Peer to Peer Mechanism in ICMAN's using Spray and Wait Routing Algorithm

(Spray and Wait:An Effiecient and Epidemic Routing Mechanism)

P. R. S. Naidu
Assistant Professor, Computer Science and Engineering, JNTUK-UCEV, Vizianagaram, India.

Ashok S
Assistant Professor, Computer Science and Engineering, JNTUK-UCEV, Vizianagaram, India.

Abstract – Mobile ad-hoc networks (MANETs) and peer-to-peer (P2P) systems are ubiquitously emerging technologies which share a common underlying decentralized networking paradigm. Deploying Peer to Peer architecture over Mobile ad hoc networks results in an efficient content distribution network. However most of the time MANETs are intermittently connected (No complete path exists from source to destination) due to scattered node densities, limited radio transmission range and power limitations. Various researches have proposed different routing/content delivery schemes under intermittent connectivity. In P2P content distribution systems searching play a major role in identifying the interested content. Because only after identifying interested content downloading or sharing begins. Existing content searching methods over intermittently connected MANETs (ICMANs) like epidemic P2P content search suffers from message delivery delays and resource wastage. This thesis focuses on developing an efficient content search scheme; as a result Spray and Wait routing is implemented for content searching in intermittently connected MANETs.

Index Terms – Spray, Wait, ICMAN's, Peer to Peer.

1. INTRODUCTION

Content sharing applications have grown in popularity and usage in the internet for the last few years. Most of them are based on Peer-to-Peer (P2P) architecture that is characterized by direct access between peer computers rather than through a centralized server. Although originally developed for the wired Internet, theseP2P based content distribution networks have greater scope in wireless networks.

The fast development in wireless technology allows people to use devices such as smart phones, tablets, laptops and different mobile form factors with great ease and at the same time allows using of different services like e-mail, live media streaming, sharing content, web browsing, etc.

Moreover, instead of the conventional cellular networks, low cost wireless connectivity such as Bluetooth and IEEE 802.11 offer the mobile devices an alternative way to communicate with each other. By exploiting such low cost wireless connectivity, Mobile Ad Hoc Networks (MANETs), the automatically self-organized wireless networks without any pre-configured infrastructure, can be established to enable independent mobile users to interact with each other. Most of the time MANETs are intermittently connected causing the network to be intermittently connected MANETs (ICMANs) where there exists no end-to-end path from source to destination. Recently there has been a growth in research activities on ICMANs because of its real-time implementations in military, inter planet satellite communications etc.

P2P and MANETs have received a lot of attention for research because they share many similarities such as decentralized architectures, self-configurable and dynamic topology. P2P and MANETs are further explained in the following sub-sections.

Mobile Ad hoc Networks:

In general, mobile ad hoc networks (MANETs) are networks formed dynamically by an autonomous set of mobile nodes that are connected via wireless links without relying on any pre-configured infrastructure or centralized administration. In other words, these participating wireless mobile nodes can freely and dynamically self-organize into arbitrary and temporary network topologies, allowing themselves to seamlessly communicate with each other in areas with no pre-existing communication infrastructure. In MANETs, each node communicates directly with any other node within its transmission range, while communication beyond this range is established by employing intermediate nodes to set up a path in a hop-by-hop manner.

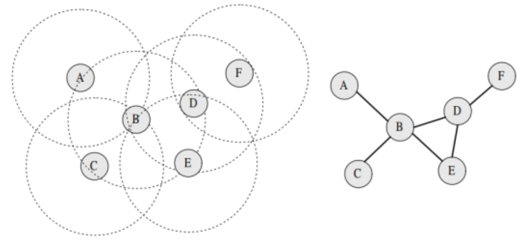


Fig. 1.1 Mobile Ad hoc Networks

Figure.1.1 shows an example of mobile ad hoc network and its communication topology. Note that due to the self-organized nature of MANETs, if these nodes are free to move randomly, they may organize themselves arbitrarily and there by the network topology can change rapidly and unpredictably. Therefore, Fig. 1.1only represents a snapshot of network topology at a certain time instant.

In general, the characteristics of MANETs can be summarized as follows:

i. *Wireless*

ii. Mobility

iii. Ad hoc based connection

iv. Infrastructure-less architecture

v. Multi-hop routing

MANETs remove the constraints of infrastructure and allow devices to establish and join the network anywhere at any time. However, as mobile nodes are moving arbitrarily, the network topology may change constantly, resulting in route changes, frequent network partitions and even packet losses. Moreover, MANETs are normally formed by mobile devices with limited battery power, and the equipped wireless interfaces may also suffer limited bandwidth and high error rate. Hence, in order to accommodate such dynamic topology of MANETs as well as resource constrained devices, an abundance of MANET routing protocols have been proposed in the literature. Based on the timing that the routes are established, these protocols can be classified into the following main categories:

Proactive routing protocols: Each node propagates route updates proactively and periodically over the network to allow any other node to maintain a consistent and up-to-date routing table. Representative proactive protocols include optimized link state routing (OLSR) protocol and destination-sequenced distance-vector (DSDV) protocol.

Reactive on−demand routing protocols: A route to a destination, on the other hand, is established only when there is a demand from the source node. When needed, the source node triggers a path discovery process over the network to set up the route to the destination. Once the route has been set up properly, it is maintained either until it becomes no longer used or has expired, or until the destination becomes inaccessible from the source. Both dynamic source routing (DSR) and ad hoc on-demand distance vector (AODV) routing are referred to as representative examples of reactive on-demand routing protocols.

Hybrid routing protocols: The characteristics of proactive routing protocols and reactive on-demand routing protocols are combined to form hybrid routing protocols. Zone routing protocol (ZRP) is a typical example representing hybrid routing protocols.

All the previously mentioned routing protocols implicitly assume that MANET is connected and there exists a complete end-to-end path between any pair of node swishing to communicate with each other.

This assumption restricts those routing protocols to networks containing enough nodes to build a fully connected topology. Unfortunately, there exist some situations where intermittent connectivity may arise from node mobility, short transmission range, sporadic node density, power limitations, and so on, and thereby most of the time a complete end-to-end path between any two nodes does not exist, or such a path is highly unstable while being discovered. Fig. 1.2 is a typical example representing such situation. As can be seen, if node D still wants to deliver data to node E over time, the existing routing protocols for conventional MANETs cannot work or may suffer serious performance. This is because there is not any end-to-end path over time, which can be discovered by such routing protocols to facilitate the corresponding data communication.

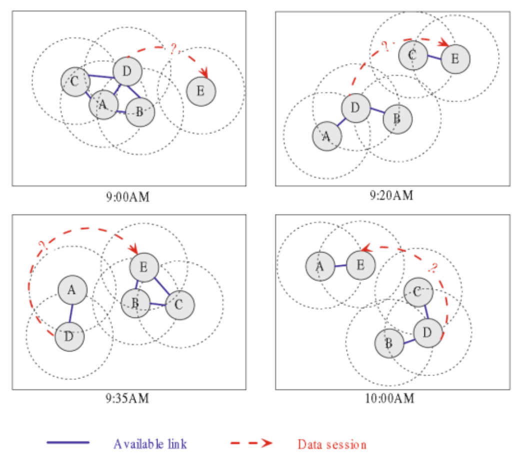


Fig. 1.2 An example of data communication in an intermittently connected mobile ad hoc network based on existing routing protocols

A military mobile ad hoc network may become intermittently connected when mobile nodes (e.g., soldiers, tanks) move out of each other's transmission range and are subject to being destroyed. Moreover, pocket switched networks are another similar examples, since they are formed by human carried mobile devices based on their Bluetooth or IEEE 802.11 interfaces with short transmission range. In addition, similar intermittent disconnectivity can be encountered by vehicular ad hoc networks (VANETs) due to high vehicle velocities (compared to the transmission range). In the literature, mobile ad hoc networks that suffer such intermittent connectivity (in the absence of end-to-end routing path) are commonly referred to as intermittently connected mobile ad hoc networks (ICMANs).

Real-time communications (e.g., voice-over-Internet Protocol (VoIP) and video streaming), which require a fully-connected path to forward sequenced packets timely, cannot work over ICMANs. However, it does not mean other applications can never be deployed over ICMANs. If delay can be tolerated by the applications, to improve the routing performance, i.e., to reduce packet loss ratio, mobility and storage spaces of intermediate nodes can be exploited to forward data to their final destinations. In other words, data can be temporarily stored in intermediate nodes until the node mobility generates the next possible forwarding opportunity. This is largely because connection topologies (constituting of mobile nodes) may overlap at different periods of time, thereby facilitating delayed data delivery to the destination over store-carry-forward strategy. Moreover, since any possible node can opportunistically be employed as the next carrier to bring data closer to its eventual destination, ICMANs are also referred to as opportunistic networks. In addition, since incurred large delays primarily limit ICMANs to delay-tolerant applications, ICMANs belong to the family of delay tolerant networks (DTNs).

Peer-to-Peer (P2P):

A peer-to-peer network is simply a number of peer machines communicating and sharing resources with each other. Traditional client-server networks have the concept of a central server connected to several client machines. Communication between connected clients takes place through server.

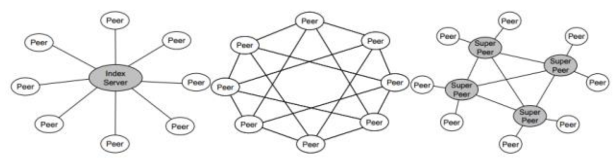
By forming an overlay network on top of the Internet, P2P content distribution allows peers to publish, search and download contents from each other. In P2P, the participating peers in the network communicate directly with each other unlike the traditional client-server network or multi-tiered server network. The peers share computer resources and services by directly exchanging information among them. Exchange of computer resources and services means the exchange of information, processing cycles, and disk storage for files. All of these are done generally without depending on any centralized servers or resources.

Thus, the P2P architecture enables true distributed computing. It creates networks of computing resources which exhibit fault tolerance and very high availability. There are many interesting and useful applications that can be accomplished by using P2P computing besides the popular applications as Napster, Kazaa, and Gnutella etc. To be successful, peer-to-peer computing requires availability of numerous interconnected peers and bandwidth. Existing desktop computing power and networking connectivity are encouraging for P2P networks.

By considering how to organize the participating peers to build overlay network and how to place the contents, these P2P content delivery networks are classified as unstructured and structured P2P networks.

Unstructured P2P:

In the unstructured P2P content delivery networks, the placement of available contents is completely unrelated to the overlay topology. However, by considering how to provide efficient content search, the unstructured P2P systems can be further categorized as fully decentralized, centralized and partially centralized.



(a) Centralized (b) Fully decentralized (c) Partially centralized

Fig.1.3 Unstructured P2P overlays

As shown in the Figure 1.3 (a) in centralized and unstructured P2P systems the contents are indexed locally, a typical centralized P2P content delivery networks employs a centralized server to manage all users and to index all available contents stored by them. Whenever a new peer wishes to join the network has to register itself and report all contents it has to the server, while the server can subsequently index all these contents along with meta-data descriptions for other peers to query. On receiving a content query from a user, the index server can return a list of matching contents with their exact locations. As a result, the user can directly contact with the peers that hold the matching contents to activate download process. Napster is an example of centralized P2P content distribution network.

Figure 1.3 (b) shows fully decentralized and unstructured P2P networks where all peers perform exactly the same functionalities and there is no central coordination of their activities. Moreover, all peers in this network are connected non-deterministically and all available contents are just stored and indexed locally to handle highly-transient node populations. Accordingly, the simplest content search scheme is to flood content queries across the network in a breadth-first or depth-first manner to locate all desired contents. To limit the spread of queries through the network, a time-to-live (TTL) field is employed by content queries. However, even TTL is employed; the flood-based content search mechanisms are not scalable for the networks with large scale.

The aim of partially centralized P2P architecture shown in Figure 1.3 (c) is to exploit and to take advantage of the inherent heterogeneity of P2P systems to present a cross between fully decentralized and centralized P2P systems. In the partially centralized P2P systems (also referred to as the hybrid P2P systems or super-peer P2P systems), the concept of super-peer is utilized to introduce hierarchy into the network. Different from a fully decentralized system, where all of the peers will be mostly equally loaded, regardless their capabilities of CPU power, storage space and even bandwidth, a partially centralized P2P system employs some super-nodes with higher capabilities to act as locally centralized index servers to their surrounding peers and proxy content queries on behalf of these peers. Even though most of the peers are only connected to their super-peers, these super-peers are connected to each other as peers in a fully decentralized system.

Structured P2P:

To address the scalability issues occurring in the unstructured networks, structured networks try to create network topology and place the contents based on some specific rules so that content queries can be efficiently forwarded to the node with desired contents. Most of such fully decentralized but structured networks employ the distributed hash table (DHT) as the underlying technology for topology construction and content placement. Unlike unstructured P2P networks with their random topology, DHTs impose a structure on the overlay topology by no longer choosing routing table entries arbitrarily. Instead, routing table entries have to satisfy certain criteria depending on the respective DHTs. At the core of each DHT lies the ability to route a packet based on a key, towards the node in the network that is currently responsible for the packet's key. This process is referred to as indirect or key-based routing.

2. LITERATURE SURVEY

A Survey on P2P File Sharing Algorithms over Mobile Ad hoc Networks:

Li Liu, Yanfang Jing, Yue Zhang, Bingbing Xi [1] analyzed the recent researches about file sharing algorithms over MANETs. According to different searching methods file sharing algorithms over MANETs categorized into four categories: the DHT-based methods, flooding-based methods, advertisement-based methods and social-based methods.

DHT-Based Method: Most of approaches such as Pastry, Tapestry, and Chordin structured P2P network are mainly based on Distributed Hash Tables (DHT). DHT-based method use DHT to map objects with corresponding nodes in a distributed fashion using hash functions. DHT-based searching methods rely on DHT to record the content information of the neighbors. When a node wants to search some content, it firstly resorts to the DHT to achieve the relative information. From this point, DHT-based searching approach is very effective. However, the establishment and maintain of the DHT are not easy due to the dynamic topology. A lot of messages are needed to keep the DHTs renew and consistence, which leads to heavy traffic in the networks.

Flooding-Based Method: Flooding-based searching methods are based on broadcasting to implement the search process. When a node wants to search some content, it firstly broadcasts the requirement to its neighbors. Then from the response messages, it can obtain the information which relate to the content owner. The most default in flooding-based methods is the high overhead because of the high amount of duplicated messages. These overhead induces the high congestion due to a high volume of traffic, which is a significant problem in MANETs. In addition, local broadcasting used in some methods cannot guarantee file searching success.

Advertisement-Based Methods: In advertisement process, each file holder regularly broadcasts an advertisement message in order to inform surrounding nodes about what files are to be shared. Advertisement-based methods also lead to high overhead, and they have low search efficiency because of expired routes caused by transient network connections. Flooding-based methods and advertisement-based methods are fit for the relative stable MANETs. However, in more disconnected MANETs, the two kinds of method are failed due to the large overhead.

Routing in Intermittently Connected MANETs and Delay Tolerant Networks: Overview and Challenges:

Zhen shengZhang [2] provided an overview of different routing protocols in ICMANs. In ICMANs at any given time, there exists no path between source and destination. In this case the only way of routing is as follows, the source forwards messages to intermediate nodes then these nodes hold the message and carry them along the network until the destination is reached. This form of routing is called opportunistic routing. Store Forward Carry (SCF) is one of the opportunistic routing methods where data is temporarily stored in the intermediate nodes until the node mobility generates the next possible forwarding opportunity.

If all the future topology of the network (as a time-evolving graph) is deterministic and known, or at least predictable, the transmission (when and where to forward packets) can be scheduled ahead of time so that some optimal objective can be achieved. If the time-evolving topology is stochastic, SCF routing performs routing by moving the message closer to the destination one hop at a time. If the nodes know nothing about the network states, then all the nodes can do is to randomly forward packets to their neighbors. Protocols in this category are referred to as epidemic. If one can estimate the forwarding probability of its neighbors, a better decision could be made. Protocols in this category are referred to as history based or estimation-based forwarding. Furthermore, if the mobility patterns can be used in the forwarding probability estimation, an even better decision may be made. Protocols in this category are referred to as model-based forwarding. In the epidemic routing category, packets received at intermediate nodes are forwarded to all or part of the nodes' neighbors (except the one who sends the packet) without using any predication of the link or path forwarding probability. Epidemic routing is a natural approach when no information can be determined about the movement patterns of nodes in the system.

Advances in Peer-To-Peer Content Search:

DeepaKundur, Zhu Liu,Madjid Merabti, and Heather Yu [3] address recent advances in P2P content search by providing an overview of influential research in the area. This gives an overview of text based P2P file search and content-based search mechanisms. Napster [4] is P2P implementations that brought P2P computing into the mainstream and which sparked a large amount of media attention. Napster was created purely for the distribution of MP3 audio files, and as such it was swamped with negative press because people were downloading digital content illegally and consequently ignoring content copyright. Each Napster node downloads and installs the client software used to connect the peer to the centralized Napster server. Once connected, peers share MP3 files stored locally on their hard drives, with text-based information about them being indexed and stored by the Napster server. Clients submit text-based queries to the Napster servers for a particular audio file. This results in a list of files that match, along with the connection information, username, IP and port address the querying client must use to connect to the peer hosting the file. Once the querying peer has this information it attempts to connect to the peer and transfer the target content in a P2P fashion. At this point the Napster server is no longer required.

Another hybrid protocol, similar to Napster called iMesh uses a centralized server, to which clients connect to in order to search for content. The iMesh model differs somewhat to Napster in two respects. First, it allows any content to be shared including MP3 audio files. Second - the reason why iMesh has not been subjected to the same legal problems as Napster - it has a mechanism to remove copyrighted files from the network. Computational expense and scalability issues associated with the above mentioned models are well documented, which has resulted in new P2P networks devoid of any centralization. The most popular being the Gnutella protocol [5]. Like iMesh it provides a generic file sharing mechanism that allows any digital media content to be shared. However it differs from iMesh and Napster because the Gnutella protocol uses a purely decentralized model, which is not reliant on any centralized authority.

The search mechanism used by Gnutella adopts a different approach to Napster in that it does not require any centralized server to manage the location of content within the network. Search packets containing text queries are used with predefined TTL values, the default value being 7, which corresponds to the number of hops the message can take. The packet is passed to all the immediate peers' the querying peer is connected to, which in turn is passed to all the peers the peer is connected to. If a node is found with a file name matching the query, the information is routed back to the querying peer. The file can then be downloaded directly from the target node. This is commonly referred to as blind search. Most existing P2P systems provide very limited content search capabilities, for example, search based on document title, author names, keywords, or descriptive text. To retrieve the relevant content more effectively, requires an approach that provides richer searching features. Content-based search is essential for querying textual documents, and it is also desirable for querying multimedia data when text annotations are nonexistent or incomplete.

Multimedia content indexing and retrieval has been an active field for more than a decade. It draws tremendous research effort from the academia, the industrial, and the standard organizations. For example, MPEG-7 is a standard sponsored by the International Organization for Standardization for describing the multimedia content. It provides support to a broad range of applications, and it will make the web as searchable for multimedia content as it is searchable for text. The evolution of the World Wide Web, including the introduction of Rich Site Syndication (RSS), Web 2.0, and the semantic web, enables the web information be machine process able (rather than being only human oriented), thus permits browsers or other agents to find, share and combine information more easily.

An Epidemic P2P Content Search Mechanism for Intermittently Connected Mobile Ad hoc Networks:

Yaozhou Ma and Abbas Jamalipour[6] propose an epidemic P2P content search scheme. This search mechanism uses epidemic routing to forward search requests. Epidemic routing is a plain flooding based mechanism. The basic idea behind epidemic routing is to utilize every possible forwarding opportunity to deliver data to its eventual destination. For example, as shown in Fig. 2.4, a source, S, wishes to transmit a message to a destination D but no connected path is availableat9:00AM. However, it can forward a copy of message to its encounter, C, and ask C to deliver the message copy in the future. Moreover, when S encounters B at 9:20 AM, it also asks B for help by forwarding another message copy to B. At the same time, C is doing the same thing i.e., asking E to delivery another message copy in the future. Eventually, the message is delivered to its destination when B meets Dat9:35AM, despite S and D cannot build a complete path between them during this period. As can be seen, epidemic routing relies on mobile nodes coming into contact through nodes mobility, since only when two mobile nodes are within their transmission range, they can exchange messages that the other node doesn't store.

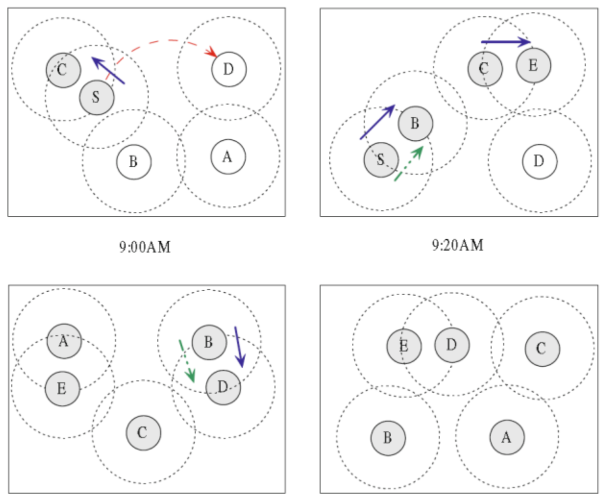


Fig. 2.1.Illustration of epidemic routing in an Intermittently Connected Mobile Ad hoc Network.

Epidemic routing extends the concept of flooding in intermittently connected mobile networks. It is one of the first schemes proposed to enable message delivery in such networks. Each node maintains a list of all messages it carries, whose delivery is pending. Whenever it encounters another node, the two nodes exchange all messages that they don't have in common. This way, all messages are eventually "spread" to all nodes, including their destination (in an "epidemic" manner). Although epidemic routing finds the same path as the optimal scheme under no contention [7], it is very wasteful of network resources. Furthermore, it creates a lot of contention for the limited buffer space and network capacity of typical wireless networks, resulting in many message drops and retransmissions. This can have a detrimental effect on performance.

This is the existing system and it explains the fully decentralized P2P architecture in ICMANs context with the framework of epidemic content search, followed by a utility based buffer management policy. The objective of such policy is to utilize the constrained buffer spaces more efficiently so that during a certain period, the network can serve more content search requests while more matching contents can be discovered for each request. Note that assuming each peer has an unlimited buffer space for its own requests while a restricted cache space is reserved to temporarily store such requests from any other peers in the network i.e., in epidemic P2P content search scheme, buffer/cache management policy is only carried out for the cached requests from other peers.

P2P epidemic content search:

Content search service here allows each peer look for its interested contents over the network based on some relevant keywords. To this effect, every content is assumed to contain a unique content identifier and related metadata with corresponding keywords, while each peer indexes its stored contents based on the keywords. Whenever a peer wants to download some interested contents without any information about the corresponding content identifiers, it can flood a content search request based on related keywords across the network. On receiving such a request, each intermediate peer performs a keyword-based search through its stored indices, while if any matching content exists, a response message is accordingly generated and forwarded back. Such response message includes all identifiers as well as all metadata related to the matching contents. On receiving the response message, the querying peer can finally determine which content it really wants based on the metadata and accordingly download it according to the corresponding content identifier.

Mobile Ad hoc P2P File Sharing:

Ahmet Duran, Chien-Chung She [8] propose two efficient search schemes that use query messages filtering/gossiping and adaptive hop-limited search, respectively, for peer-to-peer (P2P) file sharing over mobile ad hoc networks. In query messages filtering approach, a node floods a query message with requested file identifiers. The query message is distributed via link-layer flooding. The query message triggers the construction of the reverse paths to the querying node in the Response Routing Tables of all intermediate nodes, while it propagates through the network. Any node receiving a query message first checks its local files before broadcasting. When the degree of replication is high and the number of requested files in the query is low, most probably the node has most of the files. When only some of the files exist in its Local Files Repository, it will update the query message, and broadcast the 'reduced' query. And the second scheme conducts adaptive hop-limited search with local broadcast transmission. In particular, the first scheme benefits applications that require file update.

Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks:

ThrasyvoulosSpyropoulos, KonstantinosPsounis, Cauligi S. Raghavendra [9] introduce a new routing scheme, called Spray and Wait that "sprays" a number of copies into the network, and then "waits" till one of these nodes meets the destination. Despite a large number of existing proposals, there is no routing scheme that both achieves low delivery delays and is energy-efficient (i.e. performs a small number of transmissions). Spray and Wait is the first of its kind protocol that result in low end-to-end delivery delay and less network congestion.

Spray and Wait bounds the total number of copies and transmissions per message without compromising performance. The advantages of spray and wait protocol are: (i) under low load, Spray and Wait results in much fewer transmissions and comparable or smaller delays than flooding-based schemes, (ii) under high load, it yields significantly better delays and fewer transmissions than flooding-based schemes, (iii) it is highly scalable, exhibiting good and predictable performance for a large range of network sizes, node densities and connectivity levels; what is more, as the size of the network and the number of nodes increase, the number of transmissions per node that Spray and Wait requires in order to achieve the same performance decreases, and (iv) it can be easily tuned online to achieve given QoS requirements, even in unknown networks.

Desirable design goals for a routing protocol in intermittently connected mobile networks are:

i. Perform significantly fewer transmissions than epidemic and other flooding-based routing schemes, under all conditions.

ii. Generate low contention, especially under high traffic loads.

iii. Achieve a delivery delay that is better than existing single and multi-copy schemes, and close to the optimal.

iv. Highly scalable, that is, maintain the above performance behaviour despite changes in network size or node density.

Flooding based algorithms like epidemic routing [10] suffers with large delays and network contention. Single-copy schemes have also been extensively explored in [11]. Such schemes generate and route only one copy per message (in contrast to flooding schemes that essentially send a copy to every node), in order to significantly reduce the number of transmissions. Although they might be useful in some situations, single-copy schemes do not present desirable solutions for applications that require high probabilities of delivery and low delays.

Spray and Wait, manages to significantly reduce the transmission overhead of flooding-based schemes and have better performance with respect to delivery delay in most scenarios, which is particularly pronounced when contention for the wireless channel is high. Further, it does not require the use of any network information, not even that of past encounters.

Definition

Spray and Wait routing consists of the following two phases:

i. **Spray phase:** for every message originating at a source node, L message copies are initially spread – forwarded by the source and possibly other nodes receiving a copy –to L distinct “relays”.

ii. **Wait phase:** if the destination is not found in the spraying phase, each of the L nodes carrying a message copy performs direct transmission (i.e. will forward the message only to its destination).

Spray and Wait combines the speed of epidemic routing with the simplicity and thriftiness of direct transmission. It initially “jump-starts” spreading message copies in a manner similar to epidemic routing. When enough copies have been spread to guarantee that at least one of them will find the destination quickly (with high probability), it stops and lets each node carrying a copy perform direct transmission. In other words, Spray and Wait could be viewed as a trade-off between single and multi-copy schemes. Its performance is better with respect to both number of transmissions and delay than all other practical single and multi-copy schemes, in most scenarios considered.

3. EXISTING SYSTEM

Existing system proposed an epidemic P2P content search scheme for intermittently connected mobile ad hoc networks. This search mechanism uses epidemic routing to forward search requests. Epidemic routing is a plain flooding based mechanism. The basic idea behind epidemic routing is to utilize every possible forwarding opportunity to deliver data to its eventual destination.

Epidemic routing extends the concept of flooding in intermittently connected mobile networks. It is one of the first schemes proposed to enable message delivery in such networks. Each node maintains a list of all messages it carries, whose delivery is pending. Whenever it encounters another node, the two nodes exchange all messages that they don’t have in common. This way, all messages are eventually “spread” to all nodes, including their destination (in an “epidemic” manner). Although epidemic routing finds the same path as the optimal scheme under no contention, it is very wasteful of network resources. Furthermore, it creates a lot of contention for the limited buffer space and network capacity of typical wireless networks, resulting in many message drops and retransmissions. This can have a detrimental effect on performance.

This is the existing system and it explains the fully decentralized P2P architecture in ICMANs context with the framework of epidemic content search, followed by a utility based buffer management policy.

Existing also used a buffer management policy to overcome the restricted storage problem in Mobile devices. The objective of such policy is to utilize the constrained buffer spaces more efficiently so that during a certain period, the network can serve more content search requests while more matching contents can be discovered for each request. Note that assuming each peer has an unlimited buffer space for its own requests while a restricted cache space is reserved to temporarily store such requests from any other peers in the network i.e., in epidemic P2P content search scheme, buffer or cache management policy is only carried out for the cached requests from other peers.

As the network scales this system suffers with huge network delays and network contention due to flooding nature of epidemic content search.

Spray and Wait routing:

Step 1: for i=0 ; i<n ; i++

for j=0 ; j < m ; j++

Assign node_id, content_id

indexcontent_id with keywords

Step 2: Find L value for generating search requests

Step 3: Generate L no. of search requests

Step 4: Spraying Phase: Requesting node keeps one of the L copies to itself and forwards remaining L-1 copies to its neighbor nodes

Step 5: On receiving a query message, the node executes the keyword-based search through its locale index server.

Step 6: If any matching contents are discovered

returnnode_id, request_id, content_id of content holding peer

Step 7: Waiting Phase: For more search hits, all nodes holding search request wait for direct communication with the content holding peer.

Algorithm Description:

Every node in the network is configured with a node_id, every content with a content_id and every content is indexed with corresponding keywords. Requesting nodes chooses L value and generates L search request copies. Every node keeps one of the L copies to itself and forwards remaining L-1 copies to its neighbor nodes. This step is called spraying phase. On receiving a query message, the node executes the keyword-based search through its locale index server. If any matching contents are discovered, a response message is generated and delivered back to the querying peer based on the peer ID as well as the request ID.As in P2P networks there can be more than one destination (i.e. more than one peer can contain the requested data), so, the L nodes holding the query message will move around the network to find content holding peer.

4. PROPOSED SYSTEM

In the proposed mechanism, apart from the content itself, every content is assumed to contain a unique content identifier and a meta-data file with corresponding keywords. Due to intermittent connectivity, to avoid high maintenance overhead for structured, centralized and partially centralized content search, the fully decentralized architecture is employed at here. In other words, in ICMANs, every mobile node indexes all of its own stored contents (published and downloaded contents) based on their corresponding keywords. As a result, whenever a node wants to download some interested contents without any information about the corresponding content identifiers, it can facilitate a content query based on the related keywords across the network. On receiving such a content query message, each intermediate node performs a keyword-based search through its stored indices, while if any matching content exists, a response message is accordingly generated and forwarded back. The response message includes the content identifiers and the meta-data files of the matching contents. On receiving this information, the node can eventually determine which contents it would like to download and accordingly requests them over the network based on the received content identifiers.

Proposed work involves implementing an effective routing technique called spray and wait protocol to carry P2P content search requests in the network. To perform spray and wait content search, every issued search request includes peerID, request ID, interested keywords, and timestamp and expiry time. Node ID and request ID together represent each unique content query message, while expiry time is given to address content searching scalability in the ICMANs context. Note that for a given query message, if the expiry time is passed, it should be removed by all possible intermediate nodes from the storage.

In order to avoid high delays and network contention as in epidemic P2P content search, this mechanism uses only fixed no. of search request copies to find interested content. Throughout this thesis this fixed no. of search request copies are labeled as “L”. The value of L determines the network packet density.

Due to the intermittent connectivity, content query forwarding and content search only happen whenever two mobile nodes encounter each other (i.e. when two nodes are within their communication range). After connection is established between two encountering nodes, the entire procedure can be summarized as follows:

1. Firstly, each node checks its buffer space and removes all expired content query messages.

2. Then each node generates L copies of its search/query message.

3. Every node keeps one of the L copies to itself and forwards remaining L-1 copies to its neighbor nodes. This step is called spraying phase.

4. On receiving a query message, the node executes the keyword-based search through its locale index server.

5. If any matching contents are discovered, a response message is generated and delivered back to the querying peer based on the peer ID as well as the request ID.

6. As in P2P networks there can be more than one destination (i.e. more than one peer can contain the requested data), so, the L nodes holding the query message will move around the network to find content holding peer.

5. SYSTEM REQUIREMENTS

The requirements for a system are the descriptions of the services provided by the system and its operational constraints. Requirements reflect the needs of customers for a system that helps solve some problem such as controlling a device, placing an order or finding information.

Functional requirements:

These are statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

1. The system shall find out content holding peers and return their identity to requesting peer.

2. Every node, content, request shall be allocated a unique identifier (peer ID, Content ID, request ID)

3. The proposed system shall allow peers to search for interested content in the network.

Input Requirements:

1. Input to the system is the number of nodes, pause time and requesting peer ID.

2. If number of nodes is not specified then system cannot generate peers.

Output Requirements:

The output of the system generates two files, namely NAM file and TRACE file.

1. NAM file graphically shows the peers behaviour in network.

2. The TRACE file contains the nodes movement among the nodes in a network.

Storage Requirements:

1. Each and every individual node in the network will store the information by using arrays.

Non-functional requirements:

System requirements set out the system's functions, services and operational constraints in detail.

1. Contents stored in the node must be indexed with keywords and Content ID.

2. Content searching should be done even though location information of nodes are not available.

Hardware requirements:

Processor Type : Pentium –IV and later versions

Speed : 2.4 GHZ or more

Ram : 1 GB RAM or more

Hard disk : 40 GB HD or more

Software requirements

Operating System : Any Linux distribution with kernel version above 8.

Development Tools : ns-2.34-(network simulator).

Languages : TCL (Tool Command Language) & C++.

6. SYSTEM DESIGN AND IMPLEMENTATION

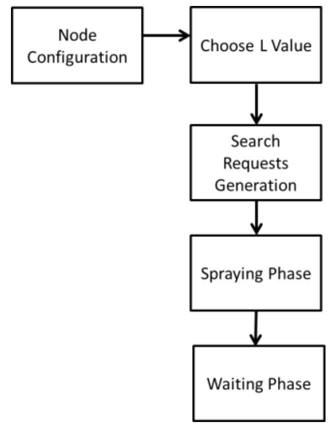


Fig 6.1 System Architecture

System Architecture

The system architecture contains components as mobile nodes which also act as routers to transmit messages between other nodes. Every component contains some content, which is indexed with appropriate keywords.

MODULE DESCRIPTION

Node Configuration

Nodes in the network are configured with node_id and contents in them with content_id. Every search request originated will be assigned a request_id. Node_id along with request_id uniquely identifies are search request.

Every search request contains requesting peer node_id, request_id, interested content keywords and time stamp.

Node_id	Request_id	Keywords	Timestamp

Fig 6.2 Search request

Choosing L value

This sub-section specifies a method for choosing L (i.e. the number of copies generated by the requesting node) in order for Spray and Wait to find content holding peer with less delay than other flooding mechanisms.

Choosing L value plays a crucial role because, if L is too low it will result in larger end-to-end delays and As L grows larger, the more the network is congested and the sophistication of the spraying heuristic has an increasing impact on the delivery delay of the spray and wait scheme. So L value must be chosen not too low nor too high.

M/2 ≤ Lmin<M………………………………… (1)

This is the basic method to choose L value. With all the above in mind this method is proposed. It state that L value should be in between half the number of nodes and number of nodes. The minimum number of copies L min needed for Spray and Wait to achieve an expected delay is independent of the size of the network N and transmission range K, and only depends on the number of nodes M.

Search Request Generation

After choosing L value, L no. of duplicate search requests are generated

Spraying Phase

Source or requesting peer forwards the L generated search copies to L encountering peers, if there is a search hit in this phase the content holding node_id, content_id are returned back to the requesting node.

Waiting Phase

Because in Content distribution networks there can be more than one destination (i.e. more than one peer contains interested content), waiting phase begins to search for interested content. In this phase all the request holding peers, waits until they directly communicate with the content holding peer.

7. TEST CASES

Test Cases for Unit Testing:

Function Name: Spraying

Unit testing for spraying function tests the pseudo code for spraying procedure by ensuring sprayed search requests are forwarded to the neighbor nodes. If the procedure forwards requests to the neighbors then the result will be PASS otherwise FAIL.

S.No	Number of Nodes	Source Node, Destination Node	Search request sprayed to nodes	Result
1	15	1	6,8,10,14,4,3	PASS
2	5	4	0,1,12,13	PASS
3	9	8	5	PASS

Table 7.1 Unit testing for spraying procedure

Function Name: Waiting

Unit testing for waiting function tests the pseudo code for waiting procedure by ensuring that the nodes in waiting phase reach the destination or not. If the destination is reached by the nodes in the waiting phase then the result will be PASS otherwise FAIL.

S.No	Number of Nodes	Source Node	Nodes in waiting phase reached the destination	Result
1	15	8	YES	PASS
2	25	2	YES	PASS
3	14	1	YES	PASS

Table 7.2 Unit testing for waiting procedure

Test case for content search mechanism

Testing was performed on the constructed network with the proposed scenario. Here, network specifications like number of nodes, requesting node and pause time are taken as input and generate proposed content search mechanism and identified desired results, out of which some test cases are listed in the following table.

S. No	No. of Nodes	Pause Time	Source Node	L Value	Content Holding Peers	Content Holding Peers found in this Phase	Result
1	15	2	4	8	8	SPRAY	PASS
2	18	4	1	11	12	SPRAY	PASS
3	20	1	9	13	1,5,11,12, 13,15	SPRAY	PASS
4	15	3	7	9	13,10	SPRAY	PASS
5	15	1	7	11	1,2,3, 4,11	SPRAY	PASS
6	5	1	4	2	2	WAIT	PASS
7	5	4	4	5	3	SPRAY	PASS
8	25	2	1	21	16	SPRAY	PASS
9	25	1	1	13	2,14,12	WAIT	PASS
10	45	15	44	31	13,26,	WAIT	PASS

					38,2		
11	20	3	6	11	-	WAIT	FAIL
12	100	1	12	51	10,45, 3	WAIT	PASS

Table 7.3 Test case for content search mechanism

8. RESULT ANALYSIS

In this section performance evaluation metrics will be shown using graphs, in which epidemic P2P content search and spray and wait P2P content search mechanisms are compared.

End-to-end Delay:

Pause Time	End-to-end delivery delay in milli sec	
	Epidemic P2P content search	**Spray and wait P2P content search**
1	5	3
2	10	7
3	17	15
4	25	20
5	40	24
6	50	38

Table 8.1 End to end Delay

Time taken by packets or search requests to traverse from requesting node to content holding peers.

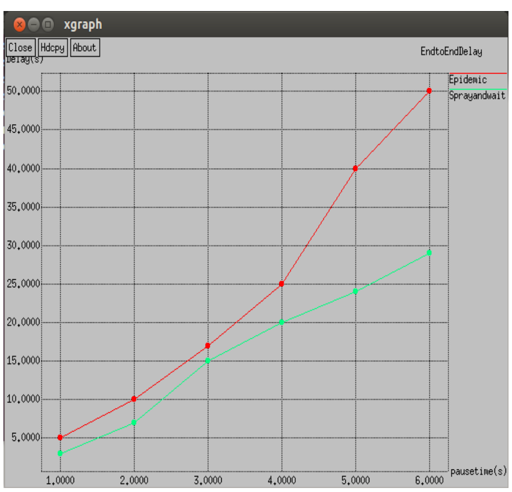


Fig 7.1 End-to-end Delay comparison

As the network grow in size and increase in pause time results in quickly identifying the content holding peer in spray and wait P2P content search mechanism.

Network Usage:

No. of Nodes	No. of total packet transmissions	
	Epidemic P2P content search	**Spray and wait P2P content search**
5	30	8
10	110	17
15	240	25
20	420	34
30	930-	50

Table 8.2 Network Usage Comparison

Number of packets generated effects the network resource usage. Number of packets generated can be called network density.

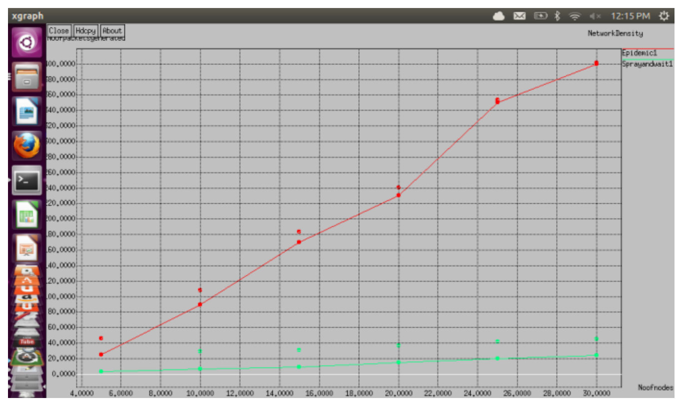


Fig 8.3 Network usage comparison

In epidemic P2P content search mechanism number of search requests generated is very high when compared to spray and wait content search mechanism due to its flooding nature. While Spray and wait P2P content generate only required number of packets that can find content holding peer.

9. CONCLUSION AND FUTURE WORK

This thesis proposed “An efficient P2P content search mechanism in ICMANs” to efficiently search for interested content in intermittently connected MANETs. Simulation shows that using this mechanism instead on flooding based mechanisms like epidemic P2P content search achieves good performance results like low end-to-end delay, low network resource usage and high throughput.

In future this mechanism can be enhanced by introducing buffer management techniques in nodes to limit the storage space restrictions on mobile devices.

REFERENCES

[1] Li Liu, Yanfang Jing, Yue Zhang, Bingbing Xi, "A Survey on P2P File Sharing Algorithms over MANETs" in Consumer Electronics Times, Vol. 2 Iss. 2, PP. 109-115, Apr. 2013.
[2] Zhen sheng Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges", in Communications Surveys & Tutorials, IEEE Volume:8 , Issue: 1, PP. 24 – 37, 05 March 2007.
[3] DeepaKundur, Zhu Liu, Madjid Merabti, and Heather Yu, "Advances In Peer-To-Peer Content Search", in Multimedia and Expo, in IEEE International Conference, PP. 404-407, July 2007.
[4] Tyson, J. (2000), Marshall Brain's HowStuffWorks, How Napster Workedhttp://www.howstuffworks.com/napster1.htm
[5] Kan, G., (2001), "Gnutella, Peer-to-Peer: Harnessing the Power of Disruptive Technologies, A. Oram (ed.), O'Reilly Press, USA".
[6] Yaozhou Ma and Abbas Jamalipour, "An Epidemic P2P Content Search Mechanism forIntermittently Connected Mobile Ad hoc Networks", in Global Telecommunications Conference, PP. 1-6, 2009.
[7] Y. Ma, M. R. Kibria and A. Jamalipour, "Optimized routing framework for intermittently connected mobile ad hoc networks," in Proc. of IEEE ICC08, Beijing, China, 19-23 May 2008.
[8] Ahmet Duran, Chien-Chung She, "Mobile ad hoc P2P file sharing", in Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE Volume:1, PP. 114-119, March 2004.
[9] Thrasyvoulos Spyropoulos, Konstantinos Psounis, Cauligi S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme forIntermittently Connected Mobile Networks", in SIGCOMM'05 Workshops, August 22–26, 2005.
[10] A. Vahdat and D. Becker. "Epidemic routing for partially connected ad hoc networks", Technical Report CS-200006, Duke University, Apr. 2000.
[11] Spyropoulos T, Psounis K, Raghavendra C (2008) Efficient routing in intermittentlyconnected mobile networks: The multiple-copy case. Networking,IEEE/ACMTransactions on 16(1):77–90, DOI 10.1109/TNET.2007.897964
[12] R. E. Shannon, "Introduction to the art and science of simulation," in Proc. Of the 30th conference on Winter simulation (WSC'98), 1989.
[13] The Network Simulator Wiki. [Online]. Available: http://nsnam.isi.edu/nsnam/ index.php/