Load Aware Broadcast in Mobile Ad Hoc Networks

Md. Tanvir Al Amin, Sukarna Barua, Sudip Vhaduri, and Ashikur Rahman
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh
Email: tanviralamin@gmail.com, {sukarna_barua, sudipcse03}@yahoo.com, ashikur@cse.buet.ac.bd

***Abstract*—In a wireless ad hoc network, the main issue of a good broadcast protocol is to attain maximum reachability with minimal packet forwarding. Existing protocols address this issue by utilizing the knowledge of up to 2-hop neighbors to approximate an MCDS (minimum connected dominating set) via heuristics derived from techniques known as Self pruning and Dominant pruning. Our experiments show that, using these greedy choice heuristics result in a biased load distribution throughout the network. Some nodes become heavily loaded and consequently packets through those nodes, whether unicast or broadcast, experience significantly larger delay. Contention and collision also increase at some regions, while they are relatively low at other regions. In this paper we address these issues, and propose various methods to evenly distribute the load caused by broadcast packets. Our algorithms take various reactive measures to dynamically include less loaded nodes in the forward list, while maintaining total number of packet forwards low. Detailed simulation using ns-2 shows fair scheduling of resources and significant improvement in distribution of packet forwarding load, packet delay, latency and overall performance.**

I. Introduction

A wireless ad hoc network finds applications in battlefields, rescue sites, sensor networks, wireless classrooms or other places where infrastructure support is either expensive or irrelevant. Here, the nodes can be highly dynamic or mobile in nature and frequent topology change, path break or mobility created congestion are common events. Thus, network wide broadcast [1] [2] operation is more likely than in wired scenario. In fact, several newer applications related to pervasive computing or multimedia over ad hoc network demand increased amount of broadcast operation. These applications emerge as wireless or mobile devices become more and more ubiquitous with higher processing and multimedia capability.

Basic idea of broadcasting is flooding. Ni *et al.* [1] discuss in detail, how much adverse the "Broadcast storm" can be, if flooding is done blindly in a wireless ad hoc network. Lim and Kim [3] formulate the solution to optimal broadcast problem in an ad hoc network as finding a connected dominating set of the minimum size (MCDS). To approximate MCDS in a distributed manner, they describe two heuristic strategies– Self pruning (SP) and Dominant pruning (DP). Self pruning uses direct neighborhood information and a node itself decides whether it will retransmit a received packet. On the other hand, dominant pruning uses extended neighborhood information and a transmitting node specifies in the forwarded packet, which of its neighbors should rebroadcast it. Each node u determines a forward list as a subset of its one-hop neighbors, whose transmissions cover all two-hop neighbors of u. This computation reduces to a set cover problem, where greedy set cover approximation is used. These algorithms are the baseline for several other broadcast protocols like SBA [4], Multipoint Relaying [5], AHBP [6] [7] or LENWB [8].

Unfortunately, prime concern of all of these protocols is reducing number of packet forwards without sacrificing reachability. When broadcast is a small portion of total traffic, these protocols perform nice. But multimedia operations like video conferencing or collaborative computing may require almost all of the traffic to be broadcast. With increased broadcast load, a previously unnoticed problem becomes apparent–the unbalanced distribution of broadcast load.

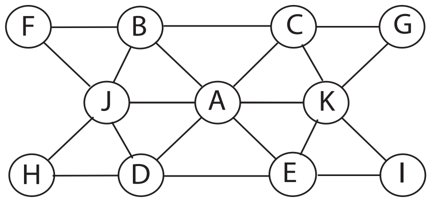
In this paper we illustrate that, network wide broadcast can create more unbalanced condition in the network than unicast. As packets are delivered to all of the nodes, always trying to select the nodes in the optimal broadcast tree creates a biased distribution of load. But in a shared and collaborative environment like ad hoc network, a good broadcast protocol should be aware of the present load condition of the nodes to ensure fair scheduling of resources and improved performance by decreased delay and jitter. To the best of our knowledge, this problem has not been addressed in depth yet.

We have described schemes for distributing the broadcast load as evenly as possible without any significant increase in the number of packet forwards. We have developed reactive strategies based on feedback of a load parameter, to dynamically include less loaded nodes in the forward list. We have modified the neighborhood based heuristic used in Dominant Pruning (DP) algorithm [3] to incorporate load balancing. We also present a better heuristic "RL" to use with that algorithm. Finally, we have developed a new algorithm "SRL," which along with "RL" heuristic, shows best performance.

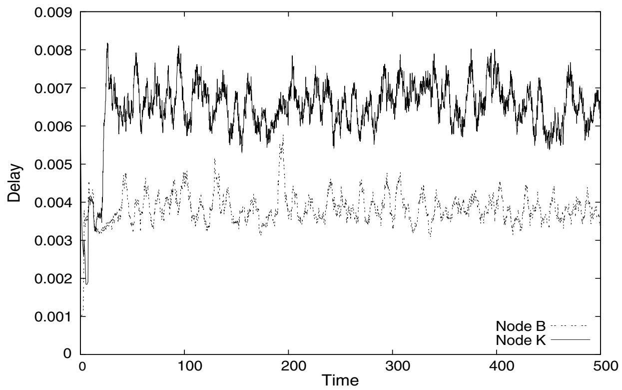
II. Broadcast & Load

The effective load on a mobile host is due to two main reasons–Queuing delay and Contention. Moreover, effect of load is different for unicast and broadcast packets. For a unicast packet, at routing layer, load is observed only at the intermediate nodes. At MAC layer, after each transmission, contention is observed at the neighbors of the transmitting node only. But in case of a broadcast packet, to achieve full reachability, non-leaf nodes in the broadcast tree [3] have to do the forwarding. A transmission is rendered to create "unwanted" contention for a node, only when it is redundant.

For example, in Fig. 1(a), if A is the broadcast source, J and K must forward to reach F, H, I, G. But then B, D, C, E



(a) An example topology



(b) Delay of Node B and Node K

Fig. 1. Experimental topology and delay of nodes

TABLE I
COMPARISON OF NUMBER OF PACKET FORWARDS BY NODES

Node	Forwards	Node	Forwards	Node	Forwards
A	964	D, E, H	2	J	962
B	1993	F	1988	K	2941
C, I	1	G	1980		

receives it again. From the forwarding of J and K, A receives the packet 2 more times. Thus, medium contention is increased compared to unicast. Moreover, as there is no provision for RTS, CTS and ACK for broadcast, CSMA must be used. Collision is more likely to occur due to hidden station problem.

Regardless of "wanted" or "unwanted," a single broadcast packet creates load throughout the network. Greater delay is experienced by other sessions. For example, in the topology of Fig. 1(a), node A broadcasts CBR packets at a rate of 2 packets/sec. CBR unicast at 4 packets/sec is going from G to I, which takes path $G-K-I$ and from F to C taking path $F-B-C$. DP selects node J and K to forward the broadcasts. Thus node B is forwarding only unicast traffic, and node K is forwarding both unicast and broadcast. As DP always selects node J and K to cover 2-hop neighbors $\{F, G, H, I\}$, these nodes always have a load due to the broadcast. Hence the unicast session passing via K, experiences more delay than the session passing via B. This is illustrated in Fig. 1(b), where K has significantly larger delay than B. Even if load aware routing protocols [9] [10] are used, unicast session from G to I could not be made better, because any path from G to I passes via at least one node from $\{J, A, K\}$ and here all three of these are loaded.

There is another aspect of broadcast load balancing. Table I enlists number of packet forwards by each node in 500 seconds. Nodes C, D, E all are relatively free while node J and K are highly loaded. This is an unfair scheduling of resources. Nodes J, K will experience faster battery depletion due to the broadcast load. So, it is expected to distribute it by selecting the forward list between $\{B, K, D\}$ and $\{C, J, E\}$ to cover the 2-hop neighbors of A.

Thus the objective of load aware broadcast is to distribute the broadcast load throughout the network so that the perceived load impact on other active sessions is lower and consumption of resources in various nodes is also as fair as possible. But, number of packet forwards should be kept near optimal, otherwise effect of broadcast storm will dominate again.

III. LOAD AWARE BROADCASTING

Our algorithms for load aware broadcasting use reactive strategies to distribute the load. Prior to broadcast packet forwarding, a node creates the forward list considering the "load feedback" from neighboring nodes. Target is to select the less loaded nodes, which tend to do the broadcasting earlier.

A. Combining neighborhood information and load

While constructing the forwarding list, considering only load information of the neighbors may increase the total number of packet forwards. In the topology of Fig. 1(a), assume node J and K have higher loads than nodes B, C, D, E. Node A is broadcasting a packet. Now if it selects less loaded nodes for constructing the forwarding list, then it requires 4 nodes (i.e. B, C, D, E) to forward the packet. However, selecting $\{J, K\}$ despite of their highest load requires only 2 packet forwarding. This implies that, considering only load may increase packet forwarding in a large scale which may result in contention and collision in the network. Therefore, the neighborhood information should be considered also.

B. Load incorporated Dominant Pruning–DNL and DRL

We describe two heuristics NL (Neighbor and Load) and RL (Rank and Load) as function of load and neighborhood information, to use with DP algorithm to find the forward list [3]. If DP is used with NL heuristic, we call it DNL (Dominant pruning with Neighbor Load) and if RL heuristic is used, we call it DRL (Dominant pruning with Rank Load).

Let, the set of 1-hop neighbors of v_p is $N(v_p)$, load for $v_k \in N(v_p)$ is $L(v_k)$, set of up to 2-hop neighbors of v_p is $N(N(v_p))$, set of 2-hop neighbors of v_p, $N_2(v_p) = N(N(v_p)) - N(v_p) - \{v_p\}$.

To forward a packet received from node v_q, node v_p builds the forward list $F_{p,q} \subset X(p,q)$, iteratively to cover the nodes of $Y(p,q)$ where,

$$X_{p,q} = N(v_p) - \{v_q\} - N(v_q) \quad (1)$$

$$Y_{p,q} = N_2(v_p) - N(v_q) - N(N(v_q) \cap N(v_p)) \quad (2)$$

At the start of iteration i, $i >= 0$, let the partial forward list is $F_{p,q}(i)$. Initially $F_{p,q}(0) = \emptyset$.
Consider the bipartition, $\mathbf{B}_i : X_{p,q,i} \rightarrow Y_{p,q,i}$, where $X_{p,q,i} = X_{p,q} - F_{p,q}(i)$ and $Y_{p,q,i} = Y_{p,q} - N(F_{p,q}(i))$.

For $x \in X_{p,q,i}$ and $y \in Y_{p,q,i}$ an edge (x, y) exists in this bipartition *iff*, $y \in N(x)$.
$\mathbf{B}_i(x) = \{y : (x, y) \in E(\mathbf{B}_i)\} = N(x) \cap Y_{p,q,i}$ i.e. the set of nodes in $Y_{p,q,i}$ that are covered by x.
$\mathbf{B}_i^{-1}(y) = \{x : (x, y) \in E(\mathbf{B}_i)\}$ is the set of nodes in $X_{p,q,i}$ that covers y.

Among all the nodes in $X_{p,q,i}$, the node v_k with maximum heuristic value $NL(v_k, i)$ or $RL(v_k, i)$ (described below) is selected, and the forward list is updated as $F_{p,q}(i + 1) = F_{p,q}(i) \cup \{v_k\}$.

1) NL heuristic: At iteration i, NL heuristic for $v_k \in X_{p,q,i}$ is,

$$NL(v_k, i) = (1 - \alpha) \times \frac{|\mathbf{B}_i(v_k)|}{\max_{v_j \in X_{p,q,i}}\{|\mathbf{B}_i(v_j)|\}} + \alpha \times \frac{\max_{v_j \in X_{p,q,i}}\{L(v_j)\}}{L(v_k)} \quad (3)$$

$0 <= \alpha <= 1$ is called "Load awareness parameter" which determines how much emphasis is given on load balancing. If $\alpha = 0$, the algorithm becomes naive greedy set cover, and if $\alpha = 1$, it becomes purely load based.

2) RL heuristic: In RL heuristic, instead of taking the number of uncovered neighbors, we incorporate rank of a node in $X_{p,q,i}$. We define rank for each node $x \in X_{p,q,i}$,

$$\Re_i(x) = \sum_{y \in \mathbf{B}_i(x)} \frac{1}{|\mathbf{B}_i^{-1}(y)|} \quad (4)$$

Consider Fig. 1(a). Here $N(A) = \{B, C, J, K, D, E\}$, $N_2(A) = \{F, H, I, G\}$. If A is to forward a packet received from J, then $v_q = J$, $v_p = A$, $X_{A,J} = \{C, K, E\}$ and $Y_{A,J} = \{I, G\}$; $\mathbf{B}_0(C) = \{G\}$, $\mathbf{B}_0(E) = \{I\}$, $\mathbf{B}_0(K) = \{I, G\}$. Then, $\Re(C) = 0.5$, $\Re(E) = 0.5$, $\Re(K) = 0.5 + 0.5 = 1$.

RL heuristic for v_k is,

$$RL(v_k, i) = (1 - \alpha) \times \frac{\Re_i(v_k)}{\max_{v_j \in X_{p,q,i}}\{\Re_i(v_j)\}} + \alpha \times \frac{\max_{v_j \in X_{p,q,i}}\{L(v_j)\}}{L(v_k)} \quad (5)$$

C. SRL algorithm with RL heuristic

To build the forward list, RL heuristic can be used in a different manner resulting in a new algorithm "SRL" (Sorted Rank Load) (Algorithm 1). Instead of taking the "best" node from $X_{p,q,i}$ and checking which nodes of $Y_{p,q,i}$ are covered, this algorithm selects node y with minimum $|\mathbf{B}_i^{-1}(y)|$ from $Y_{p,q,i}$ and covers it using nodes from $\mathbf{B}_i^{-1}(y)$. If $|\mathbf{B}_i^{-1}(y)| > 1$, the node with maximum RL heuristic value is selected.

D. Capturing the load

Lee and Gerla [10] consider number of packets in the interface queue as a load metric. Hassanein and Zhou [11] use total number of routes passing through a node and its neighbors. Wu and Harms [12] take summation of number of packets being queued at the mobile node and its neighboring nodes. Song *et al.* [9] uses packet delay as the parameter. If the arrival and successful transmission time of the i^{th} packet is a_i and d_i, then they compute estimated average delay $q_i^k = (1 - \beta)q_{i-1}^k + \beta(d_i - a_i)$ where $0 <= \beta <= 1, i > 1$. This exponential average delay equation is more appropriate, since both queuing delay and contention delay are included. However, delay values $(d_i - a_i)$ in general contain frequent spikes. Small β causes these spikes to be transferred to the average, and large β causes the average to be oversmoothed.

Hence, we use a moving average window W for load estimation. At time t, $W(t) = \{(d_i - a_i) : d_i >= t - T_l\}$, i.e. delay values for last T_l seconds are remembered. Then load $L(t) = Mean\{W(t)\}$. Experiment shows that taking $T_l = 5$ provides excellent smoothing.

Algorithm 1 Procedure SRL(v_p,v_q)

$F_{p,q}(0) \leftarrow \emptyset$, $Y_{p,q,0} \leftarrow Y_{p,q}$, $X_{p,q,0} \leftarrow X_{p,q}$, $i \leftarrow 0$
while $Y_{p,q,i} \neq \emptyset$ **do**
  Select node $y \in Y_{p,q,i}$ with minimum $|\mathbf{B}_i^{-1}(y)|$
  Let $S = \mathbf{B}_i^{-1}(y)$
  if $|S| = 1$ **then**
    Select the only node $v_k \in S$
  else
    Select $v_k \in S$ such that $RL(v_k, i)$ is maximum
  end if
  $F_{p,q}(i + 1) \leftarrow F_{p,q}(i) \cup \{v_k\}$
  $X_{p,q,i+1} \leftarrow X_{p,q,i} - \{v_k\}$
  $Y_{p,q,i+1} \leftarrow Y_{p,q,i} - \mathbf{B}_i(v_k)$
  $i \leftarrow i + 1$
end while

E. Exchanging load and neighborhood information

Each node piggybacks its present load information into the data packets sent or forwarded by it. This information is also piggybacked into periodic hello messages; hence an idle node also updates its load information.

To pass neighborhood information, "Dynamic Hello" scheme can be used for efficiency. Instead of sending periodic hello messages, a node piggybacks neighborhood information in the data packets at fixed intervals. If there is no data packet to send or forward for a predefined amount of time, the node explicitly sends a hello message.

IV. Simulation Model and Evaluations

A. Simulation Environment

We have built a detailed simulation model using *ns-2* [13] with wireless extensions. Bit rate of 2Mbits/sec and a radio range of 250m are used. For the MAC layer, IEEE 802.11 DCF [14] is used. Each simulation runs for 500 seconds of virtual time.

1) Scenario Generation: 50 mobile hosts (speed $10ms^{-1}$) are placed in a $1500 \times 300m^2$ flat area. By differing pause times in a random waypoint mobility model, various degrees of mobiliy are generated. For each pause time, mean of 5 randomly generated scenarios is taken.

2) Traffic Generation: Traffic sources are CBR. 5 to 30 nodes are randomly chosen as broadcast sources, each of which broadcasts 512 bytes packets at a rate of 4 packets/sec.

B. Performance Metrics

The performance metrics we observe are:

1) Mean Packet Forward (MPF): MPF is the average number of packet forwards by each node. We observe MPF to check how many extra packet forwards are incurred on average, due to load aware broadcasting.

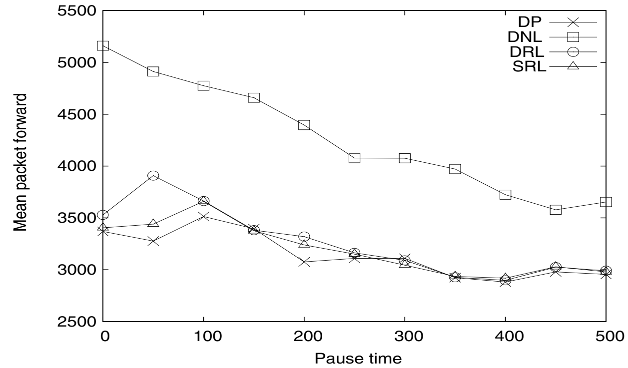
2) Standard deviation of Packet Forward (SPF): SPF is the standard deviation of number of packets forwarded by each node. It is observed as the "quality" of load balancing. Smaller value of SPF implies fairer load distribution.

3) Average Broadcast Latency (ABL): ABL is the mean interval from the time a broadcast packet is initiated to the time the last host receives it.

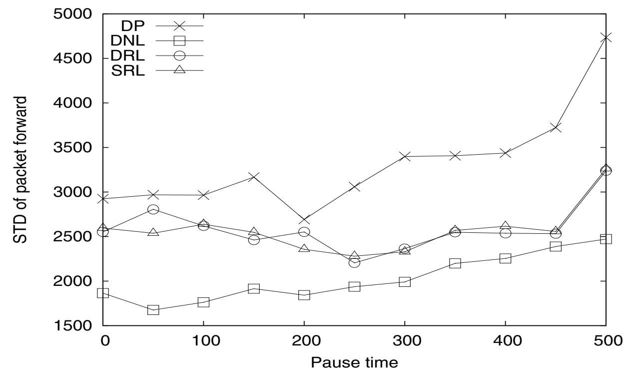
C. Simulation Results

We compare the performance of DNL, DRL and SRL with DP algorithm. Value of α is taken 0.15. By experiments, we find that choosing alpha between 0.1 to 0.3 does excellent load balancing without causing too much extra packet forwards.

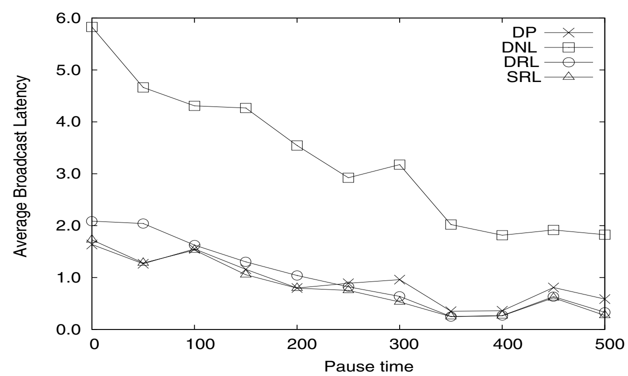
1) Performance vs. Mobility: We simulated various degrees of mobility by varying pause times from 0 to 500 seconds with an interval of 50 seconds. Fig. 2 illustrates MPF, SPF and ABL as a function of pause time.



(a) Mean Packet Forward



(b) STD Packet Forward



(c) Average Broadcast Latency

Fig. 2. Performance metrics with varying Pause time (30 sources)

Fig. 2(a) shows that, DP dominates the other three in reducing the number of packet forwards in highly mobile environments. As mobility decreases, difference in number of packet forward decreases also. SRL seems to outperform DRL, in almost all degrees of mobility and again, this difference decreases with decreasing mobility.

DNL algorithm, which is the outcome of incorporating NL heuristic in DP algorithm, clearly shows worst performance here. However, this high packet forward is justified in Fig. 2(b), where DNL outperforms the rest two in SPF. DRL and SRL perform almost similar in balancing the load. As one can imagine, DP performs worst here, as there is no load balancing scheme incorporated. Difference of SPF between load aware schemes and DP is low in highly mobile environments. This is because, dynamically changing neighborhood list due to mobility unconsciously does some load balancing here.

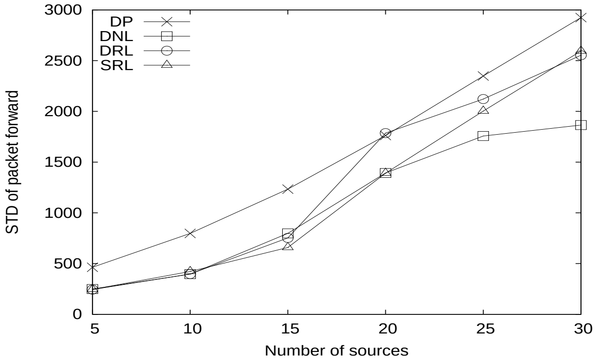
In case of broadcast latency (Fig. 2(c)), for highly mobile conditions, SRL and DP perform almost similar, but DRL and SRL outperform it and reduce latency as mobility decreases. Latency decreases with decreasing mobility for all schemes, as expected. Considering latency, DNL performs worst here.

Trend of the metrics with respect to mobility is almost same for all schemes. This is also expected, as each of the proposed schemes are based on the fundamental set cover idea of DP.

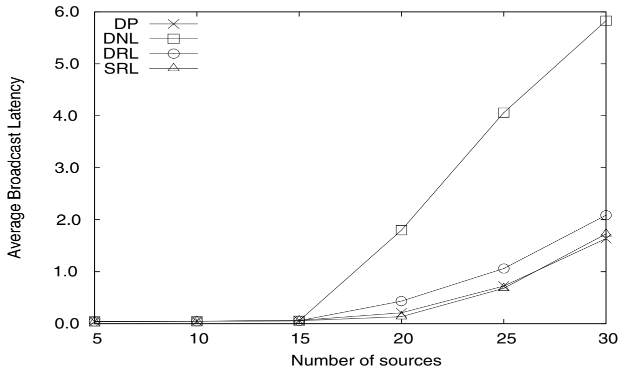
2) Varying number of sources: Here we evaluate the performance, as a function of network load. Load is varied by varying number of broadcast sources. Performance of fully mobile (Fig. 3) and fully static (Fig. 4) cases are shown here.

For static condition, there is no significant difference in MPF of any of the schemes up to 25 sources (100 packets/sec in the network). For the fully mobile case, similar characteristics is observed only up to 15 sources (60 packets/sec in the network) due to the extra load created by mobility. For number of sources more than that, SRL and DRL show slight increase in MPF while the increment for DNL is large.

Performance of load balancing between load aware schemes and DP is also significant in static case (Fig. 4(a)). For the mobile case shown in Fig. 3(a), DNL shows better load balancing performance and worse packet forwarding performance. With increased load, quality of load balancing is decreased (i.e. SPF increases) for all of the schemes. However, difference with DP increases, which is clearly visible in Fig. 4(a). DP and SRL

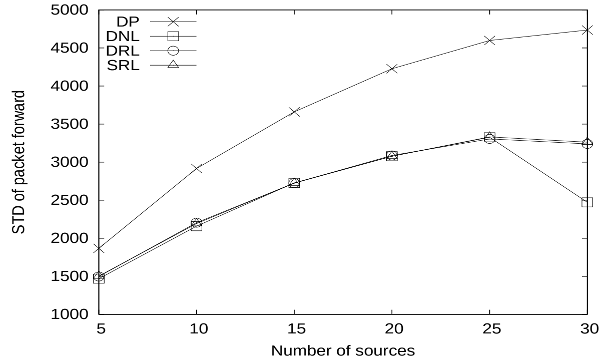


(a) STD Packet Forward

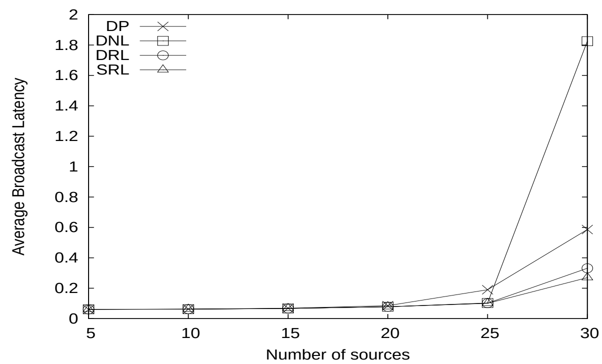


(b) Average Broadcast Latency

Fig. 3. Varying number of sources (Pause time = 0 sec)



(a) STD Packet Forward



(b) Average Broadcast Latency

Fig. 4. Varying number of sources (Pause time = 500 sec)

are competitive in case of latency as shown in Fig. 3(b) and Fig. 4(b), but SRL results in lower values.

In all load conditions, DNL shows decreased performance considering MPF and ABL. It tends to distribute the load more and decrease SPF, hurting other performance parameters. In fact, RL heuristic performs far more better than NL, and SRL gives nearly optimal packet forwards, better load balancing and decreased latency with respect to DP.

V. Conclusions

We have discussed a new problem–load aware broadcasting. We have shown why is it necessary to balance the broadcast load and proposed two new heuristics: NL and RL, to calculate the forward list in DP algorithm. A new algorithm SRL is proposed also, which calculates the forward list in a different manner. Simulation results show that, our methods decrease SPF by more than 30% in a medium loaded environment while keeping MPF near optimal, and this performance is even better with increased load. Moreover, end to end delay, jitter, latency and other performance parameters improve also. Hence our method is perfectly applicable to a wide spectrum starting from low load conditions to high load broadcasting needs.

Acknowledgment

The authors would like to acknowledge Bangladesh University of Engineering & Technology (BUET) for its generous support and research grant to make this work publishable.

References

[1] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of MobiCom*, USA, 1999.

[2] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proc. 3rd ACM Int. Symposium on Mobile ad hoc networking & computing*, Switzerland, 2002.

[3] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proc. ACM MSWiM*, USA, 2000, pp. 61–68.

[4] W.Peng and X.Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of MobiHoc 2000*, MA, USA, 2000.

[5] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," INRIA, France, Technical Report 3898 - Rapport de recherche, 2000.

[6] W. Peng and X. Lu, "An efficient broadcast protocol for mobile ad hoc networks," *Journal of Science and Technology*, 2002.

[7] M. D. Colagrosso, "Intelligent broadcasting in mobile ad hoc networks: Three classes of adaptive protocols," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. 10216, 2007.

[8] J. Sucec and I. Marsic, "An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks," CAIP Center, Rutgers University, Piscataway, NJ, USA, CAIP Technical Report 248, 2000.

[9] J. Song, V. Wong, and V. Leung, "Load-aware on-demand routing (LAOR) protocol for mobile ad hoc networks," in *Proc. The 57th IEEE Semiannual VTC*, Jeju, Korea, April 2003.

[10] S. Lee and M. Gerla, "Dynamic load-aware routing in ad hoc networks," in *Proc. IEEE ICC'01*, Finland, June 2001.

[11] H. Hassanein and A. Zhou, "Routing with load balancing in wireless ad hoc networks," in *Proc. ACM MSWiM*, Italy, June 2001.

[12] K. Wu and J. Harms, "Load-sensitive routing for mobile ad hoc networks," in *Proc. IEEE ICCCN*, Scottsdale, AZ, October 2001.

[13] "The ns manual (formerly ns notes and documentation)." [Online]. Available: http://www.isi.edu/nsnam/ns/doc/index.html

[14] I. S. Department, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1997, IEEE standard 802.11-1997.