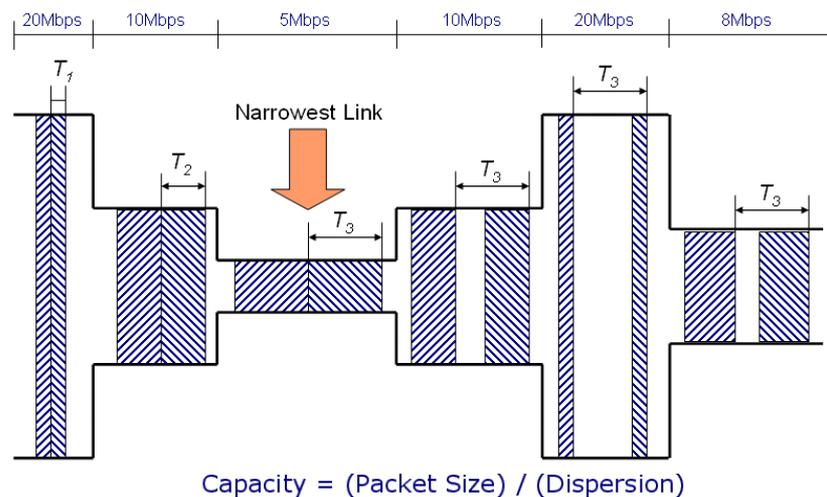


Figure 4: CapProbe: a simple and fast capacity estimation tool

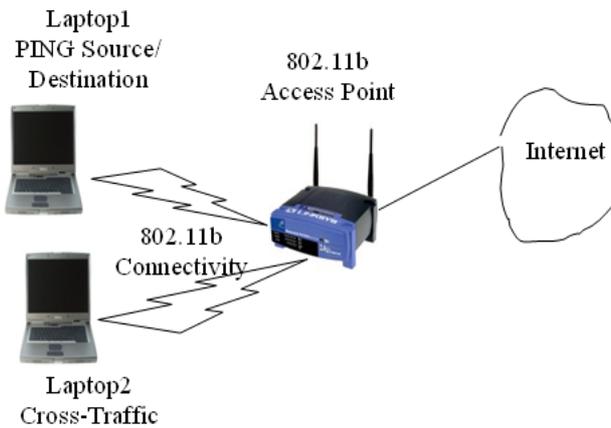
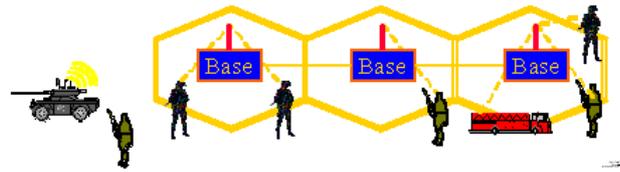


Figure 5: CapProbe testbed with last hop wireless link

3. The Mobile ad hoc Network (MANET)

A Mobile “Ad hoc” wireless NETWORK (MANET) is a network established for a special, often extemporaneous service customized to applications. The ad hoc network is typically set up for a limited period of time, in an environment that may change from application to application. As a difference from the Internet where the TCP/IP protocol suite supports a vast range of applications, in the MANET the protocols are tuned to a specific customer and application (eg, send a video stream across the battlefield; find out if there is a fire in the forest; establish a videoconference among several teams engaged in a rescue effort, etc). The customers move and the environment may change dynamically and unpredictably. For the MANET to retain its efficiency, the ad hoc protocols at various layers may need to self-tune to adjust to environment, traffic and mission changes. From these properties emerges the vision of the MANET as an extremely flexible, malleable and yet robust and formidable network architecture. Indeed, an architecture that can be deployed to monitor the habits of birds in their natural habitat; or, can be organized to interconnect rescue crews after a Tsunami disaster; or, yet can be structured to launch deadly attacks onto unsuspecting enemies.

Standard Base-Station Cellular Networks



Ad Hoc, Multihop wireless Networks

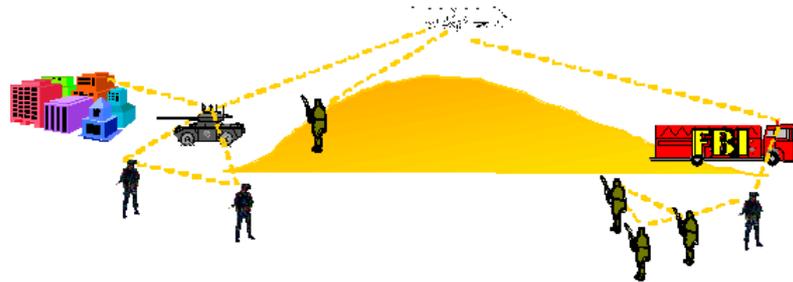


Figure 6: Infrastructure vs Ad Hoc wireless network

MANETs are set apart from conventional wired or wireless infrastructure type networks by a number of unique attributes and requirements. Perhaps the two most critical attributes are self-configurability and mobility. A third important requirement (which is critically impacted by the first two) is scalability. We review these attributes next:

Self-organization: the MANET is deployed and managed independently of any preexisting infrastructure. This is the most important prerequisite to qualify a wireless network as ad hoc. Consequently, the network must autonomously determine its own configuration parameters including: addressing, routing, clustering, position identification, power control, etc. In some large networks, special nodes (eg, mobile backbone nodes) coordinate their position and motion to provide coverage of disconnected islands. This way, an “infrastructure” may be created within the ad hoc network itself.

Mobility: the fact that nodes move is probably the most important attribute of MANETs. Mobility differentiates MANETs from their close cousins, the sensor networks. Mobility dictates network and application level protocols. For example, rapid deployment in unexplored areas with no infrastructure may require that some of the nodes form scouting teams/swarms. These in turn coordinate among themselves to create a task force or a mission. Mobility may be in some cases a challenge for the designer, and may become part of the solution in other cases. We can have several types of mobility models: individual random mobility, group mobility, motion along preplanned routes, etc. The mobility model can have major impact on the selection of a routing scheme and can thus influence performance.

Scalability: in both military and civilian applications (eg, large battlefield deployments, urban vehicle grids, etc) the ad hoc network can grow to several thousands of nodes. For wireless infrastructure-type networks (eg, urban mesh networks) scalability is simply handled by a hierarchical construction. Mobility appears to be the discriminator between easy and difficult scaling. A hierarchical model is very scalable in static networks (as demonstrated by the Internet). Limited mobility in an infrastructure can be easily handled using Mobile IP or other handoff and re-direction techniques. Pure ad hoc

networks, due to their self configuring nature and consequent unrestricted mobility, do not tolerate a classic hierarchy structure and a mobile IP approach. Thus, mobility, jointly with large scale is one of the most critical challenges in ad hoc designs.

3.1. The evolution of MANETs – from battlefield to campus networks and urban grids

MANETs were born in the early '70s on the heel of the ARPANET success, when DARPA recognized the strategic importance of the packet switching technology in the automated battlefield. Since then, the military has been the major sponsor of MANET research and development in industry and academia. A few years ago, NSF has also joined in the support of MANET research, exploring the transfer of this technology to civilian and possibly commercial applications. The support from Industry to MANETs, however, has been minimal (as compared to other areas on networking), in part due to the fact that commercial applications have been very slow in materializing. Because of the source of the funding, it is no surprise that most of the MANET problems addressed today by researchers are directed to large scale, specialized scenarios, say battlefield, civilian defense and disaster recovery. These are typically self-configured networks, totally decoupled from any commercial network infrastructure. One may say that even the network scenarios addressed by the MANET IETF working group are better fit to military and civilian disaster recovery applications than to commercial ones.

Very recently, there have been new technology developments which might bring new alternatives in the MANET area and may help the transition to commercial MANET applications. The first emerging technology is the "Personal Area Network", spearheaded by Bluetooth (802.15.1) and by the recently introduced ZigBee and 802.15.4 standards. It will make sense to interconnect a few Bluetooth piconets in a small scale MANET, called scatternet, to facilitate work group communications (to exchange business cards, files, images) and to have a more efficient connection to the Internet (eg 802.11, UMTS, etc). The second technology is the wireless LAN (802.11). The 802.11 technology and its derivatives dominate in the home; in university and industrial campuses; in public areas (mall, airport lounge, coffee shop, etc) and; in urban mesh networks (as shown in Figure 7). The single hop wireless LAN however, has range limitations. Two or three hop MANETs can be used to "opportunistically" extend the range of the wireless LAN. The third technology is DSRC (digital short range communications). This technology addresses car- to-car and car-to-Internet communications for navigation safety purposes. The DSRC technology will pave the way for the "urban communications grid" concept, where car-to-car communications between any two vehicles will be made possible in the MANET, without using the fixed Internet. While navigation safety is the top DSRC priority, the urban grid will eventually enable the support of a broad range of new mobile applications.

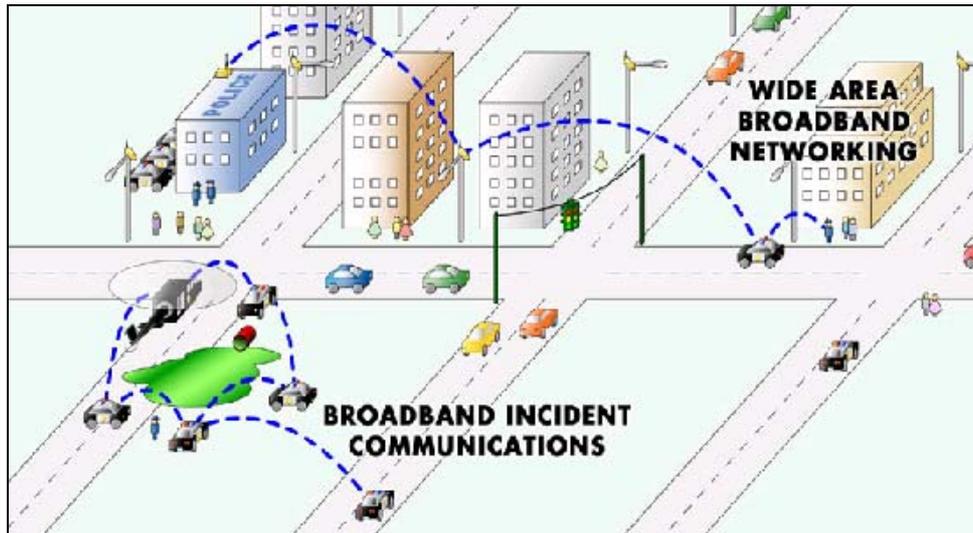


Figure 7: The urban mesh network

4. Handling Large Scale and Mobility in the Battlefield

Future battlefield operations will be characterized by massive deployment of autonomous agents such as Unmanned Ground Vehicles (UGVs) and Unmanned Airborne Vehicles (UAVs). The autonomous agents will be projected to the forefront for intelligence, surveillance, strike, enemy anti-aircraft suppression, damage assessment, search and rescue and other tactical operations. These agents will interact and support ground and airborne manned assets (eg, tank battalions, jet fighter and helicopter squadrons). They will also communicate with ground sensors. One can easily imagine how this scenario can comprise thousands of mobile nodes, some manned, some unmanned, and several more thousands of smaller, fixed nodes. Similar large scale, mobile networks are formed to recover from extensive civilian disasters such as earthquakes, tsunamis, chemical spills and urban terrorist attacks

One of the critical problems in ad hoc networks is routing. If the ad hoc network is stationary, then hierarchical routing proves to be a very scalable solution. When the network is mobile, however, the hierarchical routing solution introduces excessive overhead since the hierarchical addresses must be continuously updated to reflect the dynamically changing topology.

Mobility causes problems also with other protocol layers besides routing (eg, MAC layer, TCP). In particular, one of the major challenges in ad hoc TCP design is dealing with path disruptions caused by mobility. In large scale routing, however, mobility can also be a “friend”, in that it can be exploited to improve performance. In this section we show that “**group**” mobility can be harnessed via “landmarking” to lead to more scalable routing. Moreover, if **mobile backbone nodes** are deployed in the ad hoc network, connectivity can be enhanced.

4.1 Landmark Routing for Group Mobility

Typically, when wireless network size and mobility increase (beyond certain thresholds), current “flat” proactive routing schemes (i.e., distance vector and link state) become all together unfeasible because of line and processing O/H. In [21], we introduce a novel table driven routing protocol for wireless ad hoc networks - Landmark Ad Hoc Routing (LANMAR), LANMAR combines the features of Fisheye State Routing (FSR) [19] and Landmark routing [22].

The key novelty in LANMAR is the notion of keeping track of logical subnets in which the members have a commonality of interests and are likely to move as a “group” (e.g., brigade in the battlefield, colleagues in the same organization, or a group of students from same class). Moreover, a “landmark” node is elected in each subnet. LANMAR improves scalability by reducing routing table size and update traffic O/H. More precisely, it resolves the routing table scalability problem by using an approach similar to the landmark hierarchical routing proposed in [22] for wired networks. In the original landmark scheme, the hierarchical address of each node reflects its position within the hierarchy and helps finding a route to it. Each node has full knowledge of all the nodes within the immediate vicinity. At the same time each node keeps track of the next hop on the shortest path to various landmarks at different hierarchical levels. Routing is consistent with the landmark hierarchy and the path is gradually refined from top level hierarchy to low levels as a packet approaches destination.

We apply the wired network landmark concept to FSR (Fisheye State Routing) [20] to reduce routing update overhead for nodes that are far away. Each logical subnet has one node serving as “landmark”. Beyond the fisheye scope the update frequency of the landmark nodes remains unaltered, while the update frequency of regular nodes is reduced to zero. As a result, each node will maintain accurate routing information about immediate neighborhood and as well as to landmark nodes. When a node needs to relay a packet, if the destination is within its neighbor scope as indicated in the routing table, the packet will be forwarded directly. Otherwise, the packet will be routed towards the landmark corresponding to the destination logical subnet. The packet does not need to go all way to the landmark. Rather, once the packet gets within the scope of the destination, it is routed to it directly.

At the beginning of the execution, no landmark exists. Protocol LANMAR only uses the FSR functionality. As the FSR computation progresses, one of the nodes will learn (from the FSR table) that more than a certain number of group members (say, N) are in the FSR scope. It then proclaims itself as a landmark for this group. The landmark information will be broadcast to the neighbors jointly with the topology update packets. In case of tie, lowest ID breaks the tie. The competing nodes defer. When a landmark dies, its neighbors will detect the silence after a given timeout. A new round of landmark election then starts over the group in question.

In conclusion, LANMAR is an excellent example of routing protocol that exploits group mobility by "summarizing" routes and reducing table storage and line overhead. Simulation results have shown that LANMAR empowered network can easily scale to thousand of nodes.

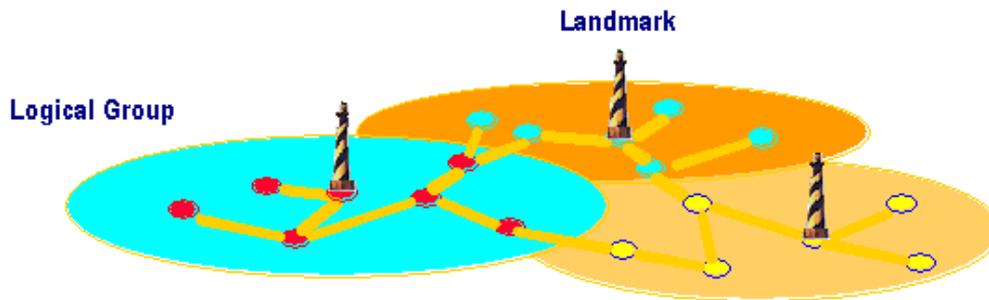


Figure 8: Landmark routing

5. MANETs and P2P mobile middleware

In the Wired Internet, a P2P network is basically an overlay network justified by the need for specialized functions that are not possible (or not cost-effective) in the IP layer. These functions must be performed at the middleware or application layer. Classic examples of Internet overlay networks are: real **time multicast overlays**, which overcome the lack for multicast support in the IP routers. And; P2P **distributed index systems** such as Gnutella, BitTorrent, and Pastry. These P2P indices are typically implemented as overlays that permit efficient “content” routing based on Distributed Hash Tables (DHTs), for example. Content based routing is not possible in the IP layer.

In MANETs there is an even stronger need for P2P overlays for the following reasons: (a) mobile ad hoc applications need sophisticated routing functions (eg, location awareness, content addressing, etc) that are well beyond what is available from standard routing protocols, and; (b) the unpredictability of the radio channel combined with user mobility pose major challenges to routing and to connectivity. The preferred strategy to overcome these problems is to implement customization functions in upper layers and P2P networking overlays while keeping the basic routing and transport protocols simple.

As an example of MANET overlay, consider a “delay tolerant” file sharing application that includes hosts partly in the Internet and partly on ad hoc “opportunistic” network extensions. Wireless nomadic users can rapidly change their connectivity to the Internet from Kbps (say GPRS) to Mbps (say, 802.11). Temporarily, the users may also become disconnected. The use of the standard network routing protocols may lead to inefficiencies, violation of delay constraints and possibly retransmission of large portions of the file. A P2P overlay network instead can keep track of connectivity among the various hosts. The overlay network can extend to wired, wireless and ad hoc network segments. It can predict disconnection/reconnection dynamics and can exploit them to deliver files efficiently and within constraints (for example, using intermediate proxy nodes for “bundle” store-and-forwarding).

Another promising environment for the emergence of “opportunistic” ad hoc networking and P2P mobile middleware is the vehicle communication grid. Future cars will come equipped with radios (for safe navigation) and with plenty of on-board storage and processing power. Car to car communications will be enabled by a standard architecture derived from DSRC and promoted by IEEE and the Department of Transportation. Most importantly, cars will have a captive audience – the passengers – with plenty of time to burn! In the following section we describe a hypothetical

application for the vehicular grid, CarTorrent. CarTorrent, inspired by the Internet based BitTorrent distributed file sharing system, allows cars to partially download multimedia files from highway WiFi access points, and to cooperatively complete file assembly using a unique P2P mobile middleware solution.

6. CarTorrent: mobile middleware for vehicle networks

In this section, we present CarTorrent, a cooperative strategy for content delivery and sharing in future vehicular networks [23]. CarTorrent represents an interesting example of mobile middleware in a scenario that oscillates spatially from infrastructure supported to completely infrastructure-less. CarTorrent targets the problem of downloading files to a moving car from the Internet. CarTorrent aims to utilize efficiently the unused bandwidth between hot spots on the freeway. Without it, cars will have to park at the hot spot (kiosk) and wait until they get served. We study the issues involved in using such a strategy from the standpoint of Vehicular Ad-Hoc networks, or VANETs.

VANET applications will include on-board active safety systems leveraging vehicle-vehicle or roadside-vehicle networking. These systems may assist drivers in avoiding collisions. Non-safety applications include real-time traffic congestion and routing information, high-speed tolling, mobile infotainment, content delivery (as discussed here) and many others.

6.1. Content Delivery Techniques for Vehicular Networks

Future vehicular networks are expected to deploy short-range communication technology for inter-vehicle communications. In addition to vehicle-vehicle communication, users will be interested in accessing the multimedia-rich Internet from within the vehicular network. Kids sitting in the back seat of the car would like to play online games with their friends sitting at home, while Mom in the front seat might want to check out *www.cnn.com* and *www.sigalert.com* for the latest breaking news and the latest traffic alerts on all the major freeways. Within a limited radius, access to the Internet would be in the form of info-stations or Wi-Fi hot spots. We are focusing here on the content delivery application, where Internet content must be delivered to the user (upon request) within a certain time constraint.

Content can be obtained directly from the hot spot, but also from peers. Referring to the latter mode, “swarming” is a peer-to-peer content delivery mechanism that utilizes parallel download among a mesh of cooperating peers. Scalability is achieved since the system capacity increases with the number of peers participating in the system. The primary purpose of the protocols is two-fold: First, from the conventional *server perspective*: reduce the load of the origin server or the content publisher, and secondly from the *client perspective*: reduce the download time.

In the Internet, the above file content download and sharing procedure is embodied in BitTorrent, a popular file-sharing tool, which accounts for a significant proportion of Internet traffic. BitTorrent is a swarming peer-to-peer file sharing solution. Simply put, BitTorrent allows a single source to disseminate a single file to many users by having each user share what they just download. It can be used to share any type of file of nearly any size, with minimal bandwidth investment by the original distributor(s).

BitTorrent needs a few things to run: a client, a torrent, and a tracker. The client opens a .torrent file, chooses a location to save the file, and connects to the tracker. The tracker keeps track of how much each user is downloading and uploading, and what parts they have, and gives information to the client about where to get the next piece of the file. Note that BitTorrent downloads are in a mostly random order, although it prefers to get pieces that the fewest people have, so even if no one person has the whole file, every piece will be available. Also, the tracker watches your 'karma'; your download speed is tied to your upload speed, so if you aren't uploading much, you'll likely have a low download speed also. Thus BitTorrent builds its overlays by randomly selecting peers, a fact that can be potentially wasteful in a MANET environment.

6.2. A swarming protocol for vehicular networks

Consider a VANET with short-range communication technology. Given an average speed of 50 miles per hour and a gateway radio range of 500 meters, a simple calculation gives a car a transmission window to and from a fixed Internet access point on the order of a minute at the most. Taking into account contention from other cars, there may not be enough bandwidth to allow each user to download email, songs, as well as browse multimedia rich web-sites in the short time that they are connected to the gateway. Another practical issue is that on intercity highways, the gateways will be hosted by gas stations and food concessions, and thus will be less frequent; say every 5-10 miles. Thus the vehicle would be connected for about a minute to the Internet before being disconnected for around 5 minutes. As we shall see, the high mobility of nodes in VANETs coupled with the intermittent connectivity to the Internet provides an incentive for individual nodes to *cooperate* while accessing the Internet to achieve some level of seamless connectivity.

For the above reasons, an interesting problem is the design of cooperative protocols to improve client perceived performance of the vehicular network as a whole. The key contribution of CarTorrent is the development of P2P mobile middleware that includes the following features:

- (1) A gossip mechanism to propagate content availability information,
- (2) A proximity driven content selection/delivery strategy, and
- (3) Leveraging the broadcast nature of wireless networks to reduce redundant message transmission.

Before presenting the protocol, we define the network model and introduce some definitions. The network consists of a set of N nodes with same computation and transmission capabilities, communicating through bidirectional wireless links between each other. This is the infrastructure-less ad-hoc mode of operation. There are wireless gateways at regular intervals providing access to the rest of the Internet using infrastructure support (either wired or multi-hop wireless). The data unit for the swarming protocol is a *chunk*. That is, the content is broken up into equal sized *chunks* each with their unique identity. These chunks are shared and transferred among the peers. We assume each node is reachable from every other node.

CarTorrent has the same generic structure of any swarming protocol. Peers downloading a file form a mesh and exchange pieces of the file amongst themselves. However the wireless setting of VANETs, characterized by limited capacity, intermittent

connectivity and high degree of churn in nodes (cars) requires it to adapt in specific ways. Figure 9 and the pseudo-code describe the basic operation of the *CarTorrent* protocol.

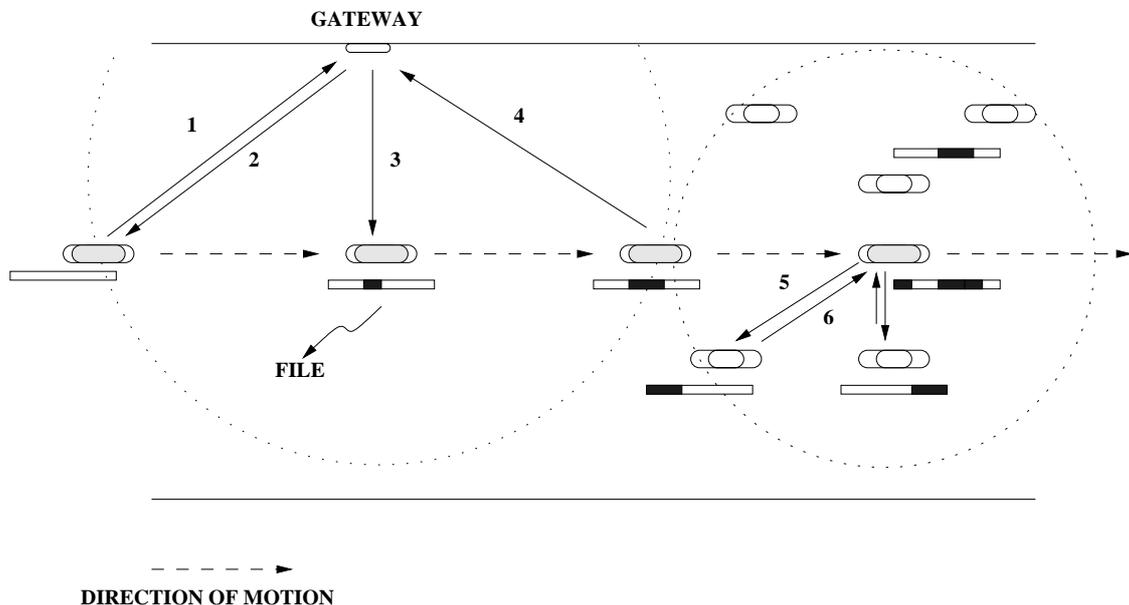


Figure 9: The basic operation of the CarTorrent protocol. A node (car) enters the radio range of a Gateway (1), initiates the connection (2) and starts downloading (3) pieces of the file. Once it goes out of range (4), it starts gossip (5) and discovers other peers/nodes with same content and starts exchanging pieces of the file (6).

There are several components to the operation of the *CarTorrent* protocol like Peer Discovery, Peer and Content Selection, and Content Discovery and Selection. In the sequel, for the sake of brevity, we provide just a simple, intuitive version of the protocol, referring to [23] for a more detailed discussion of the protocol and of the various options.

When a new car enters the vehicular network (such as entering a freeway or a section of freeway with access points), it requests the Gateway for the particular file. If the Gateway has the file in its cache, it starts uploading a chunk to the node. The node starts downloading chunks from the Gateway while it is in range. The Gateway also bootstraps it with a list of the last known peers (cars) who requested for the same file and when. Thus the car has an idea of how popular the file is and how likely it is to benefit from cooperative strategies.

Peers generate Gossip messages from time to time to advertise their presence and current content. A naive gossiping scheme has a potential of generating a large number of gossip messages as well as the problem of ping-pong of messages, where two peers keep exchanging stale data. We use a Gossip scheme inspired to methods that minimize “redundant forwarding” such as Minimum Connected Set forwarding, Passive Clustering or Multi Point Relay. Namely, only the “essential” set of neighbors forward the data/control packet for a specified number of hops. Forwarding nodes detect and suppress duplicates.

In the “simplified” swarming protocol, the newcomer, say node A, forwards upstream (in the direction of traffic) a gossip control packet with the list of chunks it needs. Selected intermediate nodes (the “forwarding” nodes for this file) turn on the forwarding

flag (according to the Passive Clustering scheme, say). The nearest peer, say node B (a few hops away) upon receiving the Gossip packet will respond with the first requested chunk. It also piggybacks its own current list. The forwarding nodes broadcast the chunk, which is thus propagated back to A. When A receives the first chunk it requested, it responds by transmitting in turn the first chunk that B requested (if any), and so on until B has received all the chunks it can possibly get from A. Basically, this is a “send/wait” protocol between A and B that is concluded when B has received all it needed from A. From this instant, the transfer is simply downstream, from B to A, until A and B have the same content. Typically, if the file is popular and the peer population dense, the transfer will be mostly downstream, from B to A. The reader will appreciate the fact that this swarming scheme requires chunk transmissions only between neighboring peers. Thus the download overhead is independent of network size and peer population; thus, the scheme scales to any network size. Since the basic scheme employs UDP transport and broadcast MAC, there is concern about potential congestion. To avoid congestion, rate control can be used. We refer the interested reader to details in [23].

6.3. The future of VANETs

Research in vehicular networks has made tremendous strides over the past decade. Prominent players like BMW, Daimler-Chrysler and Toyota are looking very carefully at this area to determine the right mix of ingredients which makes life easier for the driver reducing control or sacrificing privacy. Infotainment within the vehicle is again one of the grey areas, where it is difficult to determine when entertainment becomes distraction.

We envision the day when you are zipping down the highway listening to your favorite radio station when you hear a really good song. You hit the download button on your player. As you pass a gateway, the player initiates a CarTorrent download of the file. After you cross the gateway, your player starts gossiping with neighboring cars advertising your interest in the file. You also hear other cars advertising some pieces and start downloading pieces from them. In about 5-10 minutes, you’ve assembled all the pieces of the file with a combination of downloading through the gateway and exchanging pieces with your neighboring cars. From then on, you can keep playing that song until you get it out of your head. Until that day, research on vehicular networks will continue to strive towards getting information to the car faster, swifter, and better.

7. Conclusions

In this chapter we have reviewed two types of wireless network – infrastructure and ad hoc and have evaluated the impact of mobility. The two systems indeed present very different mobility models and problems. For the infrastructure the key issue is handoff; we have examined the model of the nomadic client that can connect to the infrastructure with multiple wireless interfaces (GPRS, UMTS, 802.11 etc) and must select the most convenient one to switch to. For the ad hoc environment one of the key issues is the design of routing algorithms that can scale and are also robust to mobility. We have in fact identified two different ad hoc scenarios and studied the routing problem for each. First, we have focused on the large scale automated battlefield scenario, where mobile middleware allows to recognize and exploit group motion, creating a robust hierarchical

routing solution based on landmarks. Then, we have shifted our attention to commercial applications. Here, we have studied the vehicular network scenario and have considered a file sharing application called CarTorrent. We have shown that even in this case the routing solution is highly dependent on coordinated car motion. Here again, mobile middleware is required to build a “routing” overlay that support “swarming” among the cars. In summary, mobility impacts “last hop wireless” (ie, infrastructure) applications in different ways than ad hoc networks. In both cases, however, mobile middleware is required to efficiently manage mobility.

References

- [1] M. Stemm and R. H. Katz. “*Vertical Handoffs in Wireless Overlay Networks*,” ACM MONET, 1998.
- [2] G. Dommety et al, “*Fast Handovers for Mobile IPv6*,” draft-ietf-mobileip-fast-mipv6-04.txt, IETF Internet draft, Mar. 2002.
- [3] R. Hsieh, Zhe Guang Zhou, and A. Seneviratne, “*S-MIP: a seamless handoff architecture for mobile IP*,” In Proceedings of IEEE INFOCOM 2003.
- [4] D. B. Johnson, C. Perkins, and J. Arkko, “*Mobility Support in IPv6*,” draft-ietf-mobileip-ipv6-17.txt, IETF Internet draft, May 2002.
- [5] K. El Malki et al, “*Low Latency Handoffs in Mobile IPv4*,” draft-ietf-monileip-lowlatency-handoffs-v4-03.txt, IETF Internet draft, Nov. 2001.
- [6] S. Deering, and R. Hinden, “*Internet Protocol, Version 6 (IPv6) Specification*,” RFC 2460, Dec. 1998.
- [7] C. Perkins, Ed. “*IP Mobility Support for IPv4*,” RFC 3344, Aug. 2002.
- [8] V. Ghini, G. Pau, P. Salomoni, M. Roccetti, and M. Gerla, “*Smart Download on the Go: A Wireless Internet Application for Music Distribution over Heterogeneous Networks*,” In Proc. of IEEE ICC, 2004.
- [9] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. “*SIP: Session Initiation Protocol*,” RFC 2543, March 1999.
- [10] D. Maltz, and P. Bhagwat, “*MSOCKS: An architecture for transport layer mobility*,” In Proc. of IEEE Infocom, p.p. 1037-1045, March 1998.
- [11] M. Schlaeger, B. Rathke, S. Bodenstern, and A. Wolisz, “*Advocating a Remote Socket Architecture for Internet Access using Wireless LANs*,” Mobile Networks & Applications, vol. 6, no. 1, pp. 23-42, January 2001.
- [12] A. C. Snoeren, “*A Session-Based Approach to Internet Mobility*,” PhD Thesis, Massachusetts Institute of Technology, December 2002.
- [13] R. Stewart et al. “*Stream Control Transmission Protocol*,” RFC 2960, Oct. 2000.
- [14] A. Matsumoto, M. Kozuka, K. Fujikawa, and Y. Okabe, “*TCP Multi-Home Options*,” draft-arifumi-tcp-mh-00.txt, IETF Internet draft, Oct. 2003.
- [15] L.-J. Chen, T. Sun, B. Chen, V. Rajendran, and M. Gerla, “*A Smart Decision Model for Vertical Handoff*,” In Proc. of the 4th ANWIRE, 2004.
- [16] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla, “*AdHoc Probe: Path Capacity Probing in Wireless Ad Hoc Networks*,” Technical Report TR050005, UCLA CSD, 2005.

- [17] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, “*CapProbe: A Simple and Accurate Capacity Estimation Technique*,” ACM SIGCOMM 2004.
- [18] UCLA Network Research Laboratory. <http://www.cs.ucla.edu/NRL/>
- [19] Guangyu Pei, Mario Gerla, Tsu-Wei Chen, “Fisheye State Routing in Mobile Ad Hoc Networks (2000)”. ICDCS Workshop on Wireless Networks and Mobile Computing, 2000
- [20] L. Kleinrock and K. Stevens, “Fisheye: A Lenslike Computer Display Transformation,” Technical report, UCLA, Computer Science Department, 1971
- [21] T.-W. Chen and M. Gerla, “Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks,” In Proceedings of IEEE ICC ‘98, Atlanta, GA, Jun. 1998,pp. 171-175
- [22] Kaixin Xu, Xiaoyan Hong, and Mario Gerla , “Landmark Routing in Ad Hoc Networks with Mobile Backbones,” Journal of Parallel and Distributed Computing (JPDC), Special Issues on Ad Hoc Networks, Feb. 2003, 110-123
- [23] A. Nandan, S. Das, et al, *Cooperative Downloading in Vehicular Ad Hoc Networks*, In Proc. of Wireless On-Demand Network and Services(WONS) 2005.