

Study of Possible Packets Travelling Algorithm for Effective Mobile Ad-hoc Network

¹S. Radha Rammohan and ²K. Rangarajan

¹Bharath University, Selaiyur, Chennai-600073, Tamil Nadu, India

²Department of HOD-MCA, Bharath University, Selaiyur, Chennai-600073, Tamil Nadu, India

Abstract: Mobile communication system has changed the life style of the human being in the past two decades. Various research and development process was carried out to increase the effective communication system to provide sophisticated communication environment to the users. The communication system components include MAC protocol, a routing protocol and the treatment of voice packets. The telephone system performance is assessed in terms of packet transformation from one device to another. The performance is based on the percentage of blocked and dropped cells, packet loss and packet delay. The telephone system efficiency can be increased through effective packet transformation and control. The packet transformations are achieved through various routing and travelling algorithms. This study is to analyze the existing packet travelling algorithm concepts and to achieve the effective architecture system to provide good quality of service in Mobile ad-hoc network.

Key words: Mobile communication, mobile network, travelling algorithm, ad-hoc network, cellular, cell splitting, cluster

INTRODUCTION

Packet travelling algorithm for effective mobile ad-hoc networks is a topic of emerging interest in the research arena of telecommunication networks. In mobile ad-hoc wireless networks (Khalil and Shroff, 2005) multiple mobile stations communicate without the support of a centralized coordination station for the scheduling of transmissions. In this project, we determine the packet routing algorithm for wireless ad-hoc network to avoid the congestion over data transmission. This algorithm takes into account network scalability by investigating how the size of the multihop ad-hoc network impacts the quality of service. We measure the performance characteristics in terms of the percentage of blocked and dropped calls, packet loss and packet delay. For this service, we propose a complete system architecture, which includes a MAC protocol, a routing protocol and treatment of voice packets.

Mobile communication principles: Each mobile uses a separate, temporary radio channel to talk to the cell site. The cell site talks to many mobiles at once, using one channel per mobile. Channels use a pair of frequencies for communication. One frequency (the forward link) for transmitting from the cell site and one frequency (the reverse link) for the cell site to receive calls from the users. Radio energy dissipates over distance, so mobiles must stay near the base station to maintain

communication. The basic structure of mobile networks includes telephone systems and radio services. Where mobile radio service operates in a closed network and has no access to the telephone system, mobile telephone services allow interconnection to the telephone network.

Mobile telephone system using the cellular concept: Interference problems caused by mobile units using the same channel in adjacent areas proved that all channels could not be reused in every cell. Areas had to be skipped before the same channel could be reused. Even though, this affected the efficiency of the original concept, frequency reuse was still a viable solution to the problems of mobile telephony systems.

Engineers discovered that the interference effects were not due to the distance between areas, but to the ratio of the distance between areas to the transmitter power (radius) of the areas. By reducing the radius of an area by 50%, service providers could increase the number of potential customers in an area fourfold. Systems based on areas with a 1 km radius would have one hundred times more channels than systems with areas 10 km in radius. Speculation led to the conclusion that by reducing the radius of areas to a few hundred meters, millions of calls could be served.

The cellular concept employs variable low-power levels, which allow cells to be sized according to the subscriber density and demand of a given area. As the population grows, cells can be added to accommodate

that growth. Frequencies used in one cell cluster can be reused in other cells. Conversations can be handed off from cell to cell to maintain constant phone service as the user moves between cells (Fig. 1).

The cellular radio equipment (base station) can communicate with mobiles as long as they are within the range. Radio energy dissipates over distance, so the mobiles must be within the operating range of the base station. Like the early mobile radio system, the base station communicates with mobiles via a channel. The channel is made of two frequencies, one for transmitting to the base station and one to receive information from the base station.

Cellular system architecture: Increase in demand and the poor quality of existing service led mobile service providers to research ways to improve the quality of service and to support more users in their systems. Because the amount of frequency spectrum available for mobile cellular use was limited, efficient use of the required frequencies was needed for mobile cellular coverage. In modern cellular telephony, rural and urban regions are divided into areas according to specific provisioning guidelines. Deployment parameters, such as amount of cell-splitting and cell sizes, are determined by engineers experienced in cellular system architecture. Provisioning for each region is planned according to an engineering plan that includes cells, clusters, frequency reuse and handovers.

Cell: A cell is the basic geographic unit of a cellular system. The term cellular comes from the honeycomb shape of the areas into which a coverage region is divided. Cells are base stations transmitting over small geographic areas that are represented as hexagons. Each cell size varies depending on the landscape. Because of constraints imposed by natural terrain and man-made structures, the true shape of cells is not a perfect hexagon.

Cluster: A cluster is a group of cells. No channels are reused within a cluster. Figure 2 shows, a seven-cell cluster.

Frequency reuse: Because only a small number of radio channel frequencies were available for mobile systems, engineers had to find a way to reuse radio channels to carry >1 conversation at a time. The solution, which the industry adopted was called frequency planning or frequency reuse. Frequency reuse was implemented by restructuring the mobile telephone system architecture into the cellular concept.

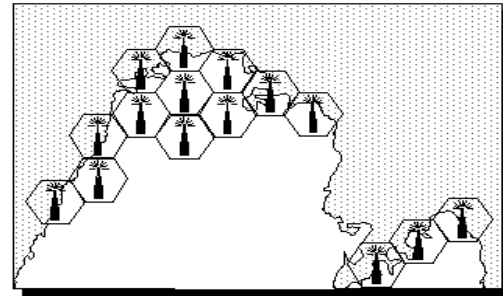


Fig. 1: Mobile telephone system using a cellular architecture

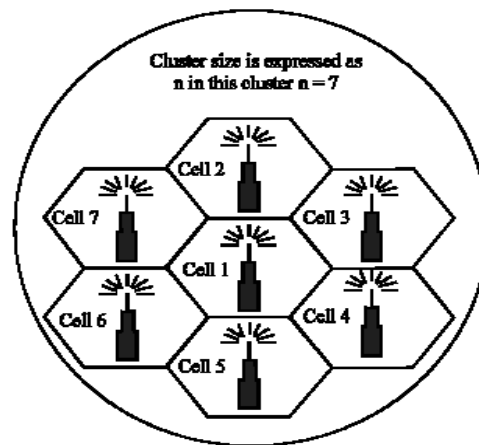


Fig. 2: A seven-cell cluster

The concept of frequency reuse is based on assigning to each cell a group of radio channels used within a small geographic area (Hu and Evans, 2004). Cells are assigned a group of channels that is completely different from neighbouring cells. The coverage area of cells is called the footprint. This footprint is limited by a boundary so that the same group of channels can be used in different cells that are far enough away from each other so that their frequencies do not interfere.

Cells with the same number have the same set of frequencies. Here, because the number of available frequencies is 7, the frequency reuse factor is 1/7. That is, each cell is using 1/7 of available cellular channels (Fig. 3).

Cell splitting Unfortunately, economic considerations made the concept of creating full systems with many small areas impractical. To overcome this difficulty, system operators developed the idea of cell splitting. As the service area becomes full of users, this approach is used to split a single area into smaller ones. In this way, urban

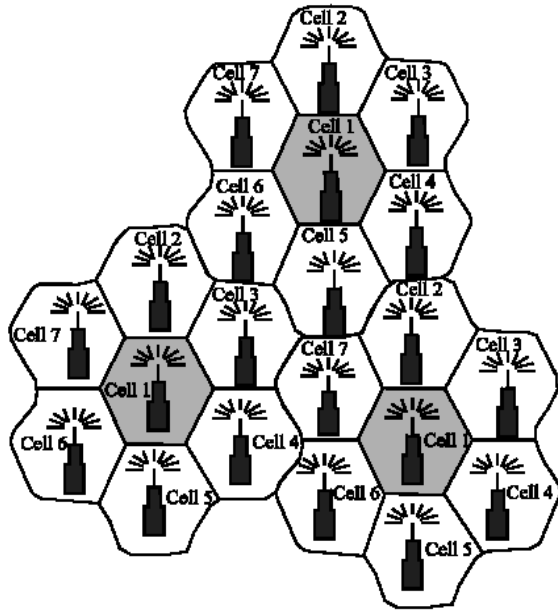


Fig. 3: Frequency reuse

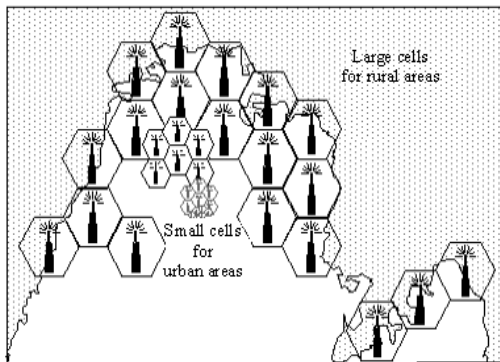


Fig. 4: Cell splitting

centers can be split into as many areas as necessary to provide acceptable service levels in heavy-traffic regions, while larger, less expensive cells can be used to cover remote rural regions (Fig. 4).

Types of clustering: Data clustering algorithms can be hierarchical or partitional.

- Hierarchical algorithms find successive clusters using previously established clusters
- Single-linkage clustering algorithm
- Partitional algorithms determine all clusters at once
- K-Means clustering algorithm

In this clustering, communication system data is travelling from one node to another using above

algorithms which is referred form wormhole attacks in wireless networks (Hu *et al.*, 2003).

Single link clustering algorithm: Assumption

- $N \times N$ proximity matrix is $D = [d(i,j)]$
- $L(k)$ is the level of the k th clustering
- A cluster with sequence number m is denoted (m)
- The proximity between clusters (r) and (s) is denoted $d[(r),(s)]$

MATERIALS AND METHODS

Methods of clustering

Step 1: Begin with the disjoint clustering having level $L(0) = 0$ and sequence number $m = 0$.

Step 2: Find the least dissimilar pair of clusters in the current clustering, say pair $(r), (s)$, according to

$$d[(r),(s)] = \min d[(i),(j)]$$

where, the minimum is over all pairs of clusters in the current clustering.

Step 3: Increment the sequence number: $m = m + 1$. Merge clusters (r) and (s) into a single cluster to form the next clustering m . Set the level of this clustering to

$$L(m) = d[(r),(s)]$$

Step 4: Update the proximity matrix, D , by deleting the rows and columns corresponding to clusters (r) and (s) and adding the newly formed cluster. The proximity between the new cluster, denoted (r, s) and old cluster (k) is defined in this way:

$$d[(k),(r,s)] = \min d[(k),(r)], d[(k),(s)]$$

Step 5: If all objects are in one cluster, stop. Else, go to step 2.

K-Means algorithm: Clustering analysis is to group data in such a way that similar objects are in one cluster and objects of different clusters are dissimilar. The k-Means algorithm basically consists of three steps:

- An initial set of 'k' (where, 'k' is the number of clusters) so-called centroids, i.e., virtual points in the data space is randomly created
- Every point of the data set is assigned to its nearest centroid

- The position of the centroid is updated by means of the data points assigned to that cluster. In other words, the centroid is moved towards the center of its assigned points
- This is done until no centroid was shifted in one iteration resulting in 'k' subsets/cluster

Scheduling algorithms: Scheduling algorithms determine, which packet is served next among the packets in the queue (s). The scheduler is positioned between the routing agent and above the MAC layer. All nodes use the same scheduling algorithm. We consider, the conventional scheduling (priority scheduling) typically used in mobile ad-hoc networks and also propose other applicable scheduling policies to study. All scheduling algorithms studied are non-preemptive.

As a buffer management algorithm, the drop tail policy is used with no-priority scheduling. The drop tail policy drops incoming packets when the buffer is full. For the scheduling algorithms that give high priority to control packets, we use different drop policies for data packets and control packets when the buffer is full. When, the incoming packet is a data packet, the data packet is dropped. When, the incoming packet is a control packet, we drop the last enqueued data packet, if any exists in the buffer, to make room for the control packet. If all queued packets are control packets, we drop the incoming control packet. We explain, the scheduling algorithms as analyzed below.

Scheduling algorithms for analysis of giving high priority to control traffic: In on-demand routing protocols such as DSR, under frequent topology changes, delivering control packets (routing packets) quickly can be more important than in proactive routing protocols for propagating route discoveries or route changes quickly. To study the effect of timely delivery of control packets in on demand and proactive routing protocols, we compare a scheduling algorithm that does not distinguish control packets from data packets with a scheduling algorithm that gives high priority to control packets.

No-priority scheduling: No-priority scheduling services both control and data packets in FIFO order. We include this scheduling algorithm in contrast with the effect of giving high priority to control packets.

Priority scheduling: Priority scheduling gives high priority to control packets. It maintains control packets and data packets in separate queues in FIFO order. Currently, this scheme is used in most comparison studies about mobile ad-hoc networks.

RESULTS AND DISCUSSION

Scheduling algorithms for analysis of setting priorities in data traffic: After examining the effects of giving priority to control traffic, we look at the effects of setting priorities in data traffic based on secure routing in ad-hoc networks (Zhen and Srinivas, 2004). We devise different scheduling algorithms by using distance metrics, considering fairness and applying the multiple roles of nodes as both routers and data sources. All the scheduling algorithms we explain below give higher priority to control packets than to data packets. Their differences are in assigning weightage or priority among data queues.

A class of scheduling algorithms uses distance metrics to set priorities in data traffic. It is well understood that in a single node if the task sizes are known, Shortest-Remaining Processing-Time (SRPT) scheduling is the policy that minimizes mean response time. We apply this concept to an ad-hoc network with multiple nodes. Although, remaining processing time is not known in many cases, in a network we can assume that the remaining processing time of a packet is likely to be proportional to the Adistance (remaining hops or physical distance) from a forwarding node to a destination. We study weighted-hop scheduling with DSR and weighted-distance scheduling with GPSR.

Besides the scheduling algorithms using distance metrics, we study round robin scheduling and greedy scheduling to see how fairness or greediness of a node's packet forwarding affects the performance.

Weighted-hop scheduling: Weighted-hop scheduling gives higher weightage to data packets that have fewer remaining hops to traverse. The fewer hops a packet needs to traverse the more the potential it has to reach its destination quickly and the less queuing it incurs in the network. In Fig. 5, the data packet scheduler serves packets in weighted round robin fashion. We use a weighted round robin scheduler instead of a static priority scheduler since the weighted scheduler guarantees all service classes at least the configured amount of service chances, thus avoiding starvation. If we consider packet length variations to allocate the correct proportion of bandwidth, we can use weighted fair queuing or deficit round robin. The data queue of the class C_i maintains data packets whose number of remaining hops to traverse is i . When the number of remaining hops of a data packet is greater than n (the number of data queues), the data packet is classified as C_n . For example, if the remaining number of hops of a data packet is 2, it belongs to C_2 . The data queue of the class C_i receives weight $(1 < i < n)$.

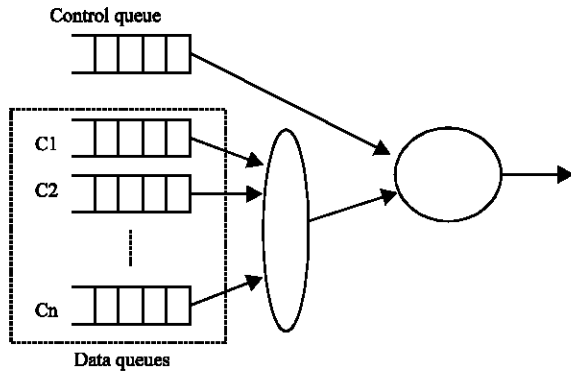


Fig. 5: Packet scheduler

In the DSR protocol, each data packet header carries a complete list of nodes through which the packet should travel.

In DSR, we thus, obtain the remaining hops to traverse from the packet headers. However, in other routing protocols, including Ad-hoc On-demand Distance Vector routing (AODV), we would obtain this information from the routing table, which stores the remaining hops to destinations.

Weighted-distance scheduling: We also consider, a scheduling algorithm that uses physical distance with GPSR. Using physical distance may be a unique feature of ad-hoc wireless networks. In the GPSR protocol, each data packet carries a destination's position. Nodes that are close in physical distance are likely to be close in the network topology (i.e., a small number of hops from each other). As the remaining physical distance to a destination decreases, the remaining hops to a destination in the network topology are likely to decrease.

The weighted-distance scheduler is also a weighted round robin scheduler. It gives higher weightage to data packets that have shorter remaining geographic distances to the destinations.

The remaining distance (remaining distance) is defined as the distance between a chosen next hop node and a destination. Each class C_i ($1 < i < n$) is determined by the virtual hop:

$$\text{Virtual hop} = \left\lceil \frac{\text{Remaining distance}}{\text{Quantization distance}} \right\rceil + 1$$

where, quantization distance is a distance for mapping the physical distance into the class. For simplicity, we choose this uniform quantization method. Better quantization methods might be used for further improvement. When, the virtual hop of a data packet is greater than n (number

of data queues), the data packet is classified as C_n . For example, if $n = 8$, quantization distance is 250 m and remaining distance is 350 m, the virtual hop = 3 and the packet belongs to C_3 . The data queue of the class C_i receives weight W_i ($1 < i < n$). The higher weight is assigned to the lower class.

Round robin scheduling: Round robin scheduling maintains per-flow queues. We identify each flow by a source and destination (IP address, port number) pair.

In Fig. 2, each C_i is equal to a flow. In round robin scheduling, each flow queue is allowed to send one packet at a time in round robin fashion. We evaluate round robin scheduling to see the effect on performance of having an equal service chance among flows.

Greedy scheduling: In the greedy scheduling scheme, each node sends its own data packets (packets it has generated) before forwarding those of other nodes. The other nodes' data packets are serviced in FIFO order.

Greedy algorithm: A greedy algorithm is any algorithm that follows the problem solving meta-heuristic of making the locally optimum choice at each stage (Yang *et al.*, 2004) with the hope of finding the global optimum.

In general, greedy algorithms have five pillars:

- A candidate set, from which a solution is created
- A selection function, which chooses the best candidate to be added to the solution
- A feasibility function, that is used to determine if a candidate can be used to contribute to a solution
- An objective function, which assigns a value to a solution, or a partial solution
- A solution function, which will indicate when we have discovered a complete solution

Greedy algorithms produce good solutions on some mathematical problems, but not on others. Most problems for which they work well have two properties:

Greedy choice property: Whatever choice seems best at the moment and then solve the sub-problems that arise later. The choice made by a greedy algorithm may depend on choices made so far, but not on future choices or all the solutions to the sub-problem. It iteratively makes one greedy choice after another, reducing each given problem into a smaller one. In other words, a greedy algorithm

never reconsiders its choices. This is the main difference from dynamic programming, which is exhaustive and is guaranteed to find the solution. After every stage, dynamic programming makes decisions based on all the decisions made in the previous stage and may reconsider the previous stage's algorithmic path to solution.

CONCLUSION

In the above specified algorithms, which analyzed and referred Prevention Algorithm in Mobile ad-hoc Networks (Sunchoi *et al.*, 2008) according to its property Single link and K-Means algorithms used to identify the path of the packet travelling. In the determined path travelling process can be effectively scheduled using Weighted-hop Scheduling, Weighted-distance Scheduling, Round Robin Scheduling and Greedy scheduling.

The identified algorithms to be executed in the specified environment and optimization to be carried out for the effective and Quality of Service in a Mobil ad-hoc network. Once the packet travel path identification and scheduling is optimized then we can provide the quality of service.

REFERENCES

- Hu, Y.C., A. Perrig and D.B. Johnson, 2003. Packet leashes: A defense against wormhole attacks in wireless networks. INFOCOM, 22nd Annual Joint Conf. IEEE Computer and Commun. Soc. IEEE Publication, March 30-April 3, 3: 1976-1986. ISBN: 0-7803-7752-4. INSPEC: 7853884. http://ieeexplore.ieee.org/xpl/free-abs_all.jsp?arnumber=1209219.
- Hu, L. and D. Evans, 2004. Using directional antennas to prevent wormhole attacks. In Network and Distributed System Security Symposium (NDSS). The Internet Society, Annual Network and Distributed System Security Symposium Catamaran Resort Hotel. San Diego, California, pp: 3750-3772. <http://portal.acm.org/citation.cfm?id=1276793>.
- Khalil, I., S. Bagchi and N.B. Shroff, 2005. LITEWOP: A lightweight countermeasure for the wormhole attack in multihop wireless networks. In: Dependable Systems and Networks (DSN), Sch. of Electr. Comput. Eng., Purdue Univ., West Lafayette, IN, USA, pp: 612-621. DOI: 10.1109/DSN.2005.58. ISBN: 0-7695-2282-3. INSPEC: 8633984. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1467835.
- Sunchoi, Doo-young, Kim Do-hyeon, Lee Jae-il and Jung, 2008. WAP: Wormhole Attack Prevention Algorithm in Mobile ad-hoc Networks. Div. Elect. Comput. Eng., Hanyang Univ., Seoul; Sensor Networks, Ubiquitous and Trustworthy Computing. SUTC. Publication, June 11-13, pp: 343-348. DOI: 10.1109/SUTC.2008.49. ISBN: 978-0-7695-3158-8. INSPEC: 10060052.
- Yang, H., H. Luo, F. Ye, S. Lu and L. Zhang, 2004. Security in mobile ad-hoc networks: Challenges and solutions. Wireless Commun. IEEE., 11 (1): 38-47. DOI: 10.1109/MWC.2004.1269716. INSPEC: 7943665. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=1269716&isnumber=28409.
- Zhen, J. and S. Srinivas, 2004. Preventing replay attacks for secure routing in ad-hoc networks. Publisher Springer Berlin/Heidelberg, pp: 140-150. DOI: 10.1007/b14230. ISBN: 978-3-540-20260-8. <http://www.springerlink.com/content/tj87g19lx5b6eydh>.