Cross-Layer Mobile Chord P2P Protocol Design For VANET

Che-Liang Liu  
Department of Electrical Engineering  
National Taiwan University

Chih-Yu Wang  
Department of Electrical Engineering  
National Taiwan University

Hung-Yu Wei*  
Department of Electrical Engineering  
National Taiwan University  
E-mail:hywei@cc.ee.ntu.edu.tw  
Corresponding Author

Abstract: Efficient content distribution is one of the emerging applications in vehicular networks. To provide scalable content distribution in vehicular networks, Chord peer-to-peer overlay could be applied. Most P2P protocols, including Chord, are designed for wired-line network, and might perform poorly in mobile networks. Mobile Chord (MChord) is proposed to enhance the P2P performance over vehicular ad hoc network (VANET). In addition, cross-layer design to improve MChord performance in VANET is also investigated. Extensive NS-2 simulations with vehicular mobility traces are conducted to evaluate the P2P overlay performance in VANET. Mobile Chord and its cross-layer design outperforms the original Chord in various aspects, including application layer forwarding steps, query response ratio, correct query response ratio, and application delay.

1 Introduction

Vehicular networking technology has become one of the most promising communication infrastructures in many countries. Various safety applications and entertainment applications will be supported over vehicular networks. Content delivery will be one of the key applications in vehicular wireless networks. Application layer peer-to-peer overlay has a great potential to realize the efficient content delivery over vehicular networks (Lee et al., 2007). For example, pushing location-dependent advertisements to drivers through peer-to-peer overlay makes sense in terms of technology scalability and business model (Nandan et al., 2006).

Vehicular ad hoc network (VANET) is one of the promising architectures of vehicular networks. VANET shares some common properties with MANET (mobile ad hoc network). However, there are some distinctive differences between VANET and MANET: 1) vehicles move at very high speed but stop at intersections when traffic lights turn red, 2) since the route is restricted by the roads, the mobility is more predictable compared with the random movement of nodes in MANET. Furthermore, for public transportation, like buses, which always follows predefined routes and time schedule, could be predicted more accurately than general vehicles, 3) vehicles usually have unlimited power supply by its equipped generator while nodes in MANET are usually battery-powered.

Peer-to-Peer (P2P) technology offers an alternative communication architecture from traditional client-server architecture. Nowadays, the most popular usages of P2P technology are file-sharing and content delivery among users. In a P2P file-sharing network, users can publish and share their own possessions with others which are currently online. P2P users can also download interesting files from others. Unstructured or less-structure P2P protocols, as their names suggested, have no or a few restrictions on the structure of the overlay. Hence, it is difficult for users to locate the resources they need in an unstructured P2P network. In such case, flooding search seems to be the only method to locate an object. However, this approach usually results in so-called broadcast storm problem, which leads to significant performance degradation in VANET. On the other hand, structured P2P protocols, such as
Chord (Stoica et al., 2001), Pastry (Rowstron and Druschel, 2001), Tapestry (Ben et al., 2001) and CAN (Sylvia et al., 2001), are designed for locating objects effectively in wired networks. These structured protocols form a fully organized overlay with the proposed protocols. Logarithmic complexity is achievable for key operations such as joining, maintaining, or querying in the P2P overlay. Due to the attractive low overhead, researchers paid lots of attention on Chord. Researchers strive to enhance the overlay performance and to apply it to large scale deployment. Nevertheless, conventional P2P protocols usually assume reliable links among nodes. This is true in wired network but difficult to achieve in VANET due to the mobility, node failures, and packet corruption under error-prone wireless communication environment. Therefore, these protocols might perform poorly in VANET.

Chord (Stoica et al., 2001) is one of the most popular structured P2P protocols. Chord operates elegantly because of the simplicity of the design and the scalability to construct large-scale overlay network with low complexity. In an N-node Chord overlay, each node only has to keep contact just $O(\log N)$ nodes. A query can be done within $O(\log N)$ steps. Furthermore, nodes only need to maintain few local information (successor and predecessor) in order to guarantee the consistency of Chord. However, in VANET, the long forwarding routes of the query packet forwarding path might differ from the underlying ad hoc routes. Moreover, the slow convergence occurs when many nodes join the overlay concurrently.

This paper is organized as follows. Section 2 describes the related work. We elaborate our protocol design in Section 3. In Section 4, we introduce a cross-layer design to enhance proposed MChord. Results of intensive simulations under diverged scenarios are provided in Section 5. Finally, Section 6 concludes our work.

2 Related Work

2.1 Chord and Application-Layer DHT

Chord is an example of DHT-based P2P protocol. Fig. 1 shows an illustrative example of a Chord overlay network over VANET. Chord inherits many attractive features from DHT abstraction such as fault-tolerant, scalable, fully distributed, and self-organized. Every node and every object is hashed into a predefined m-bits key space using a global digest function such as SHA-1, MD4, or MD5. The message digest of the digest function is treated as a globally unique key of an entity (node or object). The key space and digest function must be selected carefully to make sure that with very high probability there does not exist two entities using the same identification. As Fig. 1 illustrates, each node is put on the ring wrapped at the possible maximum and minimum of key space. Each node X maintains its 1) successor, which is the node with the smallest key among keys larger than X’s, 2) predecessor, which is the node with the largest key among keys smaller than X’s, 3) a list of nodes, called fingers, to ensure each operation can be done within $O(\log N)$ steps. Each node is responsible for the ring segment [Key of X, Key of X’s successor). In that way, the work load can be averaged since the keys occupied by nodes should be randomly distributed in the key space. Each node must maintain fresh information about its successor and predecessor. The simple approach to do so is direct message exchange realized with ping messages. A consistent state of the P2P overlay at any given instant is that every node currently deems correct nodes as its successor and predecessor. The correctness of operations could be guaranteed by reaching consistent state because each operation with a specific key could be directed to the right node which is in charge of the key currently. Each node maintains a finger table which size is equal to the number of bits of the key space, $m$. The ith entry of a finger table records the first node of which key is larger than $Key + 2^{i-1}$, where $i = 1, 2, 3, \ldots m$. The logarithmic complexity is accomplished with finger table mechanism. Notice that neighbor nodes in the overlay are not necessary close in physical position.

2.2 P2P Protocol Enhancement

Researches show that P2P protocols in wireless networks might not perform as well as they do in wired networks (Cramer and Fuhrmann, 2006) (Oliveira et al., 2005). A protocol using epidemic flooding to locate objects outperforms Chord in terms of query response rate, which is one of the key performance metrics. They concluded that maintaining the overlay becomes impractical due to the uncertainty of node mobility and other characteristics of MANET.

MeshChord (Burresi et al., 2008) uses location-dependent address to reduce traffic for maintaining the overlay. With the help of location information, some maintaining messages can be limited in local area. However, each node must acquire the current identification of the destination node before actual application layer communication taking place. That waiting time until resolving current location-dependent address of some node may cause performance devaluation. Cramer et al. (Cramer and Fuhrmann, 2006) and Montresor et al. (Montresor et al., 2005) address the slow converging problem caused by the simple join procedure of Chord when many nodes join the overlay concurrently. They evaluate the problem and provide some enhancement in wired network. However, their proposed enhancement might not be practical in MANET and VANET. In addition, other enhanced mechanisms are applied to DHT to improve performance in MANET. Ahmed et al. (Ahmed and Shirmohammadi, 2007) deploys multi-level hashing technique to reduce routing overhead. Thomas et al. (Zahn et al., 2006) extends the DHT substrate to support range query in MANET.
2.3 Cross-layer design for P2P overlay

Ding and Bhargava (Ding and Bharat, 2004) investigated the design options under P2P over MANET scenarios. Overall system performance depends on both application layer overlay path selection and network layer routing. Five network architectures: broadcast over broadcast, cross-layer broadcast, DHT over broadcast, broadcast over DHT, DHT over DHT, and cross-layer DHT are compared and investigated.

CrossROAD (Delmastro, 2005), Ekta (Pucha et al., 2004) and its successor Ekta+ (Pucha et al., 2006) use cross-layer design to boost performance of Pastry. Ekta applied cross-layer design between network layer routing and application P2P. A modified Pastry P2P protocol is closely integrated with DSR routing. Routing hop count is used for evaluating node proximity in P2P overlay. Ekta+ improves Ekta performance by introducing DHT message aggregation at both network layer MANET and application layer overlay. By aggregating small DHT messages and exploiting reusable routes, Ekta+ could reduce overhead and enhance protocol efficiency. CrossROAD uses a cross-layer approach to integrate Pastry P2P protocol and proactive routing protocol, such as OLSR. P2P common API is created to support cross-layer design in CrossROAD. Network layer information, such as routing table and information of neighbor nodes, could be obtained through the API. Overhead is reduced with the cross-layer design.

DHT might be applied at network layer to establish multi-hop relay routes. Virtual Ring Routing (VRR) (Caesar et al., 2006) is a routing protocol which uses ring structure to route packets. VRR integrates DHT substrate into the routing protocol itself, so locating an object can be achieved by key-based routing in such networks. However, additional care must be paid to realize such service. VRR uses a global naming function so that the identification of a given node is unique and independent from current geographical location. A packet travels around in the network until it reaches a node which maintains the route to that key. MA-Chord (Qi and Hong, 2007) is a unicast routing protocol using a DHT substrate. MA-Chord uses cluster-based address. Cluster heads dedicate in managing overlay network to reduce overhead. While a process called address resolution must finish before actual data transmission, a force delay is occurred when the cluster-based address is not cached.

3 Mobile Chord Protocol Design

We design mechanisms to reduce protocol overhead and alleviate problems caused by node mobility and the frequent topological change.

1. Aggressive table update: try to use any available information to update finger table (for Chord) and overlay table (for MChord).

2. Overlay table broadcasting: broadcast P2P overlay information to neighboring MChord nodes instead of using unicast ping for keep-alive mechanism in Chord

3. Greedy forwarding: select the closest node in the overlay table rather than the best finger in the Chord finger table.

4. Passive bootstrapping: a new node learns the P2P overlay network information by listening to overlay table broadcasting rather than joining P2P overlay through hook nodes.

3.1 Aggressive Table Update

When a Chord node maintains the consistency of the P2P overlay, this node is expected to update the desired random-chosen finger only, as shown in the pseudo-codes in (Stoica et al., 2001). Due to the sparsity of keys occupied by P2P overlay participants, this behavior is not adequate. For a Chord node, fingers, especially the ones with lower indexes, may share the same node as their target. If we only update a finger once and remain the others intact, the other fingers which have the same target address will become obsolete when the target address of the updated finger changes. So we relaxed that update criterion. Whenever a fresh update shows up, Chord will update all possible fingers, successor and predecessor. Furthermore, whenever a packet is delivered to Chord, it will extract any applicable information and update fingers, successor and predecessor, if it is feasible. Since we believe this update mechanism is essential, we include this feature into both the baseline Chord and MChord.

3.2 Overlay Table Broadcasting

Each node maintains the Chord overlay in a distributed manner. A node periodically check whether other nodes are still online or not. This might be implemented by sending ping messages through unicast. Ping messages are small packets, which consume little resource in wired network. Nevertheless, in VANET, frequent sending small packets will severely degrade network performance. In VANET, sending a small unicast ping message might consume a large amount of radio resource because of the ad hoc routing protocol.

Assume a reactive routing protocol, such as AODV, is used. If we perform the application-layer ping as keep-alive mechanism before network-layer route expiration, a short interval between two consecutive ping events is needed. This frequent ping events lead to high traffic load. If we apply a long time interval between two consecutive ping events, the network-layer routes may be expired or no longer valid. Network-layer routing agents has to repair the routes with AODV route discovery process, which leads to heavy traffic load as well. In addition, infrequent keep-alive update will result in staled information and damage the consistency of the P2P overlay. As a result, the conventional ping-based keep-alive mechanism is not suitable for Chord over VANET scenario. Heavy signaling overhead is
created by routing agents when a ping task is conducted in VANET.

Unlike P2P operation in wired-line network, packets loss is not a good indicator to P2P node failure in VANET. In VANET, packet loss might due to various causes including 1) route is under a reparation, 2) a temporal network partition occurs due to node mobility, 3) the packet is not received due to high packet error rate in wireless network. As packet dropping may due to various reasons, applying unsuccessful ping event as explicit indicator for Chord node failure is inappropriate. So we devised the overlay table broadcasting mechanism which exploits the nature of wireless communication to address the problem. In wireless communication, any node around the source can overhear the packet. Overlay table broadcasting is very simple in concept. Every node maintains its own overlay table about other nodes which are currently in the P2P overlay. Each node broadcasts a list to its one-hop neighbors periodically. The list is a subset of its own overlay table. Whenever a node hears the broadcasted list, it will update its overlay table. If an entry in the overlay table has not been updated after a predefined time period, the entry is invalid and is purged from the table. The entry of the table can be implemented with a few fields, such as network address, the last update time, or network address, the latest sequence number.

Overlay table broadcasting uses localized one-hop communication instead of multihop unicast ping to save wireless network resource. Overlay table broadcasting can be regarded as an aggregation of ping messages. Overlay table broadcasting mechanism has the following advantages: 1) overlay table broadcasting is transmitted with one-hop wireless broadcasting and avoids multihop routing overhead 2) overlay table broadcasting message contains more information than the conventional ping packet.

### 3.3 Greedy forwarding

A Chord node selects next hop from its acquaintance about the network, that is, finger table. With the help of overlay table broadcasting, McHord application does not need to maintain finger table anymore. The overlay table is sufficient to maintain the P2P overlay and to forward an application packet. The forwarding criterion is simple. Each node forwards packets to the node of which identification is the closest to the desired destination nodes. By forwarding packet this way, the traffic load of forwarding query messages can be significantly reduced.

### 3.4 Passive bootstrapping

The performance of nodes joining a P2P network affects the consistency and efficiency of a P2P protocol. When a node wants to join the P2P overlay, it will try to connect to a predefined node called hook node. A list of candidate hook nodes is usually needed in conventional P2P joining procedures. However, having a list of predefined hook nodes is impractical in VANET scenarios. In addition, the traffic congestion around hook nodes occurs when multiple nodes try to join the P2P overlay simultaneously. Although the congestion is a temporary phenomenon, the unbalanced traffic load around the hook nodes is not desirable. We propose the passive bootstrapping mechanism to resolve this issue. When a node wants to join the overlay, it just starts up the P2P application. Then the new node sets a timer with a random timeout and sends an overlay table broadcasting to its 1-hop neighbors after the timer expires. The following overlay table broadcasting message is sent as usual. With the overlay table broadcasting technique, each node will get acquaintance with other nodes gradually. No predefined hook nodes are required. Overloaded traffic around the hook nodes is avoided.

### 4 Cross-layer enhancement for Mobile Chord

#### 4.1 Cross-layer Knowledge Harvesting

Wireless packet transmission is broadcasting in nature. Even if a wireless packet is intended to be transmitted to a specific destination, the other nodes in the vicinity may overhear that packet, too. Why not share a packet if it carries some information of common interest? By exploiting the broadcasting nature of wireless medium, we could facilitate the overlay information dissemination efficiently. Overlay table broadcasting is an exemplary design inspired by this philosophy. Overlay table is transmitted with one-hop broadcasting, which distributes the information to neighboring nodes without significant overhead of multihop routing or broadcast storms. In our design, individual overlay table broadcasting is always one-hop, no nodes will rebroadcast anything to prevent the broadcast storm problem.

To reach the efficient P2P over VANET design goal, a cross-layer design that exploits the broadcast nature of wireless medium and harvests the available information is illustrated in Fig. 3. MAC layer will forward all P2P relevant information to the application layer Mobile Chord. After a successful reception of a Mobile Chord packet, modified MAC will send the packet to Mobile Chord application whether the destination matches the identification of the node or not. After receiving the packets from MAC, Mobile Chord will extract whatever is useful to update its overlay table. For example, if a node overhears the query from another node, it will get new information about the initiator of that query. This valuable information might be useful for future P2P operation involving these nodes. Extracting information from packet destined to itself is the only way the conventional Chord update its information. With the cross-layer design, heard mobile Chord packets from neighboring node could be used for information update. Additionally, multihop relay is used in VANET. An intermediate relay node could extract information from the en route Chord packets.
4.2 Knowledge Piggyback In Overlay Signaling

Although overlay table broadcasting helps to propagate P2P information over the network, the speed of propagation may be slow. Through overlay table broadcasting, new information from some nodes reaches other distant nodes hop-by-hop through one-hop broadcasting. When new information finally arrives at a distant node, it may be out-dated and may be pruned off quickly. For example, assume the interval of overlay table broadcasting is 10s and the period that a table entry remains valid is 20s, the information propagation distance is around four hops on average and is only two hops in the worst case. For some large networks, overlay table broadcasting may takes too much time to propagate fresh information.

We propose the knowledge piggyback in overlay signaling mechanism. Piggyback overlay table on overlay signaling is simple to implement. Some P2P overlay packets, such as Chord maintenance signaling packets or query packets, may travel through a long physical distance. We piggyback the overlay table into these packets. When a piggybacked packet is received, the receiver not only processes information in the packet but also extracts the piggybacked overlay table and then updates its table. After the receiver updates its table with fresh information, it will rebroadcast these new data in subsequent overlay table broadcasting. As a result, the vicinal nodes around the receiver will get the fresh information, too. Although this technique increases the size of overlay signaling messages, the fresh overlay information could be propagated further and faster. From the normalized disseminated overlay information per unit overhead perspective, the proposed scheme is more efficient.

If we integrate cross-layer design and piggyback overlay table in the existing MChord signaling messages, the benefit will be significant. First, if a packet is supposed to travel across the network such as query packets, carrying something beneficial in that packet may not cause too much overhead compared to creating new packets, as neither additional ad hoc route discovery nor MAC layer contention is needed. Second, when a packet transmits hop-by-hop from the origin to the destination, the nodes around the route may overhear that useful information encapsulated in the packet by cross-layer approach. Third, the integration of the overlay table piggybacking with the cross-layer knowledge harvesting mechanism provides an efficient way of overlay information dissemination which might require frequent overlay table broadcasting that causes heavy overhead. Actually, the overlay table piggybacking disseminates information to a much wider range. In MChord, maintenance signaling messages are sent periodically to keep the overlay table up-to-date and to detect node failure. In our implementation, we piggyback the overlay table in the Chord maintenance signaling messages. Other P2P signaling message, such as query messages, could also piggyback with the overhead table. With cross-layer knowledge harvesting, all on route nodes and their neighbors will get fresh P2P overlay information.

5 Performance evaluation

We evaluate our design through intensive simulations under various environments. We use Network Simulator version 2.32 (McCanne and Floyd, 2006) to do detailed packet level simulations. Each mobile node uses built-in 802.11 MAC module and Wireless PHY without any adjustment of default parameters. The radio transmission range is up to 250m. We use AODV as routing protocol with default parameters. We implemented the Chord following the guideline with a little modification (see aggressive table update). Mobile Chord is implemented with all advanced features. The important parameters of Chord are shown in Table 1. Some parameters are determined by a uniform random distribution; hence, the number of concurrent updates and signaling events decreases. Unless otherwise specified, a hook node is employed and each node joins the overlay one by one in every second. Each node runs an instance of Chord application on top of a UDP agent.

The vehicular network traffic scenario is created with the NS-2 Random Trip Mobility Model Tool (PalChaudhuri et al., 2005). We captured four street maps with different characteristics from U.S. TIGER (Levergood, 1998) database. Fig. 4 illustrates the street layouts of the three locations. The first map is from the Rice University in Houston. The size of Houston street map is 1200m x 1200m, and the speed setting of vehicles is between 10 m/s and 20 m/s. A vehicle goes from one crossroad to another, and decides the next crossroad to go with zero waiting time. We believe this traffic scenario can give us a general evaluation on the performance enhancement of MChord. To evaluate the effect of map size on Chord and MChord, we captured the maps sized 1600m x 1600m and 2400m x 2400m from New York City. Several speed settings on traffic scenarios are also selected to evaluate the effect of vehicular speed. We choose three ranges of speed: 5 m/s to 15 m/s, 10 m/s to 20 m/s, and 15 m/s to 25 m/s to generate the vehicular traffic scenarios. In the last map, we choose a 1600m x 1600m area in Idaho State to simulate a vehicular network in country area. We also generate three traffic scenarios with three speed settings, which are 10 m/s to 20 m/s, 15 m/s to 25 m/s, and 20 m/s to 30 m/s. Actual queries starts at 1150 seconds. After the bootstrapping phase, a P2P query with a random key by a random node is sent every second for the P2P protocol convergence. Totally 3800 queries are sent in each trial. Finally, at the end of each simulation run, an additional 50 seconds is used to complete the last few query events. Every query has a second-chance if no answer is returned in the first time. Each scenario is simulated 10 times with different random seeds that produce different mobility traces and queries.

5.1 Mobility Model Characteristics

We first briefly analyze the characteristics of the geographical map and network mobility model. We analyzed the network scenarios under the assumption that nodes can con-
nect to other nodes within the default NS-2 802.11 transmission range 250m. We measured the average duration of each individual connection between two nodes in the simulated network schemes. The results are shown in Fig. 8. We observed that the major factor in the duration of connection is the speed of vehicles. When the speed is lower, the connection duration will be longer. Another factor is the street layout. With the same average speed setting, the connection duration in New York is much shorter than in Idaho. This is because there are limited number of roads and intersections in Idaho. When two vehicles are within communication range, it is highly possible they are still on the same road for a while. In contrast, in New York City, the one of the two vehicles might turn into a different road at one of the many intersections. So the connections can not last for a long time.

We also analyze the degree of connectivity in these schemes. The degree of connectivity is defined by the number of connections a node can establish within one hop at any time instant. The results are shown in Fig. 9. Obviously, the degree of connectivity depends mostly on the node density in the vehicular network. When the node density increases, the degree of connectivity increases linearly. From observations, the speed setting has little effect on this, so we only show one speed setting scheme in each map. A small network in 1200m x 1200m Houston has larger degree of connectivity than 2400m x 2400m New York network. We have another interesting observation that the degree of connectivity is larger in Idaho than in New York. We believe this is because the map layout in Idaho is simpler, so vehicles are more likely on the same road. In New York vehicles usually are on different roads.

Another important factor in a vehicular network is the degree of partitioning. Ideally, we would prefer one unpartitioned network. In reality, a vehicular network may be partitioned into small groups due to physical or routing limitations. Nodes in different groups cannot connect to each other. We observed that the average number of groups is still almost unaffected by the speed settings. As shown in Fig. 10, the main factor is the street layout. Networks in Idaho has much smaller average number of groups than the case in New York. As we expected, the Houston map has the lowest number of groups because of the smallest map size and the high degree of connectivity. The network in 2400m x 2400m New York has the largest number of groups. It is worthy to mention that nodes distributes unevenly among groups. In most case, one group is dominant and other groups only consist of a few nodes.

We evaluate the performance of Chord and MChord with the following metrics.

1. Application layer forwarding steps: the number of hops that a query message is processed by application layer. The minimum value of application layer forwarding steps of a responded query is two. One contributed from the response node. Another is from the query initiator. If a node commits it as the node who should answer its query, the query is responded immediately. We called such queries as immediate queries. Immediate queries are excluding from calculation. Also, if queries fail at the first time and be responded at the second time, the application steps cause by first time are excluding from calculation.

2. Query response ratio: the ratio of queries is responded. Noted, even a query is responded, the answer may be wrong since the inconsistency of the overlay.

3. Correct query response ratio: the ratio of queries is correct corresponding to the current overlay. For example, node A with ID 12345678, node B with ID 23456789, and node C with ID 34567890 are currently in the overlay. Node A loses contact from B due to consecutive ping message loses, temporal network partition, etc. As a result, A will think he is responsible for the keys ranging from 12345678 to 34567890; however, there is a node B with key 23456789. If a query with key 13456789 and A response, the query is thought to be correct since A should cover the key. While if a query with key 24567890 and A response, the query is considered to be wrong since that is B’s responsibility to answer the query. Any nodes, including A, can repair their views of the overlay. As time goes by, A is supposed to fix the anomaly. As a result, queries answered by wrong nodes are considered wrong.

4. Delay: after a query is sent, the delay measures the time passes by until a successful reception of answer. Queries with delay equal to zero, that is, immediate queries, are excluding from calculation.

5.2 Bootstrapping Performance

We first evaluate the performance of passive bootstrapping under Houston scenario. The result is illustrated in Fig. 2. A consistent state is that every node in the overlay has correct information about its successor. The average duration that the P2P overlay reaching consistent state is plotted. Every node starts its MChord instance at the beginning of a simulation. Experiments repeat 10 times. Since the size of the map among experiments is fixed, the speed of reaching consistency becomes higher as the node density grows. As Fig. 2 shows, the passive bootstrapping mechanism outperforms basic hook-node solution slightly. Passive bootstrapping not only converges faster than hook-node solution, but also prevents network congestion around the hook nodes since passive bootstrapping does not try to connect to hook nodes.

5.3 Traffic Reduction

Table 2 summarizes one of the Houston 80-node network simulations. This example is used to demonstrate the traffic reduction in MChord and MChord with cross-layer extension. MChord reduces not only application layer traffic,
but also network layer traffic because 1) MChord uses overlay table broadcasting instead of unicast ping. No route is built or maintained, so lots of signaling overhead due to route discovery is saved. 2) An overlay table broadcasting message contains more information than a unicast ping, so more packets are saved. In addition, in such network, the Chord introduces heavy traffic that lead to network congestions. A big fraction of packets are dropped due to the congestion. MChord with cross-layer enhancement saves more traffic when using a long overlay table broadcasting interval. We can calculate the number of overlay table broadcasting packet is about 114,286 in MChord and about 13,115 in MChord with cross-layer extension. Between two of them, it could save about 101,171 packets. Indeed, it saves 97,791 packets in this simulation. Also, the AODV in NS2 do not use hello message. To determine whether a link is still there or not relies on the feedback from 802.11 MAC. If a route breaks, a packet through this route cannot be acknowledged by the intended receiver. So 802.11 MAC will retransmit this packet again and again until it reaches the retry limit. When a packet reaches the retry limit of 802.11 MAC, the packet is dropped by 802.11 MAC and 802.11 MAC will send a signal to AODV to indicate a link failure of some vicinal node. Due to the vehicular mobility, a link is vulnerable to break. Most packets are dropped due to mobility, i.e., link failure. Since only unicast packets need routes, so the packet dropping due to reach retry limit are almost the same in MChord with and without cross-layer enhancement. The big fraction of Chord packet is dropped is inevitable and reasonable.

5.4 Vehicular Network P2P Performance

Fig. 5 illustrates the performance of Chord and MChord with and without cross-layer enhancement under Houston scenario. It is clear that MChord beats Chord by a considerable amount under any giving metrics. Chord cannot get a high performance in query response ratio or in correct query response ratio. MChord reflects very good query response ratio and correct query response ratio. We can conclude that the approach Chord using in wired network is not adequate in vehicular network. The theoretical average application steps are $1/2O(\log N)$. In vehicular network, maintaining a route between two nodes is cost-prohibitive because of the considerable overhead. Construct a route on-demand seems a good solution. A Chord query needs to construct $O(\log N)$ routes to complete the query correctly. Each application step needs a route construction if there is no route available. It is very rare in a vehicular network without group mobility that a connection could exist for a long time. Making every route available to finish a query correctly turns to overlay consistency, the overlay consistency is the ratio of the number of nodes which have correct successor to the number of all nodes in the network. It is clear MChord always keeps in high consistency while Chord is not. That is also evidential that correct query response ratio is low in Chord.

Fig. 6 and 7 illustrate the performance of Chord and MChord with and without cross-layer enhancement under Idaho and New York scenarios. MChord still beats Chord. As mentioned in the previous section, queries with less application steps tend to success. In these scenarios, only Chord query which travels shorter can reach the correspondent node, so the delay of Chord seems better than that of MChord. Although there are limited responses in the original Chord, most of the query results are wrong. When node density is so low that reaching others are not easy, the query response ratio and correct query response ratio are low. Because only queries travel not too far can get a response, the delay is small, too. When the node density grows, reaching others becomes easier. As a result, the query response ratio and the correct query response ratio increase. While delay also increases, this is because the connectivity is still not enough to let a stable connection proceed. As the network becomes denser and denser, the query response ratio and the correct query response ratio boost. Moreover, the delay drops in the densely connected scenario. It is very interesting that performance drops only slightly when the speed of vehicles increases. The proposed MChord can complete a query within a short time that is less than the connected time duration in high-speed cases. Most communication overhead contributes from 802.11 MAC layer RTS/CTS/ACK and delay of route request from AODV. MChord saves a lot of unnecessary application steps. As a result, MChord can complete queries and response on time even when vehicles drive at very high speed while Chord cannot.

5.5 Overlay Table Broadcasting Interval

We adjust the interval of overlay table broadcasting from [3.4] to [15.16] and [30.31] with and without MChord cross-layer enhancement. The results are shown in Fig. 11. It is clear that MChord without cross-layer enhancement performs terribly when the overlay table broadcasts less frequently. When node density becomes higher, MChord behaves better.

With a given broadcast interval parameter, nodes will hear more overlay table broadcasting during a fixed period when node density is higher. Thus, the performance is enhanced because of the more overlay information in dense networks. The situation changes when we deploy MChord cross-layer enhancement. As expected, performance degrades at shorter broadcast intervals, as information is updated less frequently. An interesting observation is that MChord performance degrades more compared with
cross-layer MChord. The cross-layer knowledge harvesting and knowledge piggybacking could effectively increase the speed of overlay information update, even under infrequent broadcasting operating conditions. The cross-layer enhancement for MChord could reduce the overlay table broadcasting overhead while maintaining good overlay protocol consistency. As a result, we could apply MChord with cross-layer enhancement to provide efficient P2P over VANET operation.

5.6 Comparative Performance in Different Mobility Schemes

We compare the results from different network mobility schemes. We first discuss the correct query response ratio. In Fig. 12 we observed that the correct query response ratio is majorly affected by the map size. Networks in 1200m x 1200m Houston map have the best correct query response ratio while networks in 2400m x 2400m New York map have the worst performance. In the same size of maps, networks in Idaho outperform networks in New York. This is because the number of partitioned groups is less in Idaho, as we discussed in Fig. 10. If the network is less partitioned, the query initiator and the query responder are more likely in the same reachable group. The P2P query can be correctly responded only when the two nodes are in the same partitioned group.

Next, we compare the average delay in different network schemes. The map size still affects the delay greatly. In Fig. 13 we can see the network in Houston has lower delay while in the larger New York map the delay is much higher. In 2400m x 2400m New York map, delay increases as the number of nodes increases. Similarly, we could also observe that in the 1600m x 1600m New York and Idaho maps, but the curve transition is at different node number parameter. Two factors: transmission distance and the degree of connectivity affect delay. When the number of nodes increases, the number of partitioned groups decreases and the size of connected groups grows. The distance and the hop counts between two nodes in the same group increases due to the larger group size. However, the degree of connectivity also increases, so it is possible to find a shorter route with fewer hops. The curve transition happens when the node density is large enough to find shorter routes easily.

Finally, we discuss the no responded queries in different mobility schemes. The no responded query ratio is the ratio of queries without any response from other nodes to by the total query events. In Fig. 14, we observed that the major factor is still the network partitioning issue. When nodes are in different groups, it is impossible to have response from the respondent node. We observed that the curve of no response ratio is very similar to the curve of partitioned group number in Fig. 10. An interesting observation is that the no response ratio in the 2400m x 2400m New York is not as high as we expected from the partitioned group curve. This is because the number of groups in the network is too large that the consistency cannot be hold in Mobile Chord. The inconsistency results in a significantly amount of false information. Although the nodes may respond to the query, the responses are usually false.

6 Conclusion

Cross-layer design to improve P2P overlay performance over vehicular ad hoc network is investigated. The baseline Chord protocol performance and the enhanced Mobile Chord performance are compared with extensive NS-2 simulation. The proposed Mobile Chord scheme performs well in the challenging VANET environment while Chord does not. The P2P overlay consistency of Mobile Chord is significant higher than that of the baseline Chord. The proposed bootstrapping technique also reduces control signaling cost and improves P2P network convergence time in highly dynamic vehicular environment. Cross-layer design for Mobile Chord could effectively reduce the signaling cost by efficiently disseminating P2P overlay information. In summary, Mobile Chord outperforms Chord in terms of overlay consistency, the number of application layer forwarding steps, query response ratio, correct query response ratio, and the average query delay.

Acknowledgement

This work was supported by the National Science Council of Taiwan under Grant NSC-97-2221-E-002-138-MY2.

REFERENCES


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Chord</th>
<th>MChord</th>
<th>Cross-layer MChord*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Successor Reachability Check Interval</td>
<td>[4, 6] sec</td>
<td>[9, 11] sec</td>
<td></td>
</tr>
<tr>
<td>Overlay Table Broadcasting Period (MChord)</td>
<td>[3, 4] sec</td>
<td>[9, 11] sec</td>
<td></td>
</tr>
<tr>
<td>Overlay Table Entry Expiration Time (MChord)</td>
<td>60 sec</td>
<td>[9, 11] sec</td>
<td></td>
</tr>
<tr>
<td>Finger Entry Reachability Check Interval (Chord)</td>
<td>[5, 7] sec</td>
<td>[9, 11] sec</td>
<td></td>
</tr>
<tr>
<td>Maximum Retry of Finger Entry (Chord)</td>
<td>3 ping loses</td>
<td>[9, 11] sec</td>
<td></td>
</tr>
<tr>
<td>Random Finger Entry Update (Chord)</td>
<td>[15, 18] sec</td>
<td>[9, 11] sec</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: The important parameters of Chord and MChord

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Chord</th>
<th>MChord</th>
<th>Cross-layer MChord*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of AODV RREQ</td>
<td>16326</td>
<td>13250</td>
<td>17912</td>
</tr>
<tr>
<td>Number of AODV RREQ packets</td>
<td>650729</td>
<td>503191</td>
<td>582379</td>
</tr>
<tr>
<td>Chord introduced packets at application layer</td>
<td>674809</td>
<td>215168</td>
<td>117377</td>
</tr>
<tr>
<td>Chord packet drops</td>
<td>531621</td>
<td>84857</td>
<td>76555</td>
</tr>
<tr>
<td>Number of Chord packets dropped by reaching retry limit at 802.11 MAC</td>
<td>418784</td>
<td>72029</td>
<td>71268</td>
</tr>
</tbody>
</table>

Table 2: The traffic analysis of a 80-node network (*: overlay table broadcast interval=[30, 31])

Figure 1: An example of Chord overlay over VANET
Figure 2: The seconds MChord needs to reach consistent state in a vehicular network

Figure 3: Cross-layer design

Figure 4: Maps from Houston, Idaho, and New York
Figure 5: Vehicular network in Houston (a) Query response ratio and correct query response ratio (b) Application layer forwarding steps (c) Query delay (d) Overlay consistency in an 80-node network
Figure 6: Vehicular network in Idaho (a) Query response ratio and correct query response ratio (b) Application layer forwarding steps (c) Query delay (d) Correct query response ratio in different speed
Figure 7: Vehicular network in New York (a) Query response ratio and Correct query response ratio (b) Application layer forwarding steps (c) Query delay (d) Correct query response ratio in different speed

Figure 8: Average connection duration
Figure 9: Degree of connectivity

Figure 10: Number of partitioned network groups

Figure 11: Interval of overlay table broadcasting: 3s, 15s, and 30s
Figure 12: Correct query response ratio

Figure 13: Average delay

Figure 14: No response ratio