Dealing with node mobility in ad hoc wireless network

Mario Gerla, Ling-Jyh Chen, Yeng-Zhong Lee, Biao Zhou,
Jiwei Chen, Guang Yang, Shirshanka Das

Computer Science Department, UCLA, Los Angeles, CA 90095, USA

1 Introduction

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, and which, in other circumstances, can be organized to interconnect rescue crews after a Tsunami disaster, or yet can be structured to launch deadly attacks onto unsuspecting enemies.

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.

In this chapter we will be studying the “mobility attribute” of ad hoc networks and its impact on protocols and operations. We will study mobility under two different aspects, namely: mobility as an enemy that must be fought, and; mobility as a friend that can help us design more efficient networks. To be more precise, we will first address the damage that mobility causes in terms of path breakage and connectivity partitioning, and will review approaches to alleviate such effects. Then, we will observe that mobility can actually harnessed to overcome some of the very problems originating from mobility itself - for example, the fact that nodes move in groups can be exploited to achieve a highly scalable routing design not possible with individual random mobility. This will lead to define situations where mobility “helps”. An important factor that determines how dangerous or how helpful mobility can be is the mobility pattern. Thus, we advocate the need for accurate, comprehensive mobility models. Finally, we will introduce a case study, the Car Torrent, that illustrates the design of an application based on the mobility of cars in the urban grid.

Before we start, we will provide, in the next section, an overview of recent trends in the evolution of ad hoc networking. This section will serve as reference for the problems that will be introduced in later sections.

2 Setting the context: the evolving ad hoc network market

The concept of ad hoc wireless networking was born in the early 70’s, just months after the successful deployment of the ARPANET. It was the US Army who first discovered the potential of wireless packet switching for their mobile tactical operations. The first packet radio systems were deployed in the early 70’s, almost two decades earlier than any other cellular and wireless LAN technology. By the mid 70's, the packet radio concept was so familiar and so well understood that when Bob Metcalf (Xerox Park) came up with the Ethernet design in 1976, the word spread that this was a very ingenious way to demonstrate " ALOHA packet radio" technology on a cable!

In view of these early successes of the ad hoc network technology, one may ask how come there has not been any significant transfer of technology from military to commercial over the past 30 years. The main reason is that the original military applications scenarios had really nothing to do with "commodity" customers. The military solutions could not be easily adapted to commercial needs. In fact, until recently, the driving ad hoc network model has been the instant deployment in an unfriendly, remote infrastructure-less area. Battlefield, planet explorations, disaster recovery etc. have been an ideal match for that model. Early DARPA packet radio scenarios were consistently featuring dismounted soldiers, tanks and ambulances. If any transfer of the battlefield technology will occur, it will probably be to the areas of homeland security and civilian emergency recovery. In those applications, unmanned vehicles (UGVs and UAVs) are rapidly deployed in areas hostile to man, say, to establish communications before engaging agents and medical emergency personnel. Still, this is a far cry from commercial every day applications.

Yet, commercial ad hoc networking is finally emerging in many sectors of our society. Commercial networks will start small (one or two hops), as an extension of the wireless Internet infrastructure or of existing personal wireless LANs. They will not be very demanding in terms of QoS nor efficiency (they use unlicensed spectrum). Eventually, they may grow large (in fact, very large, in the urban grid case); but they do not need large size in order to attract the first customer! Early ad hoc networks will coexist and comply with whatever technology they will try to "opportunistically" extend, ie 802.11, Bluetooth, ZigBee, UWB etc. An important consequence of this trend (which is radically different from that followed by tactical networks) is the fact that commercial ad hoc nets will require new design criteria and new research in order to evolve efficiently. Some of the emerging commercial applications are:

- Indoor W-LAN extended coverage
- Hot spot (Mesh Networks) extensions
- Urban vehicle communications
- Campus networks
- Shopping malls, airport lounges

Of this set, we briefly review the last three scenarios, namely, the vehicle urban grid, the Campus network and the shopping malls. Starting with the **vehicle communications** scenario, cars communications today are mostly voice, to the fixed network, via the cellular system. The wireless data technology advances will stimulate an explosion of new car applications. To begin, **car to Internet** data communications will be greatly enhanced by **3G** and **mesh network** technologies. In addition to traditional Internet services, there are plenty of "mobile"

Internet applications for cars, from resource discovery (restaurants, shops, tourist attractions) to entertainment (movie previews) and navigation safety. Extending the wireless LAN and 3G connections (to negotiate, say, radio signal propagation obstructions) via opportunistic car to car multihopping will be simple and very cost-effective. **Within the car**, short range wireless communications (e.g., PAN technology) and ad hoc networking will be used for monitoring and controlling the vehicle's mechanical components as well as for connecting the driver's head-set to the cellular phone. However, by far the most innovative set of applications will be enabled by **car to car** communications. Potential applications include car to car road alerts, coordinated navigation, network video games, road "situation awareness" and other peer-to-peer interactions. To support these applications, an "opportunistic" **multihop** wireless network will evolve, which will connect cars among each other and to the wireless infrastructure spanning the urban grid and eventually extending also to intercity highways. This ad hoc network will alleviate the overload of the fixed wireless infrastructures (3G and hotspot networks), for example, by allowing direct car to car sharing of popular files. As shown in Fig. 1, the urban vehicle grid network can also offer an emergency backup in case of massive fixed infrastructure failures (e.g., terrorist attack, act of war, natural or industrial disaster, etc). The synergy of car multihop network, on-board PAN and cellular wireless infrastructure is a good example of **heterogeneous wireless network** aimed at cost savings, performance improvements and enhanced resilience to failures.

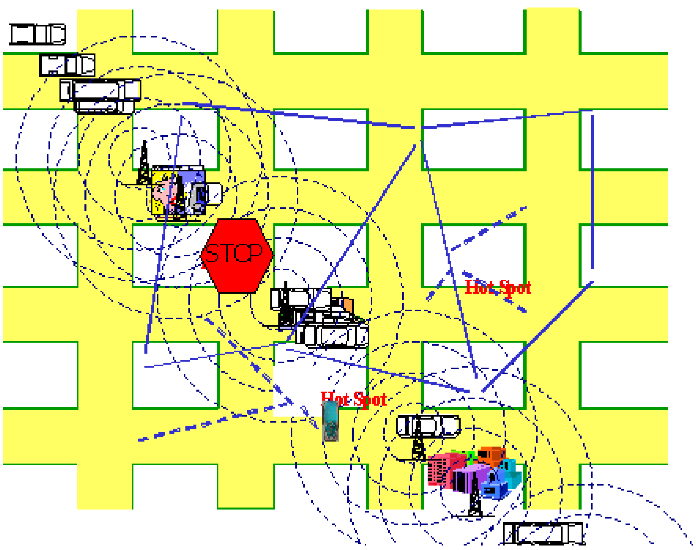

Fig. 1. The urban grid as an emergency network

The next scenario is the **Campus and shopping center**. For simplicity, we use the term "Campus" to refer to a place where people congregate for various cultural and social (possibly group) activities, thus including Amusement Park, Industrial Campus, Shopping Mall, etc. Today, on a typical Campus there are various wireless LAN access points in shops, hallways, street crossings, etc., that enable nomadic access to the Internet from various portable devices (e.g., laptops, notebooks, PDAs, etc.). Not all areas of a Campus or Mall are covered by department/shop wireless LANs. Thus, other wireless media (e.g., GPRS, 1xRTT, 3G, etc) may become useful to fill the gaps. There is a clear opportunity for multiple interfaces or agile radios that can automatically connect to the best available service. The Campus will also be ideal environment where group networking will emerge. For example, on a University Campus students will form small workgroups to keep track of their respective locations, to exchange files and to share presentations, results, etc. In an Amusement Park groups of youngsters will interconnect to play network games, etc. Their parents will network to exchange photo shots and video clips. To satisfy this type of close range networking applications, Personal Area Networks such as Bluetooth and IEEE 802.15 may be brought into the picture. As in the vehicular example, "opportunistic" ad hoc networking will extend access point coverage for Internet applications.

At this point it is appropriate to mention another type of wireless multi-hop network that is rapidly emerging in urban environments, namely, the **mesh network**. This architecture shares many of the characteristics of Wireless LANs and ad hoc networks at the same time. The MAC protocol is an extension of the IEEE802.11 family. The IEEE Committee is now working on the IEEE802.11s standard for such networks. Routing in the mesh is following the ad hoc network models (eg, AODV and DSR). Moreover, in an urban environment, mesh networks play an important role interconnecting the main types of commercial ad hoc networks we just described, namely, vehicular networks and the shopping malls. However, we must remember that the mesh network is actually an "**infrastructure**" network. It is a "permanent", public extension of institutional Wireless LANs. In this respect, the mesh network is similar to the cellular network - though faster to deploy. It would not be a surprise if in the future mesh networks will be regulated, tariffed and assigned a dedicated spectrum like other infrastructure networks. Yet, the design of mesh networks as multihop wireless architectures and topologies will radically differ from that of both the large scale battlefield designs and the smaller scale, very dynamic, opportunistic commercial network design. One can certainly argue that new design criteria and new research opportunities are present here as well.

Since our chapter is about mobility, the question is: how will this evolution from tactical, civilian networks to commercial networks affect our approach to mobility? The first answer is that in both scenarios the designer must protect the protocols from the damage of mobility (eg, path breakage, disconnection, etc). In this respect, the tactical and emergency networks are larger and have more stringent QoS requirements; thus, they will pose more challenges than the commercial counterparts. The second answer is that we also must look at opportunities to exploit mobility - namely, spontaneous, epidemic dissemination of information; flexible downloading from peers, etc. These opportunistic approaches generally make more sense in commercial than in tactical networks - in part because of the strict constraints of the latter. The reader should keep these considerations in mind while digesting the wealth of material presented in the rest of this chapter.

3 Motion considered harmful: how to protect the network

In this section, we view mobility as the “enemy”, i.e. as the cause of damage and disruption in the ad hoc network. We identify three challenges caused by mobility:

1. **Path breakage** - we must prevent packet loss, for instance presetting backup paths etc
2. **Topology control traffic overhead** - one approach to combat path breakage is to “update” the topology very frequently. But, this can have dangerous side effects
3. **Long lasting disconnections** - how can we cope with network partitions caused be motion? One approach will be to design delay and disruption tolerant network protocols.

We address these challenges in the following sections.

3.1 Preventing path breakage

To start, we observe that the most visible problem is **path breakage**: as nodes move the previously computed path fails. Here several techniques come to our help. First, we can “prevent” path breakage by predicting when a link on the path will break (**link prediction**) and computing an anticipatory backup path “before” the first path fails. We can also be cautious when we compute the path, and just choose a path that will connect nodes that are sort of associated with each other by a common motion, e.g, they “travel” together (i.e., **Associative Based Routing**). The third approach consists of using geo-routing. With **geo-routing** there is no notion of path, rather, of direction to destination. There no state kept in the network (as in popular proactive and reactive schemes). As long as intermediate nodes can forward to destination, the integrity of forwarding is preserved, even if a stable path cannot be maintained because of high mobility.

Link prediction In typical mobile networks, nodes exhibit some degree of regularity in mobility patterns. For example, a car traveling on a road is likely to follow the path of the road and a tank traveling across a battlefield is likely to maintain its heading and speed for some period of time. By exploiting a mobile user non-random traveling pattern, we can **predict** the future state of the network topology and provide a transparent network access during the period of topology changes. Moreover, by using the predicted information, we can reduce the number of control packets needed to reconstruct routes and thus minimize overhead.

In this section, we present mobility prediction to enhance unicast and multicast routing protocols. The proposed scheme utilizes GPS location information [1]. In our protocol, GPS position information is piggybacked on data packets during a live connection and is used to estimate the expiration time of the link between two adjacent nodes. Based on this prediction, routes are reconfigured before they disconnect. Our goal is to provide a seamless connection service by reacting before the connection breaks.

We assume a free space propagation model [2], where the received signal strength solely depends on its distance to the transmitter. We also assume that all nodes in the network have their clock synchronized; for example, by using the NTP (Network Time Protocol) [3]. or the GPS clock itself. Therefore, if the motion parameters of two neighbors (such as speed, direction, and radio propagation range) are known, we can determine the duration of time these two nodes will remain connected. Assume two nodes i and j are within the transmission range T of each other. Let (x_i, y_i) be the coordinate of mobile host i and (x_j, y_j) be that of mobile host j. Also let v_i and v_j be the speeds, and θ_i and θ_j $(0 \leq \theta_i, \theta_j \leq 2\pi)$ be the moving directions of nodes i and j , respectively. Then, the amount of time two mobile hosts will stay connected, D_t, is predicted by:

$$D_t = \frac{-(ab + cd) + \sqrt{(a^2 + c^2)r^2 - (ad - bc)^2}}{a^2 + c^2} \tag{1}$$

where $a = v_i \cos\theta_i - v_j \cos\theta_j$, $b = x_i - x_j$, $c = v_i \sin\theta_i - v_j \sin\theta_j$, and $d = y_i - y_j$. Note that when $v_i = v_j$ and $\theta_i = \theta_j$, D_t becomes ∞. The predicted value is the link expiration time (LET) between the two nodes.

Based on the above mechanism, we propose Distance Vector with Mobility Prediction (DV-MP) [4]. The protocol uses the route expiration time as the metric in the route table. Triggered update transmissions are eliminated because routes are established based on stability. Hence, routing update interval is relaxed and frequent updates are not required. In addition, using stable routes minimizes the disruption caused by mobility since a different route with a greater expiration time is used prior to a given route gets disconnected.

To utilize the prediction information, the mobility vector field is appended to the route update packet. In addition, the route expiration time (RET) metric is inserted into the routing table entry. Each node periodically broadcasts a route table. A sequence number is issued when generating updates, and it is incremented after each route table broadcast. The sequence number is associated with routing table entries for a particular origin of the route update. When node A receives a route table from its neighbor node B, the LET between nodes A and B is calculated based on the mobility vector contained in the received route table. Node A's route table is updated with the following rules:

- If an entry for destination D with a better RET is received and the received sequence number is greater than or equal to the old entry's sequence number, node A's entry for destination D is updated.
- If an entry for destination D with a higher sequence number is received, node A's entry for destination D is updated.

Fig. 2 illustrates the route table updating process. Values shown next to each link are LETS. In Fig. 2(a), node A's next hop to node D is node E and the RET through node E is 1. After node A receives the route update packet from node B, it updates its next hop for destination D to node B as shown in Fig. 2(b) since the route via node B has a higher RET value of three.

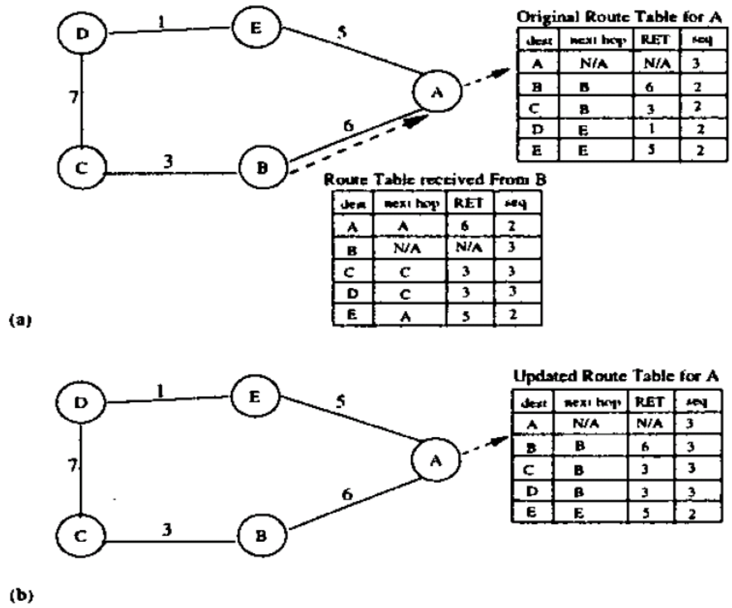

Fig. 2. A routing table update example.

There is a tradeoff between route distance and route stability. A route that has the largest RET will remain connected the longest, but may not have the shortest hop and/or delay.

Associativity-Based Routing The problem at hand is to devise a scheme to compute routes that can adapt well to link changes. Conventional distributed routing schemes attempt to maintain consistent routing information in the face of motion by performing ever more frequent link and topology updates. These however are undesirable as they result in high transmission overhead (we will address overhead reduction strategies in a later section). An approach that attempts to overcome this problem is the Associativity Based Routing algorithm (ABR) [5]. ABR is based on a new metric called degree of association stability.

Every node monitors its ‘Associativity’ with its neighbor nodes in order to determine the best route. Stability is determined using beacon “ticks”. Each node periodically transmits a beacon. The higher the number of ticks heard, the more stable the neighbor is. If all the mobiles along the route have high associativity ticks, the route is stable. The associativity ticks are reset when the neighbors or the node in question move out of proximity, not when the communication session is completed and the route made invalid. The distinctive feature of ABR, a unicast MANET routing protocol, is the use of ‘associativity’ as a primary metric in order to select more stable and thus long-lived routes. ABR is on-demand and maintains only the information for the desired routes. The route maintenance algorithm allows locally reconstructing a subsection of the route, instead of the entire route. There is no need for periodic route updates.

Geo-routing and TCP On-demand routing involves no periodic exchanges of route information but instead establishes routes when needed by flooding a route request to the network. This approach works well for small and moderate sized systems, and for large systems with relatively stable routes and limited communication patterns with significant destination locality. However, for large systems with bursty any-to-any communication patterns, and for systems with fast moving nodes, the overhead (and latency) of route discovery can become significant. In these cases, an interesting alternative is Geo-Routing. Geographic routing uses nodes locations as their addresses, and forwards packets (when possible) in a greedy manner towards the destination. Geographic routing is scalable, as nodes only keep state for their neighbors, and supports any-to-any communication pattern without explicit route establishment. Since there is no explicit path establishment, the scheme is by definition robust to path breakage. Namely, for successful forwarding it suffices that a neighbor is present in the direction to destination.

Geographic routing has been shown to dramatically improve the performance of TCP in ad hoc networks. Suppose that we need to set up a TCP connection over a mobile ad hoc network. As mentioned earlier, node mobility breaks routes forcing conventional routing schemes such as DSR or AODV to flood the network with discovery messages. Flooding is costly, and frequent flooding can saturate the network and thus degrade performance. A packet may experience significant long delay before discovering a new route. If the delay is larger than the RTO at the sender, the TCP sender times out, retransmits a packet, and experiences low throughput. On the other hand, if route discovery or reply message get lost, the packet will be dropped. In either case, the result is reduced TCP throughput. In a network where the route must be frequently recomputed due to node mobility, TCP will never get an opportunity to transmit at the optimal state and the congestion window will always be significantly small [6].

Since a node in geo-routing listens for neighbor's position updates, it passively estimates the moving velocity of its neighbors and infers immediately whether a neighbor is still reachable when transmission failure occurs. If the neighbor is estimated within transmission range, the packet could be "salvaged", ie transmitted to this neighbor again, or to other neighbors. This local repair is free of overhead and efficient. It is a network layer solution to flow contention and to random loss. With local repair, the packet transmission is robust to High BER, and can be lost only due to buffer overflow or route failure.

To demonstrate the efficacy of Geographic Perimeter Stateless Routing (GPSR) we run TCP-NewReno over GPSR [7]. The experiment consists of 20 nodes randomly placed within a 1000mx1000m area. Each node will continuously move within this area with speed randomly selected from 0 to 50m/s for 400 seconds. Random waypoint model is used for node mobility. In all simulations, we only show one TCP connection for clarity. The plot, as shown in Fig. 3, clearly shows that TCP over GPSR achieves much higher throughput than TCP over AODV.

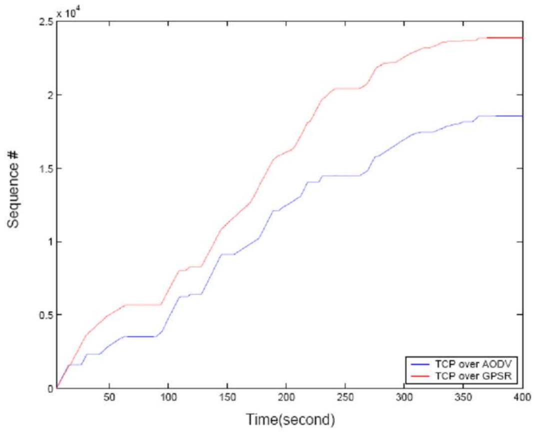

Fig. 3. Sequence Number of TCP over GPSR and AODV in Mobile Ad hoc Network.

3.2 Minimize the control O/H caused by motion: Fisheye State Routing

As it became apparent in the previous sections, a common way to combat mobility is to refresh the routing tables very frequently. This however leads to another problem, namely, loss of performance due to high overhead. A solution to the refresh overhead problem is provided by Fisheye State Routing (FSR) [8] [9]. FSR introduces the notion of multi-level fisheye scope to reduce routing update overhead in large networks. Nodes exchange link state entries with their neighbors with a frequency that depends on distance to destination. From link state entries, nodes construct the topology map of the entire network and compute optimal routes.

FSR is an implicit hierarchical routing protocol. It uses the “fisheye” technique proposed by Kleinrock and Stevens [10], where the technique was used to reduce the size of information required to represent graphical data. The eye of a fish captures with high detail the pixels near the focal point. The detail decreases as the distance from the focal point increases. In routing, the fisheye approach translates to maintaining accurate distance and path quality information about the immediate neighborhood of a node, with progressively less detail as the distance increases.

FSR is functionally similar to LS (Link State) Routing in that it maintains a topology map at each node. The key difference is the way in which routing information is disseminated. In LS, link state packets are generated and flooded into the network whenever a node detects a topology change. In FSR, link state packets are not flooded. Instead, nodes maintain a link state table based on the up-to-date information received from neighboring nodes, and periodically exchange it with their local neighbors only (no flooding). Through this exchange process, the table entries with larger sequence numbers replace the ones with smaller sequence numbers. The FSR periodic table exchange resembles the vector exchange in Distributed Bellman-Ford (DBF) (or more precisely, DSDV [11]) where the distances are updated according to the time stamp or sequence numbers assigned by the node originating the update. However, in FSR link states rather than distance vectors are propagated. Moreover, like in LS, a full topology map is kept at each node and shortest paths are computed using this map.

In a wireless environment, a radio link between mobile nodes may experience frequent disconnects and reconnects. The LS protocol releases a link state update for each such change, which floods the network and causes excessive overhead. FSR avoids this problem by using periodic, instead of event driven, exchange of the topology map, greatly reducing the control message overhead.

When network size grows large, the update message could consume considerable amount of bandwidth, which depends on the update period. In order to reduce the size of update messages without seriously affecting routing accuracy, FSR uses the Fisheye technique. Fig. 4 illustrates the application of fisheye in a mobile, wireless network. The circles with different shades of grey define the fisheye scopes with respect to the center node (node 11). The scope is defined as the set of nodes that can be reached within a given number of hops. In our case, three scopes are shown for 1, 2 and > 2 hops respectively. Nodes are color coded as black, grey and white accordingly. The number of levels and the radius of each scope will depend on the size of the network.

The reduction of routing update overhead is obtained by using different exchange periods for different entries in routing table. More precisely, entries corresponding to nodes within the smaller scope are propagated to the neighbors with the highest frequency. Referring to Fig. 5, entries in bold are exchanged most frequently. The rest of the entries are sent out at a lower frequency. As a result, a considerable fraction of link state entries are suppressed in a typical update,

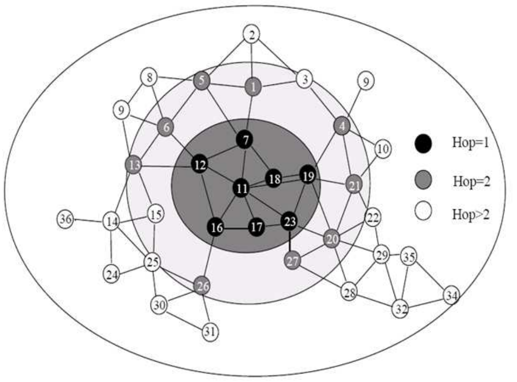

Fig. 4. Scope of fisheye

thus reducing the message size. This is exactly how FSR can handle high mobility with low O/H. In fact, FSR produces timely updates from near stations; it ignores the detailed motion of far nodes, and creates large latencies in the propagation of updates from stations afar. However the imprecise knowledge of the best path to a distant destination is compensated by the fact that the route becomes progressively more accurate as the packet gets closer to destination. As the network size grows large, a "graded" frequency update plan must be used across multiple scopes to keep the overhead low.

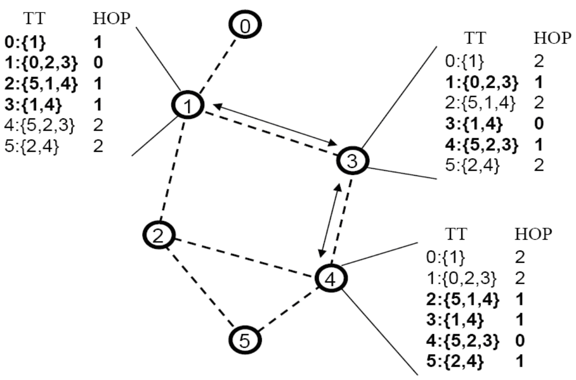

Fig. 5. Message reduction using fisheye

Through updating link state information with different frequencies depending on the scope distance, FSR scales well to large network size and keeps overhead low without compromising route computation accuracy when the destination is near. By retaining a routing entry for each destination, FSR avoids the extra work of "finding" the destination (as in on-demand routing) and thus maintains low single packet transmission latency. As mobility increases, routes to remote destinations become less accurate. However, when a packet approaches its destination, it finds increasingly accurate routing instructions as it enters sectors with a higher refresh rate.

3.3 Dealing with long term disconnections: Disruption Tolerant Networking

There have been theoretical studies on the node density required for connectivity [12]. Low density and mobility combined can easily cause long lasting disconnections. When the ad hoc mobile network becomes partitioned, one may attempt to "reconnect" it, or simply to cope with the "temporary" disconnection. Several approaches have been proposed to reconnect the network. Li and Hou investigated how to deploy as few additional nodes as possible to improve the connectivity in [13]. The problem was formulated as NP-complete and heuristically solved with triangulation-based algorithms. Zhao et al proposed another solution called Message Ferrying to connect partitioned sub-networks [14]. Message ferries are special nodes that, with their mobility patterns and trajectories under control, relay packets between network partitions. Message Ferrying shares a similar idea with the work of Delay-Tolerant Networking Research Group [15] and DARPA Disruption Tolerant Networking [16] in that data is bundled and transmitted when intermittent connectivity is available.

In this section we outline another systematic solution to network partitioning, incorporating the concept of Disruption Tolerant Networking (DTN) with peer-to-peer (P2P) overlays [17] [18] [19] [20]. The main motivation stems from a paradox existing in current ad hoc networks. On the one hand, ad hoc networks are set up in emergency scenarios, e.g. a battlefield or disaster scene. Application data is in general important and cannot be lost. On the other hand, ad hoc networks often operate in an adverse environment and are much less reliable than the Internet or infrastructure-based wireless networks. To bridge the gap between application needs and network limitations, we propose to build a Disruption-Tolerant Storage (DTS) overlay on top of the ad hoc network.

The DTS overlay consists of nodes equipped with large storage and powerful CPUs. These nodes form a P2P overlay network and jointly provide safe data storage to connections affected by network partitioning. More specifically, when the network is partitioned, the routing protocol will detect it and notify the source of each affected connection. The source node then sets up a conversation with the closest DTS overlay peer and submits the data files for storage. The submitted files are indexed using distributed hash table (DHT) indexing techniques and are replicated to a set of peers. Data is delivered when the connection between a DTS overlay peer and the original destination node is stored. Two methods of data delivery exist: either a DTS peer pushes data to the destination, or the destination submits a query to the DHT index and pulls data from the DTS overlay.

Benefits of the DTS overlay solution are summarized as follows. First, it exploits node heterogeneity in ad hoc networks by storing data at nodes with large storage capacities. Second, data is replicated across the overlay and therefore more robust to node failures. Third, DTS overlay peers can process stored data, e.g. transcode video to an appropriate bit rate, before delivering to the destination. The DTS overlay thus provides safe and flexible storage services to important data that would otherwise be dropped when the network is partitioned.

4 Mobility considered helpful: exploiting mobility to our advantage

In earlier sections we depicted mobility as a the necessary "evil" of the ad hoc network flexibility and selfconfigurability. We have shown ways to seek protection from path disruptions caused by mobility, and to make our applications operate correctly in spite of mobility. In reality, mobility can also be a "friend", in that it can be exploited to improve performance. In this section we show several examples of "friendly" mobility. To start, 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. Related to the concept of conveniently relocatable nodes is **data ferrying** using data mules. Finally, **last encounter routing** exploits node motion and gossiping to achieve free dissemination of information.

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) [9] and Landmark routing [22]. The key novelty is the use of landmarks for groups of nodes that move together (e.g., a team of co-workers at a convention or a tank battalion in the battlefield) in order to reduce routing update overhead. Like in FSR, nodes exchange link state only with their neighbors. Routes within Fisheye scope are accurate, while routes to remote groups of nodes are "summarized" by the corresponding landmarks. A packet directed to a remote destination initially aims at the Landmark, as it gets closer to destination it eventually switches to the accurate route provided by Fisheye. In [23], we introduce an enhanced version of LANMAR which supports landmark election and provides a flexible way for the protocol to cope with a dynamic and mobile network without compromising scalability.

Network Model and Data Structures Each node has a unique identifier, transmission range R, and landmark flag. Nodes move around and change speed and direction independently. An undirected link (i, j) connects two nodes i and j when the distance is less than or equal to the transmission R. For each node i, one list and three tables are maintained. They are: a neighbor list A_i, a topology table TT_i, a next hop table $NEXT_i$ and a distance table D_i. Each destination j within fisheye scope has an entry in table TT_i which contains two parts: $TT_i.LS(j)$ and $TT_i.SEQ(j)$. $TT_i.LS(j)$ denotes the link state information reported by node j. $TT_i.SEQ(j)$ denotes the time stamp indicating the time node j has generated this link state information. Similarly, for every destination j which is within its fisheye scope or which is a landmark node. $NEXT_i(j)$ denotes the next hop to forward packets destined to j on the shortest path, while $D_i(j)$ denotes the distance of the shortest path from i to j. The entries in next hop table $NEXT_i$ which point to landmarks form a new table called $LMDV_i$. Additionally, one or more link weight functions may be defined and used to compute the shortest path based on a specific metric, possibly with constraints. For instance, a bandwidth function can be used to support QoS routing. In this paper, we limit ourselves to mm hop paths, thus the link weight is 1.

Landmark Ad Hoc Routing Protocol(LANMAR) 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. The scheme is an extension of FSR. It 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 above landmark concept to FSR 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.

The routing update exchange of LANMAR routing is similar to FSR. Each node periodically exchanges topology information with its immediate neighbors. In each update, the node sends entries within its fisheye scope. It will also piggyback a distance vector of all landmark nodes. Through this exchange process, the table entries with larger sequence numbers replace the ones with smaller sequence numbers. As a result, each node has detailed topology information about its neighborhood and has a distance and routing vector to all landmark nodes.

Typically, all members in a logical subnet are within the scope of the landmark, thus the landmark has a route to all members. It may happen, however, that some of the members "wonder" outside of the scope because of lack of coordination in the group mobility pattern. To keep track of such "outsiders", i.e. to make a route to them known to the landmark, the following modification to the routing table exchange was made. Each node, say i, on the shortest path between a landmark L and an "outsider" member l of such landmark keeps a distance vector entry to l. Note that if l is within scope of i, this entry is already included in the vector. When i transmits its distance vector to neighbor j, say, then j will retain the entry for member l only if $d(j, l) <$ scope or $d(j, L) < d(i, L)$. The latter condition occurs if j is on the shortest path from i (and therefore from l) to L.

Landmark Election 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. The landmark information is a status pair containing the ID of the landmark and the number of group members it can reach within the FSR scope. When more than one node declares itself as a landmark in the same group, the node with the largest number of group members wins the election. In case of tie, lowest ID breaks the tie. The competing nodes defer.

After the first few topology updates, nodes near the center of a group will have enough group members in their table to qualify as landmarks. These nodes will take the role of landmark, and build their LMDV. The landmark status pair and the LMDV will be broadcast to neighbors with the next FSR exchange packet. When its non-landmark neighbors receive this update message, they will update their LMDV using the incoming LMDV. If a neighbor is a landmark itself, a winner competition is performed. The landmark status pair and LMDV at this node is set up corresponding to the competition result. The updated LMDV and the node's landmark status pair will be propagated again jointly with the routing update packets. When the last landmark change information reaches every node, only one node will remain as landmark for each group. The election converges quite rapidly. At steady state, a landmark propagates its presence to all other nodes like a sink in DSDV [11].

In a mobile environment, an elected landmark may eventually lose its role. The role shifting is a frequent event. In a transient period, there exist several landmarks in a single group. The transient period may be actually the norm at high mobility. This transient behavior can be drastically reduced by using hysteresis.

A great advantage of landmark election in LANMAR is recovery from landmark failures. In a dynamic network, nodes may die and come up. When a landmark dies, its neighbors will detect the silence after a given timeout. The neighbors of the same group will then take charge as landmarks and broadcast this new landmark information. A new round of landmark election then starts over the entire network.

In conclusion, LANMAR is an extension of Fisheye Routing which exploits group mobility by "summarizing" the routes to the group members with a single route to a landmark. LANMAR provides an efficient, scalable solution for wireless, mobile ad-hoc network as well as a dramatic reduction in route table storage overhead with respect to FSR leading to both line and storage overhead reduction.

4.2 GeoLANMAR: Geo Assisted Landmark Routing

GeoLANMAR routing protocol is based on LANMAR and extends LANMAR by applying greedy forwarding to route data packet to remote landmark nodes instead of relying on table driven forwarding using the routing info propagated by DSDV. The main advantage of Geo LANMAR over LANMAR is **additional robustness to mobility** (which is granted by the use of geo-forwarding - as discussed earlier). The geo-routing scheme is used to route packet to the remote landmark nodes (i.e. *geo-landmarks*) outside of the local scope. The number of landmark nodes is typically much smaller than the total number of nodes in the network. In GeoLANMAR, the geo-routing scheme offers much lower update rate required for advertisements and more robust forwarding for long distance routing, while local scope routing based on link state reduces update overhead.

In the greedy forwarding scheme applied to GeoLANMAR, intermediate nodes do not have to store routing tables to landmark nodes. They do not need to keep routing tables up-to-date either. The advantage of greedy forwarding is to permit dynamic adjustment for update rate of landmark routing packets. The dynamic update rate is determined by its movement and it offers GeoLANMAR a better scalability than LANMAR in terms of control overhead, delivery ratio, and throughput. Compared to geo-routing protocols, GeoLANMAR will overcome the inaccuracy of positions from the GPS devices since it uses link-state routing for packets near destinations. GeoLANMAR does not need any location service which is required by most geo-routing protocols.

A geo-landmark node is a special node dynamically elected by a group of nodes that are moving together (e.g., a rescue team). The Geo-landmark node propagates ID group, IP address and Geo-location to all other nodes in the network. As depicted in Fig. 6, Geo-Landmark LM transmits the information of its group to other nodes in the network.

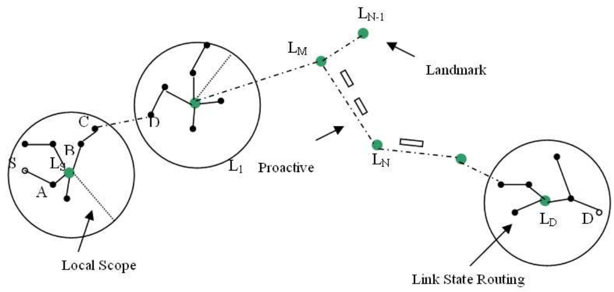

Fig. 6. Long distance Greedy forwarding applied through Landmarks as reference points to reach the destination D

Referring to Fig. 6, if the source S wants to communicate with mobile node D, it verifies whether the destination D can be reached immediately through the local link-state routing. If there is no entry found in local routing table, it tries to send the data packet toward destination D through geo-forwarding. By virtue of landmark distance vector advertising, the GeoLANMAR protocol can get the position of the destination node D without using a Location Server (normally required in conventional geo-routing protocols). From the group ID, and from geo-location of destination D, one can apply geo-forwarding by using the knowledge of the destination landmark. When the packet reaches the local scope of destination D, the data packet can be directly sent to D through the table-driven forwarding.

To perform such management, each node needs to maintain the following tables: a local topology table, a local routing table, and a landmark table with the geo-location information and the group IDs of all landmarks in the networks. When a node needs to send a packet outside its local scope, it checks its local topology table and selects as the next hop the nearest to the destination landmark node.

For the management of very large network with group mobility, the GeoLANMAR protocol seems to offer a good solution. The drawback of the protocol is the distance vector periodic updating, which is required in order to maintain accurate landmark tables. Fortunately, because the number of landmarks is much lower than the total number of nodes inside the networks, this protocol can get a good trade-off for large size networks with group motion.

4.3 Mobile Backbone Network

A Mobile Ad Hoc Network (MANET) is usually assumed to be homogeneous, where each mobile node shares the same radio capacity. However, a homogeneous ad hoc network suffers from poor scalability. Recent research has demonstrated its performance bottleneck through both theoretical analysis and simulation experiments and testbed measurements. Poor scalability is due to the fact that in ad hoc networks, most bandwidth of a node is consumed by forwarding packets. This is further exacerbated by heavy routing overhead of ad hoc routing protocols when the network size is large. This will significantly affect several large scale ad hoc applications, such as in a digital battle field, where hundreds or even thousands of nodes must be supported. Building a physically hierarchical ad hoc network is a very promising way to achieve good scalability. We present a design methodology to build a hierarchical large-scale ad hoc network using different types of radio capabilities at different layers in [24]. In such a structure, nodes are first dynamically grouped into multi-hop clusters. Each group elects a cluster- head to be a backbone node (BN). Then higher-level links are established to connect the BNs into a backbone network. Following this method recursively, a multilevel hierarchical network can be established.

Mobile Backbone Network The Mobile Backbone Network (MBN) is a hierarchical network in which a set of nodes functionally more capable than the ordinary nodes form the backbone. The basic scenario consists of a large numbers of mobile nodes deployed over a large area. Among these, the backbone nodes (BN) have the ability of forming multilevel backbone networks using long range radios. Usually, radios at each backbone level use some form of channel separation (eg, antenna directivity, different codes, different frequencies, or combinations thereof) in order to minimize interference across levels. Radios in the same level share the same frequency and channel resources. Unlike the wired network, the nodes in the mobile backbone network are also moving, thus the backbone topology is dynamically changing. In many scenarios such as the battlefield, the hierarchical structure is an inherent feature of the application. Different units have different communication devices and capacities. For example, the wireless radios installed in military vehicles have a more ample energy supply and thus are more powerful than those carried by the dismounted soldiers. Unmanned Aerial Vehicles (UAVs) and even satellites can be used for providing higher level and broader reach connections. Fig. 7 illustrates a three level hierarchy where the first level supports ground communications among soldiers; and second and third level are implemented using tanks and UAVs respectively. In this section, most of our discussions are based on a two level hierarchical architecture. However, the routing and clustering algorithms and protocols can be easily extended to multi-level hierarchical networks.

Hierarchical ad hoc networks have great potential in real time constrained applications, especially in the digitized battlefield. However, the backbone design is quite challenging if the nodes are mobile. Three critical issues are involved in building such a MBN, namely: optimal number of BNs, BN deployment and routing. In theory, a multi-level MBN can solve the scaling law problem observed in flat networks. However, MBNs with too many levels are not easy to operate and suffer from hardware limitations (e.g. each levels requires an additional radio). Thus, one generally opts for a MBN with a few levels (say, two) and must decide the number of BNs.

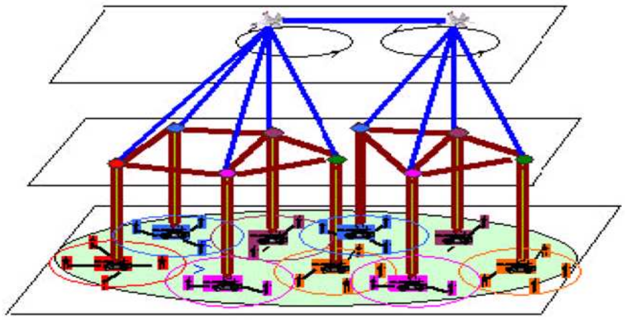

Fig. 7. Illustration of an ad hoc network with multilevel mobile backbones

After the number of BNs is decided, the second issue is how to deploy them. The main difficulties are mobility and BN failures. Using a clustering scheme to elect the BNs is a natural choice. Clustering has been widely used to form logically hierarchical networks [25] [26] and to partition a large scale network into small groups. However, a drawback of current clustering schemes is cluster instability, as indicated in many papers such as [25]. Conventional clustering schemes work effectively only in networks with very low mobility or no mobility at all, such as the sensor networks. Instability of the clusters and frequent changes of BNs introduce high routing O/H and make the hierarchy difficult to operate. In this paper, we will present a new clustering scheme to achieve good stability.

Routing is the third critical issue: The main requirement is to utilize the wireless backbone links efficiently and in a robust way. The main challenge of MBN routing with respect to the general Internet routing problem is mobility: address prefixes would need to be continuously changed as nodes move! The ensuing address management problem would be very complex and would offset the hierarchy advantages.

Backbone Node Deployment and Clustering After identifying the optimal number of BNs as a function of number of nodes and channel bandwidths, the second critical issue is how to achieve an optimal BN deployment. The simplest way is to pre-assign backbone nodes and scatters them uniformly across the field at initialization. However, such a static deployment has two main problems. First, the BNs are constantly moving. Thus after some time, some BNs may congregate in small geographical areas, creating congestion; while other areas may be depleted of BNs all together. This certainly is not a good scenario. The second concern is fault tolerance. BNs may fail or even be destroyed (a likely event considering the emergency applications envisioned for MANETs). New BNs should be deployed to replace the defunct ones. Static deployment cannot fulfill these requirements. Our solution is to deploy a large enough number of backbone capable nodes (ie, nodes with long range radios) and to dynamically elect a proper subset to BNs. When one BN is destroyed or moves out of a certain area, a new BN will be selected from the backbone capable node pool. If two backbone nodes move near to each other, one of them will give up its backbone role. The backbone node election is completely distributed and dynamic. It must result in a backbone node distribution that reflects the distribution of ordinary nodes. A Distributed Clustering algorithm is the most common approach to this problem, as described in [25] [26].

Ad Hoc Routing with Mobile Backbones Once elected, the BNs establish connections among each other using the long range radios. The next issue is routing. The routing scheme in the MBN has a main requirement: it must be able to exploit the high level backbone links, enhancing throughput and delay with respect to scheme without a backbone. It must do so without compromising (in fact, possibly enhancing) scalability and fault tolerance. In fact, considering the emergency recovery, unfriendly or even hostile environments where ad hoc networks are deployed, the backbone nodes can very possibly become disabled or may fail to operate. Maintaining connectivity in the face of backbone node failures is a strong requirement. Thus, the addressing and routing scheme cannot be totally "dependent" on the health of the backbone. For this reason, a cellular network like addressing and routing scheme will not work here. In a cellular network, the HLR/VLR (Home Location Register/Visiting Location Register) scheme will properly route the call request packet to the area where the roaming user has now registered. This requires that the Home Location of the user is up, and has a pointer to the Visited Location. In our Mobile Backbone Network where BNs disappear and come up very frequently, there is no reliable Home Location for any mobile. Redundant, robust Name Server schemes haven been recently proposed [19]. But they are not appropriate for our application, as their complexity would offset the advantages reaped by the hierarchical routing. To meet the challenges of our extremely volatile environment, we extend the Landmark Ad Hoc Routing (LANMAR) [21] [23] to operate in the MBN. We call this solution Hierarchical LANMAR Routing (H-LANMAR).

Hierarchical Landmark Ad Hoc Routing (H-LANMAR) LANMAR can be well integrated into the MBN by virtue of the fact that it is itself logically hierarchical. Routing information to remote nodes is summarized by landmarks. Now, we will extend such a logical hierarchical structure to utilize the physical hierarchy. In the original LANMAR scheme, we route the packet toward the corresponding remote landmark along a long multi-hop path. In the hierarchical MBN, we can route the packet to the nearest BN, which then forwards it through a chain of MBN links to a remote BN near the remote landmark. Finally, the remote BN sends the packet to the remote landmark or directly to the destination if it is within its scope. This will greatly reduce the number of hops. The procedure is illustrated in Fig. 8. We can see that by utilizing the backbone links, the 8-hop path is reduced to be 4 hops long, a great improvement!

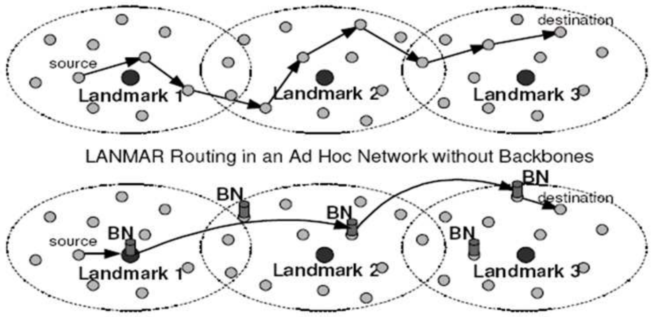

Fig. 8. Illustration of H-LANMAR routing in a MBN

We extend the LANMAR routing protocol so that it can take the "short cut" described above. First, all mobile nodes, including ordinary nodes and BNs, are running the original LANMAR routing via the short-range radios. This is the foundation for falling back to "flat" multi-hop routing if BNs fail. Second, a BN will broadcast the landmark distance vectors to neighbor BNs via the backbone links. The neighbor BNs will treat this packet as a normal landmark update packet. Since this higher level path is usually shorter, it will win over (and thus replace) the long multi-hop path in the level 1 network. From landmark updates the ordinary nodes thus learn the best path to the remote landmarks, including the paths that utilize the backbone links.

One important feature of our routing scheme is reliability and fault tolerance. The ordinary nodes are prevented from knowing the backbone links explicitly. The backbone links are indirectly learned via BN routing broadcasts. Now, suppose a BN of one group is destroyed by enemies, the shorter paths via this BN will expire. Then new landmark information broadcasted from other nodes will replace the expired information. Thus, in the worst case, routing in this group goes back to original landmark routing while other groups with BNs can still benefit from backbone links among themselves. When all backbone capable nodes are disabled, the whole network becomes a "flat" ad hoc network running the original level 1 LANMAR routing, which can still provide connectivity, yet at lower performance.

In this section, we presented schemes to establish and operate a "physical" multi-level hierarchical ad hoc network with mobile backbones (MBN). The optimal numbers of backbone nodes at each layer are derived through theoretical analysis. A stable multihop clustering scheme is also proposed to elect required backbone nodes and organize the hierarchical network. For efficient routing in such a hierarchical structure, we proposed to use an extension of the LANMAR routing scheme. The LANMAR routing solution is key to the feasibility and efficiency of the hierarchical structure. It is robust to mobility and yet reaps the benefits of the hierarchy. For example, backbone links are automatically selected by the routing scheme if they can reduce hop distance to remote destinations. Fault tolerance and system reliability of the proposed scheme have also been discussed. In essence, the proposed scheme combines the benefits of “flat” LANMAR routing and those of a physical network hierarchy.

4.4 Last encounter routing

In large-scale ad hoc networks, some or all the nodes may be moving. Therefore, the network topology changes with time. Routing algorithms have to base routing decisions on at least a partial knowledge of the network topology. The collection and exchange of topology information (e.g., distance vectors or link states) consumes valuable bandwidth and energy. A variety of routing algorithms have been developed that trade off the quality of routes, their computing and transmission overhead, and the degree of permissible mobility [27].

An elegant way of reducing this cost is by exploiting the *distance effect* [28]: basically the precision with which the destination location must be known to make a good, albeit suboptimal, routing decision, decreases with distance. If the node is far away from the destination, an imprecise estimate is sufficient, and vice versa. Routing schemes such as DREAM [28] exploit this effect to develop more “lazy” approaches to maintaining location information about all the nodes in the network. This approach essentially amounts to trading off a smaller location maintenance overhead, which is incurred continually with every topology change, for a slightly larger routing cost, as routes are in general suboptimal.

The authors of [29] go one step further and try to completely eliminate the cost to update location state. If nodes are not allowed to periodically relay any explicit location updates (as in Link State or Distance Vector), then the only topology information available at a node is the history of other nodes it has encountered in the past, i.e., nodes it has directly come into contact with. More specifically, we assume that every node remembers the time and location of its last encounter with every other node (i.e., when these two nodes last were directly connected neighbors; cf. Fig. 9). We call a routing algorithm a last encounter routing (LER) algorithm if at every node along a packet’s route, the next hop decision depends only on (a) the time and location of that node’s last encounter with the destination, and (b) auxiliary information carried by that packet. The main question we ask in this paper is the following: if all the nodes in the network are moving, is it possible for LER schemes to compute efficient routes, despite the absence of a location service? We show that, depending on the mobility processes, this is indeed possible. This is quite remarkable, given that LER invests no network capacity to track nodes, i.e., to maintain distributed location information.

The insight at the root of our investigation is the following. On the one hand, mobility of the nodes creates uncertainty about their location. On the other hand, consider some node d that is the destination of a packet. Some other node i that has encountered d in the past remembers the location of that last encounter. Three observations explain why LER routing can give rise to efficient routes: (a) the location of the last encounter is still a reasonably good estimate

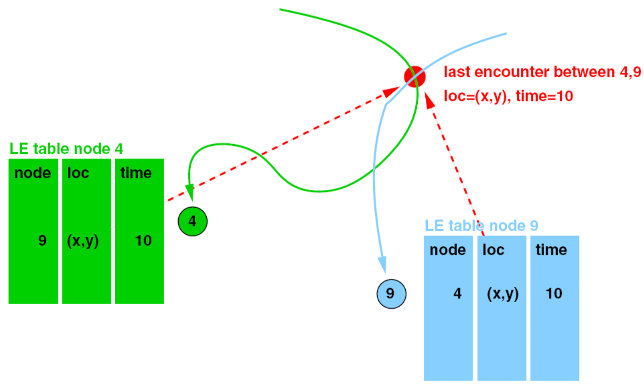

Fig. 9. A last encounter table in every node remembers the location and time of the last encounter with every other node in the network. In last encounter routing (LER), this table is queried by a packet to improve, if possible, its estimate of the location of its destination node.

of the destination's location after some time; (b) the time of that encounter, or equivalently, the "age" of the estimator, is a measure for the precision of that estimate; (c) node i's own mobility means that a recent estimate of d's position is available at some distance from d; given that d encounters other nodes all the time due to mobility, this essentially leads to a diffusion effect of noisy position estimates around d. The locality in the mobility processes inherently leads to a distance effect, in that better position estimates for d become available as a packet approaches d's current position.

Clearly, the feasibility of LER schemes will depend on the mobility process. If at any point in time, a node can jump uniformly over the entire surface of interest, an estimate based on the previous location is of no help. However, in the more likely scenario where the process has some locality, such as a random walk, then aged location information is useful, and diffuses at the same speed as the node moves itself. If the density of neighbors is sufficient both along the path of the destination node (so as to diffuse sufficiently) and along the path of a packet moving towards the destination (to get enough new estimates), then LER routing can work well.

4.5 Data ferrying with mules

Routing in ad hoc networks has been an active research field in recent years. However, most of the existing work focuses on connected networks where an end-to-end path exists between any two nodes in the network. In this section, we focus on mobile networks where nodes are sparsely distributed such that network partitions can last for a significant period. Sparse networks naturally arise in a variety of applications. For example, imagine the following hypothetical disaster scenario. A severe earthquake has occurred which collapses buildings, traps people in the debris, damages utilities and roads, and causes fires and explosions. Under this situation, the ability to communicate, even at low rates, is extremely valuable for sharing vital information (such as the number and locations of survivors, damages and potential hazards) and coordinating rescue efforts. However, providing communication capacity is difficult. First, fixed and stable communication infrastructure might be destroyed. Even if some infrastructure is usable, most rescue participants and victims may not have access to it. Second, available devices such as cell phones or PDAs can only communicate within a limited range. Due to the size of the area affected, a connected ad hoc network cannot be formed using these devices alone. To overcome partitions in sparse networks, a straightforward approach is to use radios with longer transmission ranges and maintain persistent network connectivity. However, since many mobile nodes use batteries for power supply, the use of a long range radio leads to excessive energy consumption. In addition, the availability of such devices in critical scenarios would be questionable.

To help overcome disconnection problems like discussed above, a Message Ferrying (MF) approach for data delivery in sparse networks is proposed [8]. Referring to Fig. 10, MF is a proactive mobility assisted approach which utilizes a set of special mobile nodes called message ferries (or ferries for short) to provide communication services for nodes in the network. Similar to their real life analog, message ferries move around the deployment area and take responsibility for carrying data between nodes. The main idea behind the Message Ferrying approach is to introduce non-randomness in the movement of nodes and exploit such non-randomness to help deliver data. Message ferrying can be used effectively in a variety of applications including battlefields, disaster relief, wide area sensing, non-interactive Internet access and anonymous communication. For example, in the earthquake disaster scenario, unmanned aerial vehicles or ground vehicles that are equipped with large storage and short range radios can be used as message ferries to gather and carry data among disconnected areas. This enables rescue participants and victims to use available devices such as cell phones, PDAs or smart tags for communication.

While the previous paper [30] has studied the idea of Message Ferrying in networks with stationary nodes, [14] considers networks with mobile nodes. More specifically, [14] develops two variations of the MF schemes, depending on whether ferries or nodes initiate non-random proactive movement. In the Node-Initiated MF (NIMF) scheme, ferries move around the deployed area according to known routes and communicate with other nodes they meet. With knowledge of ferry routes, nodes periodically move close to a ferry and communicate with the ferry. In the Ferry-Initiated MF (FIMF) scheme, ferries move proactively to meet nodes. When a node wants to send packets to other nodes or receive packets, it generates a service request and transmits it to a chosen ferry using a long range radio. Upon reception of a service request, the ferry will adjust its trajectory to meet up with the node and exchange packets using short range radios. In both schemes, nodes can communicate with distant nodes that are out of range by using ferries as relays.

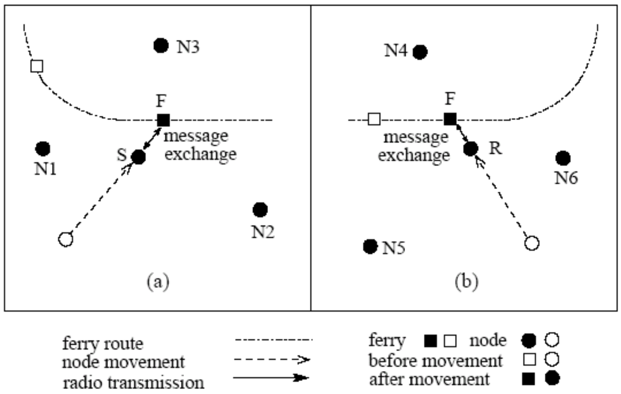

Fig. 10. An example of message delivery in the node-initiated MF scheme.

5 Mobility modeling: a Virtual Track Markov Chain approach

The mobility model is one of the most important factors that impact the performance of a mobile ad hoc network (MANET). Traditionally, the random waypoint mobility model has been used to model the node mobility, where the movement of one node is modeled as independent from all others. However, in reality, especially in large scale military scenarios, mobility coherence among nodes is quite common. One typical mobility behavior is group mobility. Thus, to investigate military MANET scenarios, an underlying realistic mobility model is highly desired. We recently proposed a "virtual track" based group mobility model (VT model) that closely approximates the mobility patterns in military MANET scenarios. It models various types of node mobility such as group moving nodes, individually moving nodes as well as static nodes. Moreover, the VT model not only models the group mobility, it also models the dynamics of group mobility such as group merge and split. The VT based mobility model is one of Markov Chain Driven Models. This model uses random seeds to determine the speed and direction of nodes.

Group mobility models have drawn a lot of interest recently. The mobility models proposed so far in the literature assume some kind of permanent group affiliation. Also they require that each node belong to a single group. In reality in a typical military scenario, a much more complex mobility behavior is observed. Some nodes move in groups; while others move individually and independently; a fraction of nodes are static. Moreover, the group affiliation is not permanent. The mobile groups can dynamically re-configure themselves triggering group split and mergence. All these different mobility behaviors coexist in military scenarios. A good realistic mobility model must capture all these mobility dynamics in order to yield realistic performance evaluation results, which, unfortunately, is not satisfactorily captured in any of the existing models.

We refer to the non-uniform, dynamic changing scenario described above as "heterogeneous" group mobility scenario. Here, different mobility behaviors such as group motion (including group merge and split), individual motion as well no motion can all coexist. Our proposed "virtual track" based group mobility model (VT model) handles all these heterogeneous mobility behaviors. In this model, a certain number of "switch stations" are randomly placed in the field. These stations are all interconnected by "virtual tracks" with given track width. Groups move along the virtual tracks towards the stations. At a station, a group can then be split into multiple groups heading to different stations (e.g. swarming). Groups entering the same station may also merge into one group. The individually moving nodes are then modeled as random moves (using the waypoint model) without the constraint of the virtual tracks.

The key idea of the proposed model is to use some "virtual tracks" to model the dynamics of group mobility. Some "switch stations" are first randomly deployed in the field. These stations are then connected via virtual tracks with given track width. The grouped nodes must move following the constraint of the tracks. At the switch stations, a group can then be split into multiple smaller groups; some groups may be even merged into a bigger group. Such group dynamics happen randomly under the control of configured split and merge probabilities. Nodes in the same group move along the same track. They also share the same group movement towards the next switch station. In addition, each group member will also have an internal random mobility within the scope of a group. The mobility speeds of these groups are randomly selected between the configured minimum and maximum mobility speeds. One can also define multiple classes of mobile nodes, such as pedestrians, cars, UGVs, and UAVs, etc. Each class of nodes has different requirements: such as moving speed etc. In such cases, only nodes belonging to the same class can merge into a group.

The proposed VT model is also capable to model randomly and individually moving nodes as well as static nodes (such as sensors). Such non-grouped nodes are not restricted by the switch stations and virtual tracks. Instead, their movements are modeled as random moves in the whole field.

Fig. 11 illustrates a main idea of the proposed virtual track based group mobility model. In this example, 5 switch stations are randomly placed in the field connected via 8 virtual tracks with equal track width. Group moving nodes are moving towards switch stations along the tracks. They split and merge at switch stations as shown in the figure. The black nodes in Fig. 11 represent the individually moving nodes and static nodes. They are placed and move independent of tracks and switch stations.

The proposed VT mobility model is suitable for both military and urban environment. In the battlefield, the "switch stations" can be viewed as the gathering points or hot spots of military forces. The virtual tracks are roads or trails or valleys connecting those hot spots. The troops usually move following the predefined track. In the urban environment, the virtual tracks can be viewed as the

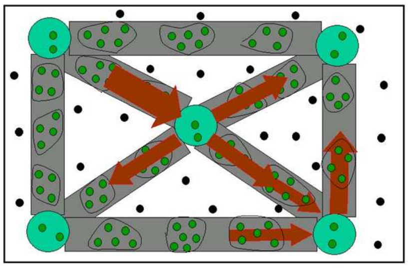

Fig. 11. Overview of Virtual Track Based Group Mobility Model.

streets. The switch stations are then the intersections of the streets. In a suburban scenario, the virtual tracks can represent the highways. The switch stations are then viewed as the inter-sections of the highway. The mobile nodes are then the cars running on the highway (e.g. under the constraint of the tracks). The convoys of cars on the highway can only split at the intersections.

Groups split and merge happen at the switch stations. Each group is defined with a group stability threshold value. When at the switch stations, each node in the group will check whether its stability value is beyond its group stability threshold value. If it is true, this node will choose a different track from its group. A group split happens. When several groups arrive at the same station and select the same track for the next movement, naturally, they will be merged into one bigger group.

Simulation experiments show that the performance is quite different under the track model and random waypoint mobility models. The virtual track model has better connectivity within groups and is less prone to geographic separation, route breakage and packet loss. But the track model has less radio space resource. The nodes are forced to share a restricted space and need a longer path to route data packets in the track model. There are more contention, collisions and congestion among nodes moving restriced in virtual tracks. The above constraints lead to performance degradation in the track based group mobility model. In contrast, individual randomly roaming nodes enjoy shorter paths and lower contention, which gives the random waypoint model a better performance.

From the simulations, it is also observed that the performance under the virtual track group mobility model can be enhanced by the introduction of individual nodes and static nodes. The reason is that the connectivity among multiple groups is increased by the roaming nodes outside of the virtual tracks. The above phenamon has a practical implication: the deployment of "relay" nodes in a group mobility environment can significantly improve performance, for example, the Mobile Backbone Overlay, etc.

6 Case study: Car Torrent - opportunistic file downloading in the urban grid

In this section we report a rather extensive case study that illustrates the limits, trade-offs and opportunities associated with mobility. The application is about downloading files to a moving car from the Internet. While going through the details of the "Car Torrent" protocol, the careful reader should notice that the car to car download, if done efficiently, allows to utilize the unused bandwidth between hot spots. In fact, this is an excellent example where car **mobility helps** expand systems capacity. Without Car Torrent, each car should park at the "hot spot kiosk" and wait until it gets served. With popular file distribution, this may easily exceed hot spot capacity!

6.1 Cooperative downloading in Vehicular Ad-hoc Wireless Networks

Future vehicular networks are expected to deploy short range communication technology for inter-vehicle communication. In addition to vehicle-to-vehicle communication, users will be interested in accessing the multimedia-rich Internet from within the vehicular network. This motivates a compelling application of Co-operative Networking in the Vehicular Ad-Hoc network (VANET) where the Ad Hoc network extends and complements the Internet.

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 contributions of CarTorrent are as follows:

1. A gossip mechanism to propagate content availability information,
2. A proximity driven content selection strategy (which takes into account the fact that transport-layer throughput degrades over multi-hop wireless connections), and
3. Leveraging the broadcast nature of wireless networks to reduce redundant message transmission.

6.2 Preliminaries

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). A unicast routing protocol is available to support packet transmissions between the network nodes. Nodes may or may not run the peer-to-peer application protocol. Nodes use TCP for reliable transfer of data and UDP for dissemination of *gossip* messages for content availability. 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. The terms chunks and pieces are used interchangeably throughout this article and have the same meaning. The problem is to design an application level protocol for vehicular ad hoc networks that disseminates data over this network in an efficient and scalable fashion and improves client perceived performance in the presence of transient connectivity.

We propose *CarTorrent* which builds on the fundamental mechanisms of partial downloading and sharing of content in *BitTorrent* but adapts to the wireless scenario by using different mechanisms for peer discovery, selection and content delivery.

6.3 The Protocol

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. Fig. 12 and the pseudo-code describe the basic operation of the *CarTorrent* protocol.

There are several components to the operation of the *CarTorrent* protocol like Peer Discovery, Peer and Content Selection, and Content Discovery and Selection.

6.4 Peer Discovery

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. Decision policies with respect to chunk choice are discussed later. 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. Thus the car has an idea of how popular the file is and how likely it is to benefit from cooperative strategies.

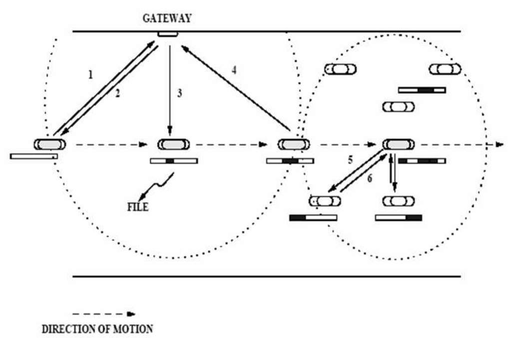

Fig. 12. Evolution of a file in a node using the SPAWN protocol. (1) A car arrives in the range of a gateway, (2) initiates a download (3) downloads a piece of the file. (4) After getting out of range, (5) starts to gossip with its neighbors about content availability and (6) exchanges pieces of the file, thereby getting a larger portion of the file as opposed to waiting for the next gateway to resume the download.

The centralized approach to peer discovery in *BitTorrent* has several disadvantages (beyond the most obvious disadvantage of having a central point of failure). In our scenario, the Gateway can only bootstrap an incoming peer with the last few peers that passed by and were interested in the same file. This set is too small for efficient sharing/downloading. We propose a decentralized mechanism for peer discovery to be carried out en route. We utilize the broadcast medium of the wireless channel to gossip information about the content availability at neighbors.

In *CarTorrent*, the centralized approach and the gossiping mechanism can be used in conjunction to construct the mesh of peers and update connectivity information continuously.

Gossip is the mechanism used to advertise the chunks that a particular peer possesses. A Gossip Message contains information to identify the file being distributed by the Gateway, and representing the list of chunks that the originator possesses, a timestamp indicating when it was originated, and a list of node-ids indicating which nodes processed it along the route. All nodes within range will hear it and process it depending on their type. We evaluate various gossiping schemes which we describe in this section.

Probabilistic Spawn Spawners not interested in the particular file listen to gossip messages of that file and forward them with a low probability. *Interested Spawners* listen to those gossip messages and forward them with a higher probability after stamping the route-list of the packet with their own id. An *Interested spawner* who is currently downloading a file will generate Gossip messages on completion of downloading a new piece.

Rate-Limited Spawn Each *Spawner* maintains two caches, a Non-Interested cache of gossip messages about files that it is not interested in, and an Interested cache. Periodically, gossip messages are picked up from one of the caches and re-broadcasted (without updating the origination time-stamp). Interested cache messages are selected at a higher frequency. The decision about which message to select from a particular cache can be made in different ways.

1. **Rate-Limited-Recent Spawn:** The gossip message with the most recent origination time-stamp is forwarded.
2. **Rate-Limited-Random Spawn:** The gossip message is selected at random from the relevant table.

6.5 Peer & Content Selection

TCP connections spanning fewer hops perform better in multi-hop wireless networks. To that end, *CarTorrent* does some intelligent Peer Selection based on the distance of the peer possessing a certain piece it intends to download. This information is gathered from the gossip messages. One could also gather this information from GPS enabled traffic-safety messages that are likely to become "standard" applications running on vehicles in the future. However we decided to keep our "inference" methodology independent of other applications that may co-exist on the same node.

We introduce a *proximity-driven piece selection* strategy. It uses several distinct strategies to choose which piece to download, based on how much has been downloaded already. Selecting pieces to download in an order that reduces contention at the peer serving the piece has a definite impact on performance as observed earlier. We employ several strategies that might perform better in the wireless setting. We estimate proximity based on hop-count. Other approaches to estimate proximity can be using ping messages to measure round-trip times, however this approach inadvertently introduces more delay and message overhead. Our hop-count based estimate performs well in a mobile wireless scenario. By bringing proximity awareness to content selection, users will experience seamless downloads. We refer the interested reader to a more detailed paper [31].

6.6 Design Rationale

It has been argued that the key deciding factor to whether a large ad hoc network is feasible is the locality of traffic. The effect of traffic locality determines to a large extent the scalability of per node capacity. *CarTorrent* tries to minimize the peer-side wall clock time taken to download a large file. *CarTorrent* like all swarming protocols is motivated by the fact that for popular files, the content distributor becomes the bottleneck as far as bandwidth and processing is concerned while the downloaders have ample spare capacity. In vehicular networks this problem is further exacerbated due to the intermittent and short-lived connectivity to the infrastructure. This form of cooperative data transfer encourages locality in network traffic and consequently scales while at the same time providing extended perceived connectivity. We prove it more formally using simulation and analysis in [31].

6.7 Simulation

In this section we describe the simulations we performed to evaluate the gossip schemes proposed. We implemented the gossip schemes in *Nab* a network simulator written in *Ocaml*. *Nab* [32] is a fast (For example, a 100 node simulation run for 300 simulated seconds completes in 4 minutes), flexible and scalable simulator for ad-hoc networks. We incorporated our mobility model, and a simple traffic model into the simulator. The car arrival process at the access point follows a poisson distribution with the average interarrival time varying from 0.5 to 4 seconds. We consider only one direction of vehicle motion in the highway. The peer group is maintained among cars driving in the same direction. When a car comes within range of the gateway, it starts downloading random pieces of the file. The tracker running on the gateway bootstraps the car with a set of 6 peers who last crossed that gateway and were interested in the same file. Each car possesses an initial speed which is varied at random by a small amount every 5 seconds. Cars maintain the same direction throughout and are not affected by the speeds of cars around them. The simulation parameters are as follows: File Size is 5MB, the piece size is 64KB and the velocity varies from 40-80mph.

We used a simplified version of the 802.11 DCF protocol implemented in the NAB simulator. In particular, the gossip messages are broadcast in the CSMA mode of 802.11. At the network layer we used AODV (Ad-Hoc On-Demand Distance Vector Routing). There are other on demand routing protocols such as DSR (Dynamic Source Routing) which can be potentially used in Vehicular Ad-Hoc Networks. Moreover, proactive routing protocols (e.g. OLSR) could also be used. The optimal choice of routing scheme is clearly an important issue. However since the focus of this paper is to evaluate application layer strategies, we will keep our study routing protocol agnostic. Leveraging routing protocol specific messages(for instance coupling our gossip messages with RREQ messages for efficiency purposes is part of continuing research effort). For channel data transfer rate we assume the typical 802.11a data transfer rate. This is a conservative assumption given that DSRC has a rate varying from 6-27Mbps. We are interested in the efficiency of the gossip schemes, the message overhead each scheme introduces. We analyze the impact of each of the simulation and traffic model parameters on the performance of the gossip schemes.

6.8 Analysis of Gossip Schemes

There are essentially three characteristics which we observe while evaluating the gossip mechanisms: (a) Good Peer Set Length: “Good Peer” is defined as the set of peers that are within k hops of a particular node. In all our simulations we set k to be 3. (b) Local File Downloaded (c) Peer-Space File Downloaded: the total fraction of the file that is present at a node and its Peer-List nodes.

We are interested in the Peer-Space File Evolution since this is the upper bound for the achievable fraction of the file for a particular car at a particular instant. The number of Good Peers in the Peer-List is a measure of the locality awareness of the peer discovery scheme. Figure 15 shows the evolution of the good peer list with the different gossip schemes at a typical node. The performance of a swarming protocol without any gossip clearly falls off as the peer starts moving away from the gateway. The various gossip schemes perform the same as far as the good peer set is concerned. The local File Evolution shown in Figure 16 for different schemes supports the intuition that gossip will help in retrieving more pieces of the file. The Peer-Space File Evolution in Figure 18 depicts that the gossip does enable robust peer discovery in the presence of high churn of peers.

6.9 Message Overhead

The advantages of gossip are clearly visible in the simulation results we presented. A natural question to ask is what is the cost of this robustness and location awareness? We ran simulations to analyze the Message Overhead of each of the gossip schemes. One of the simulation parameters that would have an impact on the overhead is the Forwarding Interval of the gossip messages. Figures 17 and 19 show that by varying the forwarding interval the overhead reduces considerably while still keeping the local File- Chunk evolution relatively stable. For our simulation runs, a forwarding interval of 1 second provided low message overhead and decent evolution rate.

6.10 Piece Selection Strategy

We experimented with three different piece selection strategies: *First Available*, *Rarest First* and *Rarest Closest*. First Available tries to fill the first empty chunk in the bit-field that can be filled. The search procedure is from low index to high, so lower index bitfields get filled up faster. Such a strategy is useful for files which have partial content usefulness. Some Mpeg files will play parts of the file if you have the partial file, so in these cases it would be advantageous to assemble the initial parts of the file first. Rarest First is the BitTorrent policy of searching for the rarest bit-field in your peerlist and downloading it. In wireless networks this could suffer from problems such as trying to download a rare piece from someone quite far away, while a slightly less rare piece is located very close to you. Connections to far away hosts are likely be unstable and lossy so we experiment with a variation of the rarest first scheme called Rarest Closest which weighs the rare pieces based on the distance to the closest peer who has that piece. Rare pieces which are situated closer to the node are preferred.

A node can guess the distance of a particular peer by looking at the gossip message of the peer, and calculating the number of nodes which have stamped the packet from the relevant field. Figure 20 shows the experienced download times for the three strategies; it is clear that Rarest Closest consistently gives shorter download times than Rarest First. First Available does the worst since it encourages determinism and reduces the entropy of the system.

6.11 Popularity Index

One of the critical factors in determining the download time of a file is its popularity. We varied the popularity index of the file (the percentage of cars

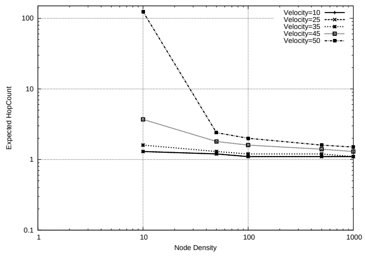

Fig. 13. Impact of Average Velocity on the Expected # of Application Hops needed to find a peer, with varying node densities

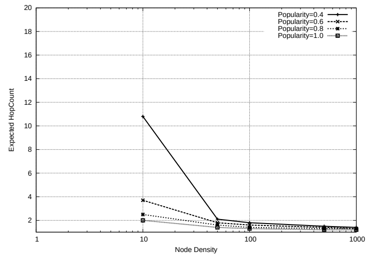

Fig. 14. Impact of Popularity of the File on the Expected # of Application Hops needed to find a peer with the file, with varying node densities

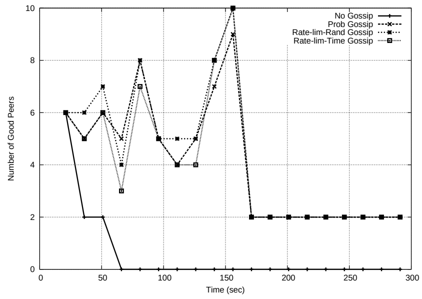

Fig. 15. Number of Good Peers

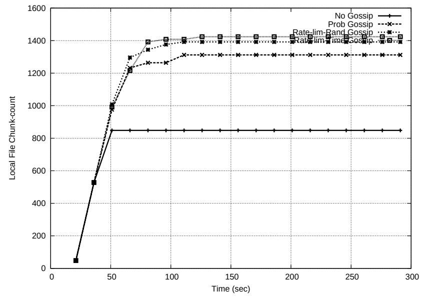

Fig. 16. Local File-Chunk Evolution

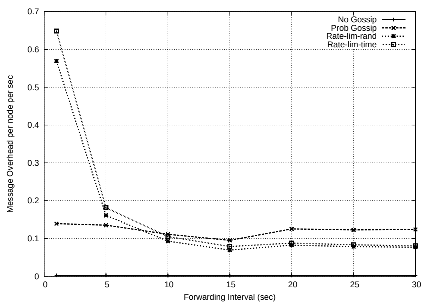

Fig. 17. Message Overhead with Forwarding Interval

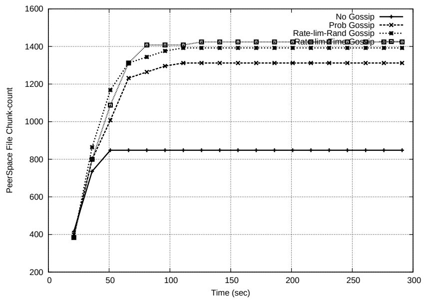

Fig. 18. Peer-Space File-Chunk Evolution

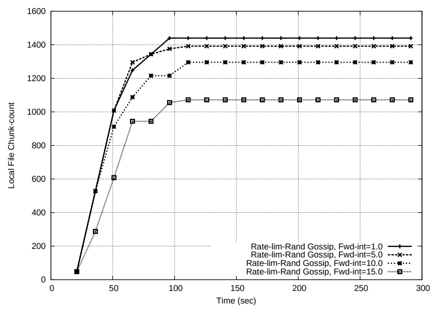

Fig. 19. Effect of Forwarding Interval on Chunk Evolution

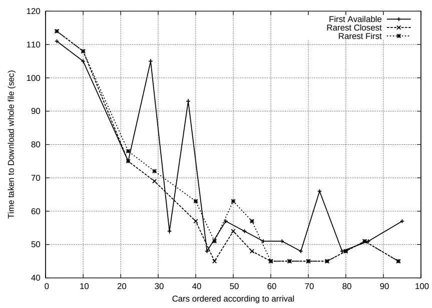

Fig. 20. Different PieceSelection Strategies

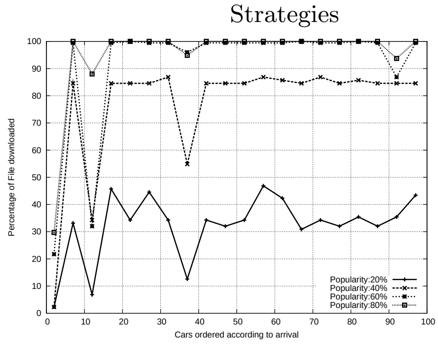

Fig. 21. Popularity works !

that are interested in this file) from 20% to 80%. Figure 21 shows the percentage of the file that is downloaded by the cars in the allotted 300 seconds time. It is clear that low popularity files are slow to download, however the speed ramps up pretty fast and gets bottlenecked by the capacity of the wireless channel at around 60% popularity. From there on, there are always “enough” new chunks for cars to keep downloading until they finish.

6.12 The Future of VANETs

We looked at vehicular ad-hoc wireless networks and how advances in peer-to-peer research can be adapted to these settings to improve the perceived throughput of the network. We gave a brief overview of the product of our research, the *CarTorrent* protocol which tries to achieve the design goals of scalability in wireless networks, improved perceived performance for individual clients using co-operation in highly mobile scenario. Research in vehicular networks has made tremendous strides over the past decade. Prominent players like BMW, Daimler-Chrysler, and Toyota, are looking at this area very carefully to determine the right mix of ingredients which makes life easier for the driver without taking away personal control or jeopardizing 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 and future trends

In this chapter we have looked at the “mobility attribute” of ad hoc networks under two different aspects, namely: mobility as an enemy that must be fought, and; mobility as a friend that can help us design more efficient networks. These two aspects are often intertwined, as shown in the Car Torrent case study. The exploitation of mobility to assist routing via epidemic dissemination and the use of “data mules” to help with long lasting disconnections are very novel, and perhaps a bit “controversial” concepts. The epidemic dissemination in particular requires the willingness of third parties to help as store and forwarders. Looking into the future, as ad hoc networks will move from battlefield to commercial scenarios, we will witness a shift from large scale, reliable, structured, preplanned operations to smaller scale, spontaneous, casual interaction between nomadic users. This will lead to an increase in popularity of “opportunistic”, epidemic, peer to peer routing and information dissemination schemes. In this opportunistic, “autonomic” world, it will be indeed essential to deal with mobility - starting with realistic individual and group models of mobility, and also make any possible effort to harness mobility to advantage. We expect that in future ad hoc network studies the mobility “variable” will receive at least as much attention as other important parameters such as time varying channel characteristics, traffic distribution, node distribution, Quality of Service requirements and energy constraints.

References

1. Kaplan, E.: Understanding the gps: Principles and applications (1996)
2. Rappaport, T.: Wireless communications: Principles and practice (1995)
3. Mills, D.: Internet time synchronization: the network time protocol. IEEE Transactions on Communications **39** (1991) 1482–1493
4. Su, W.: Motion Prediction in MobileNireless Networks. PhD thesis, UCLA, Computer Science Department (1999)
5. Toh, C.: Associativity-based routing for ad-hoc mobile networks. Wireless Personal Communications **4** (1997) 103–139
6. Liu, J., Singh, S.: Atcp: Tcp for mobile ad hoc networks. IEEE Journal on Selected Areas in Communications **19** (2001) 1300–1315
7. Karp, B., Kung, H.T.: Greedy perimeter stateless routing for wireless networks. In: ACM MobiCom. (2000) 243–254
8. Pei, G., Gerla, M., Chen, T.W.: Fisheye state routing in mobile ad hoc networks. In: ICDCS Workshop on Wireless Networks and Mobile Computing. (2000)
9. Pei, G., Gerla, M., Chen, T.W.: Fisheye state routing: A routing scheme for ad hoc wireless networks. In: IEEE ICC. (2000)
10. Kleinrock, L., Stevens, K.: Fisheye: A lenslike computer display transformation. Technical report, UCLA Computer Science Department (1971)
11. Perkins, C., Bhagwat, P.: Highly dynamic destionation-sequenced distance-vector routing (dsdv) for mobilie computers. In: ACM SIGCOMM. (1994)
12. Xue, F., Kumar, P.R.: The number of neighbors needed for connectivity of wireless networks. Wireless Networks **10** (2004)
13. Li, N., Hou, J.C.: Improving connectivity of wireless ad-hoc networks. Technical Report UIUCDCS-R-2004-2485, UIUC DCS (2004)
14. Zhao, W., Ammar, M., Zegura, E.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: ACM MobiHoc. (2004)
15. DTNRG: Delay tolerant networking research group. (http://www.dtnrg.org/)
16. DTN: Disruption tolerant networking. (http://www.darpa.mil/ato/solicit/DTN/)
17. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: ACM SIGCOMM. (2001)
18. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In: 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001). (2001)
19. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: ACM SIGCOMM. (2001)

20. Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D., Kubiatowicz, J.D.: Tapestry: A resilient global-scale overlay for service deployment. IEEE JSAC **22** (2004)
21. Pei, G., Gerla, M., Hong, X.: Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility. In: ACM MobiHoc. (2000)
22. Tsuchiya, P.F.: The landmark hierarchy: a new hierarchy for routing in very large networks. ACM Computer Communication Review **18** (1988) 35–42
23. Gerla, M., Hong, X., Pei, G.: Landmark routing for large ad hoc wireless networks. In: IEEE Globecom. (2000)
24. Xu, K., Hong, X., Gerla, M.: Landmark routing in ad hoc networks with mobile backbones. Journal of Parallel and Distributed Computing (JPDC), Special Issues on Ad Hoc Networks (2003) 110–123
25. Banerjee, S., Khuller, S.: A clustering scheme for hierarchical control in multi-hop wireless networks. In: IEEE Infocom. (2001)
26. Sinha, P., Sivakumar, R., Bharghavan, V.: Enhancing ad hoc routing with dynamic virtual infrastructures. In: IEEE Infocom. (2001)
27. Perkins, C.E.: Ad hoc networking (2001)
28. Basagni, S., Chlamtac, I., Syrotiuk, V.R.: A distance routing effect algorithm for mobility (dream). In: ACM MobiCom. (1998)
29. Grossglauser, M., Vetterli, M.: Locating nodes with ease: Last encounter routing in ad hoc networks through mobility diffusion. In: IEEE Infocom. (2003)
30. Zhao, W., Ammar, M.: Proactive routing in highly-partitioned wireless ad hoc networks. In: 9th IEEE International Workshop on Future Trends of Distributed Computing Systems. (2003)
31. Nandan, A., Das, S., Pau, G., Sanadidi, M.Y., Gerla, M.: Cooperative downloading in vehicular ad-hoc wireless networks. In: Wireless On-Demand Network and Services. (2005)
32. Ferriere, H.D.: Nab (network in a box). (http://nab.epfl.ch/)