A Path Maintenance Protocol for Uninterrupted Communication in Ad hoc Wireless Networks Using Self-Adjusting Transmission Range

Krishna Paul
Cognizant Technology Solutions
Sector V, Saltlake
Calcutta 700 091 India
krishnap@cal.cts-corp.com

S. Bandyopadhyay
PricewaterhouseCoopers Ltd.
Sector V, Saltlake
Calcutta 700 091 India
somprakash.bandyopadhyay@in.pwcglobal.com

ABSTRACT

There has been a growing interest in mobile, ad hoc wireless networks in recent years. In this environment, all nodes are normally assumed to operate with a fixed transmission range. However, it has been observed that a low transmission range of mobile hosts will not guarantee proper connectivity among them to ensure effective communication. On the other hand, if the transmission range is high, it will ensure connectivity but will increase collision and congestion of control packets, which will increase the end-to-end delay significantly. Hence, a protocol based on variable transmission range for connectivity control would be highly effective in such a dynamic environment. Additionally, because of mobility, a node involved in a communication process may move out of the fixed transmission range of the sender, thus disturbing the communication process. Our transmission range control mechanism will try to protect connectivity between any two nodes involved in a communication process at that instant of time by adjusting their transmission ranges. Thus, the proposed scheme eliminates the need for route maintenance during data communication in ad hoc networks.

I. INTRODUCTION

There has been a growing interest in mobile, multi-hop wireless networks in recent years. These networks are also termed as ad-hoc network [1,2] where the network may or may not be connected with the infrastructure such as internet, but still be available for use by a group of wireless mobile hosts operating without any base-station or any centralized control. However, the successful operation of an ad-hoc network will be disturbed, if a node, participating in a communication between two other nodes, moves out of range in between message transfer.

To achieve this objective, we propose a communication protocol, which will allow a path to be retained during a data communication along that path. Because of mobility, a node involved in the communication process may move out of the fixed transmission range of the sender, thus disturbing the communication process. Our mechanism is based on self-adjusting variable transmission range of mobile hosts, which will not allow this to happen during a communication process. At the same time, variable transmission range of mobile hosts will allow us to control the congestion of control packets in a dynamic setting. It has been observed that a low transmission range will not guarantee proper connectivity among mobile hosts to ensure effective communication. On the other hand, if the transmission range is high, it will ensure connectivity but will increase collision and congestion of control packets, which will increase the end-to-end delay significantly. For a fixed number of nodes uniformly distributed over an operating area, an optimal transmission range can be worked out. But in an ad hoc network environment, the number of nodes as well as the concentration of node in different area of the operating zone varies. Hence, a protocol based on variable transmission range would be highly effective in such a dynamic environment. Additionally, adaptive transmission range will ensure balanced consumption of battery power of the mobile nodes in an ad hoc network environment by controlling the transmission power consumed by the nodes.

The protocol is based on a neighborhood agreement/denial scheme through periodic beacon exchange among the neighboring nodes only. The basic assumption is that, even if a node is within the transmission range of another node, it will not be considered as its neighbor unless both of them agree to be the neighbor of each other. On the other hand, a node can increase its transmission range to include someone as its neighbor. A node will vary its transmission range so that the number of its registered neighbor is six to eight. It has been pointed out earlier [3,4] that if the number of neighbors is six to eight (referred to as Magic Number), it would optimize the throughput by reducing congestion and collision as well as ensuring connectivity of the network. At the same time, by adjusting the transmission range, a node-pair will try to maintain their neighborhood relationship i.e. protect the link between them, if that link is involved in a communication process at that instant of time.

II. A COMMUNICATION PROTOCOL BASED ON SELF-ADJUSTING TRANSMISSION RANGE

The protocol has three components :

- Transmission Range Control Protocol (TRCP) that will help each node to adjust periodically its transmission range in order to maintain the number of its registered neighbors within six to eight ;
- Path Finding Protocol (PFP) which is a demand-driven, source-initiated routing protocol to find a set of paths between any source-destination pair; and,
- Path Evaluation and Data Communication Protocol (PEDCP) which helps the source to evaluate the most stable path among the set of paths and to start the data

communication along that path. Additionally, it sets a flag in each link in that path which helps TRCP to protect those links during data communication.

A. Definitions

Affinity : *Affinity* a_{nm}, associated with a link l_{nm}, is a prediction about the span of life of the link l_{nm} in a particular context [6]. Thus, the stability of connectivity between n and m depends on a_{nm}. To find out the affinity a_{nm} , node m samples the strength of signals received from node n periodically. Since the signal strength of n as perceived by m is a function $f(R_n, d_{nm})$ where R_n is the transmission range of n, and d_{nm} is the current distance between n and m, we can predict the current distance d_{nm} at time t between n and m. If M is the average velocity of the nodes, the average worst-case affinity a_{nm} at time t is $(R_n-d_{nm})/M$, assuming that at time t, the node m has started moving outwards with an average velocity M.

Stability of a Path*:* Given any path p from any node i to another node m as p = (i, j, k, …, l, m), the *stability of path p* [6] will be determined by the lowest-affinity link (since that is the bottleneck for the path) and is defined as: $\min[a_{ij}, a_{jk}, \ldots, a_{lm}]$. In other words, stability of path p between source s and destination d, η^p_{sd}, is given by $\eta^p_{sd} = \min [_{\forall i,j}\, a^p_{ij}]$.

Physical Neighbor : We define a node m as a *physical neighbor of n* if and only if m is within the transmission range of n.

Neighbor (or Logical Neighbor) : We define a node m as a *neighbor of n* and vice versa if and only if both n and m are within the transmission range of each other and there is a neighborhood agreement between n and m. A neighbor is sometimes referred to as logical neighbor. It follows from the definition that if m is a neighbor of n, then n has to be a neighbor of m.

Magic_Number : Assuming uniform distribution, if the average node density per unit area is λ (= N/A) and A_R is the area of coverage of node i for transmission range R ($=\Pi R^2$), then the optimal value of $\lambda.A_R$ which maximizes throughput has been referred to as *Magic Number* [3,4]. It has been shown that the value of $\lambda.A_R$ from six to eight will have a high probability of leading to a connected network as well as reducing congestion and collision in the network.

Magic_Low (M_L): *Magic_Low* is defined as the minimum number of neighbors which each node should have for acceptable connectivity. In our simulation, $M_L = 5$.

Local_Magic_Low (LM_L): It is a locally-maintained variable within each node to indicated the minimum number of neighbors which that node should have. Normally, Local_Magic_Low = Magic_Low.

Magic_High (M_H)*: Magic_High* is defined as the maximum number of neighbors of each node beyond which the congestion and collision in the network becomes unacceptable. In our simulation, $M_H = 8$.

Potential Neighborhood Count (C_{PN})*:* A node i will receive periodic neighborhood acknowledgments from its registered neighbors and will also receive periodic neighborhood requests from other nodes who are willing to be the neighbors of i. At any instant of time, the sum of neighborhood acknowledgments and neighborhood requests at i is defined as *Potential Neighborhood Count* of i. If there is no request, Potential Neighborhood Count is same as the number of neighbors of node i at that instant of time.

B. Overview of Transmission Range Control Protocol

Each node maintains two tables: A Message_table (MT) to accumulate messages received from other nodes; and, a Neighborhood_Table (NT) that contains the updated information of registered neighborhood status. The routing algorithm for data communication uses NT to get the neighborhood information in order to forward data/control packets. Whenever a node receives a message, it appends it in MT. At each periodic interval T, (say, every 100 ms.) the node will process MT, update NT accordingly, adjust its transmission range and sends messages to other nodes.

MT has two components : the first three fields are used to maintain the neighborhood information; the last field indicates whether the link between i and neighbor_id needs to be protected due to data communication. The same link may be used in multiple data communication simultaneously. If this link is selected for a data communication with a communication id, a Start_Data_Comm packet will set that flag and an End_Data_Comm packet will reset it . This will be discussed in detail in the context of Path Evaluation and Data Communication Protocol.

The protocol helps a node to adjust its transmission range periodically and sends the following messages to establish, retain or deny a neighborhood relationships with other nodes:

Neighborhood Establishment : If Potential Neighborhood Count of node i is less than Magic_Low, then node i would like to have more neighbors. It increases its transmission range R in steps of ΔR (20 units in our simulated model) and sends a request (REQ) message to all the nodes within its transmission range. If a receiving node j wishes to respond to this request, it keeps a tag on this request and sends an acknowledgment (ACK) message to i. If i is within the transmission range of j, a neighborhood relationship will be established between i and j.

If i is not within the transmission range of j, i will not receive this ACK from j at that periodic interval. However, if other nodes are there to satisfy the request from i, i will not send any further request and j will delete the request-tag. But, if there is no one to respond to i's request, i will increase its transmission range in steps of ΔR and sends request in the next periodic interval also. Now, if j receives request from i in two consecutive time intervals, j increases its transmission range in steps of ΔR and sends an ACK message to i. If i is still outside the transmission range of j, i will further increase its transmission range in steps of ΔR and send request in the next periodic interval also. On receiving this request, j will also increase its transmission range in steps of ΔR and sends an ACK to i. This will continue until i is satisfied, either by other nodes or by j.

As a receiving node, if the Potential Neighborhood Count of j is less than Magic_Low or within Magic_Band, j would always like to accommodate i's request. However, if the Potential Neighborhood Count is more than Magic_High, j needs to evaluate the request from i before responding. There are two basic policies we are following in this context. First, each node should try to accommodate at least one request; second, at any instant of time, number of ACK messages from a node should not exceed Magic_High.

So, when the Potential Neighborhood Count is more than Magic_High, two situations needs to be considered. First, the number of registered neighbors of j is already equal to Magic_High and i is the only node requesting for a new neighborhood agreement. Based on the first policy, j should try to accommodate the request from i in the manner described above and de-register one existing neighbor (the furthest neighbor that is not involved in a communication process at that instant of time). Second, the number of registered neighbor is less than Magic_High (say, x), but there are multiple requests including the request from i (say, y). Following the second policy, j will send x number of ACK to its registered neighbors, and selects (Magic_High –x) number of requests from y and sends ACK only to those requests. The selection of requests is done based on their closeness from j, as evaluated from the signal strengths of the request and the transmission range of the requesting node.

Neighborhood Retention: If a node i wishes to retain its neighbor j, it sends a ACK message to j. If j is in the periphery of the transmission range of i and if i wishes to retain j in order to protect the link (i,j) during a data communication, it adjusts its transmission range to include j well within its transmission range before sending a ACK message. If a node i has its neighborhood count within Magic_Band and if it finds that it can retain all its neighbors within its transmission range even after reducing its transmission range by ΔR, it will do so.

Neighborhood Denial: If a node i wishes to de-register its neighbor j, it stops sending ACK message to j. If j does not receive ACK from i, it will de-register i as its neighbor.

C. Oscillation, its Detection and Prevention Mechanism

Depending on the network topology, one or more nodes in the system may enter into an oscillatory state where the nodes would acquire a neighbor and loose it in alternate time interval. This oscillation would go on unless there is a change in topology. Following example describes this oscillatory nature of the algorithm:

Let us take three nodes A, B and C such that at time t

- for A , number of neighbors < Magic_Low, transmission range is R_a
- for B number of neighbors = Magic_Low, transmission range is R_b
- for C, number of neighbors = Magic_High , transmission range is R_c
- $R_c, R_a >$ dist(C,A), $R_c, R_b >$ dist(C,B), where dist(x,y) denotes the distance between x and y
- B has a neighborhood relationship with C
- the relative distance of A, B and C is such that for any N_j belonging to neighborhood of C, dist(C,N_j) < dist(C,A), dist(C,B)
- all other existing neighbors of A, B and C have a stable relationship with A, B and C.

By this protocol, at time t the node A will generate a request which will be received and processed by node C. Since C is at Magic_High, it will accommodate A by rejecting its furthest neighbor, which is in this case B. Therefore, A will receive and process an acknowledgment from C at next time interval and stop generating request. On the contrary, B will not receive any acknowledgment from C and it will go to a state with number of neighbors of B = (Magic_Low-1). Consequently, it will generate a request at the same transmission range. Therefore, at next time interval, node C will again receive and process a request from B. C will now accommodate B and reject the furthest neighbor which is in this case A . Therefore B will go to a state with number of neighbors of B = Magic_Low but A will reenter in a state where its number of neighbors < Magic_Low. The node A will regenerate request at the same transmission range which will lead to same sequence of state transitions for A, B and C. The oscillation of A and B as a neighbor of C continues until an event occurs that contradicts any of the conditions stated above.

We suggest a strategy for detecting oscillation at the request-granting node that induces oscillation between two requesting nodes. The request-granting node essentially decides to acknowledge only the closer requesting node (the closeness is derived from the distance of the two requesting nodes from itself) while rejecting the other for a finite period of time.

Each node n maintains two tables to detect and prevent oscillation. The first table is updated by n at each periodic interval and it contains the history of all acknowledgements sent by the node n during last six consecutive periodic intervals. The second tables contains the list of rejected nodes i.e. the request from these nodes will not be acknowledged by n . The second table is cleared, when Neighborhood_Count of node n goes below Magic_high due to mobility of nodes, i.e. the node n becomes ready to accept more requsts.

D. Analysis of TRCP Algorithm

The analysis is based on the assumption that all packets transmitted over an operational link are received correctly and in the proper sequence within a finite time.

Theorem 1. *If two requesting nodes n_1 and n_2 shows oscillatory behavior with a request-granting node n_3, then the worst case time required for the algorithm to converge is ([gap (n_1,n_2)+gap (n_2,n_1)] / ΔR) *T, provided the transmission range of n1 does not reach Max_Trans during this process.*

Here, gap (i, j) is equal to (dist(i, j)-R_i), T is the periodic interval after which a node processes its Message Table, R_i is the current transmission range of node i, dist(i, j) is the current distance between i and j, and, ΔR is the progressive increment in transmission range.

Proof : From the oscillatory behavior, it is evident that both n_1 and n_2 are below the lower boundary condition of stability and n_3 is at higher boundary condition of stability. So, let us assume that n3 rejects n1 as its neighbors. So, n1 will progressively increase its transmission range in steps of ΔR and generate request at each periodic interval T.. If not, n1 will eventually include n2 within its transmission range after a time (gap (n_1,n_2) / ΔR) *T. Now, n2 is in a position to respond to its request. However, since n1 is not within the current transmission range of n2, n2 will now progressively increase its transmission range in steps of ΔR and generate request at each periodic interval T. At the same time, n1 will also goes on increasing its transmission range in steps of ΔR and generating request until it receives an acknowledgement from n2.After a time (gap (n_2,n_1) / ΔR) *T, n1 will receive acknowledgement from n2. Thus, there will be a neighborhood tie-up between n1 and n2 after a time ([gap (n_1,n_2)+gap (n_2,n_1)] / ΔR) *T.

During this process, if any other nodes responds to the request of n1, the neighborhood requirement of n1 will be satisfied and the process terminates. Also, during this process, if the transmission range of n1 reaches Max_Trans, n1 will reduce its neighborhood requirement by reducing Local_Magic_Low by 1, reduce its transmission range to a value Min_Trans, and start the process again with a modified Local_Magic_Low.

Theorm 2 *The algorithm terminates in finite time after the last topological change happened .*

Proof : On termination of TRCP algorithm, each node in the network will have its neighborhood_count within Magic_Band. It implies that no new message are being transmitted or processed until there is a change in topology which will disturb the existing neighbourhood relationship. If the algorithm does not terminate in finite time, it implies that there is at least one node whose neighborhood_count is less than its Local_Magic_Low. We will prove that this can never happen.

If a node n has a neighborhood_count less than its Local_Magic_Low, it will progressively increase its transmission range in steps of ΔR and send request. This will continue until its transmission range reaches Max_Trans. If n is a remote node and there is no other node within its Max_Trans, n will eventually reduce its Local_Magic_Low to zero and the algorithm terminates. If there is node m within the Max_Trans of node n and node m has a neighborhood count within Magic_Band, then node m will accommodate n as its neighbor. If m has a neighborhood count at Magic_high and n does not induce any oscillation, then also node m will accommodate n as its neighbor. If m has a neighborhood count at Magic_high and n induces any oscillation, then node m may reject n as its neighbor. However, from theorem 1, that will also lead to the convergence of the algorithm. Same argument applies if there are more than one node within the transmission range of n. In all the cases, the algorithm terminates with n having a neighborhood count between Local_Magic_Low and Magic_High.

This theorem additionally implies that no node will suffer from starvation, if its distance from at least one node in the network is less than Max_Trans and it is not creating an oscillatory relationship with that node.

E. Path Finding Protocol (PFP)

In this scheme, a source initiates a route discovery request when it needs to send data to a destination [2]. The source broadcasts a route request packet. All the nodes within the transmission range of the source will receive this. Each route request packet contains source id, destination id, a request id, a route record to accumulate the sequence of hops through which the request is propagated during the route discovery, and a count max-hop which is decrement at each hop as it propagates. When max-hop=0, the search process terminates. The count max-hop is 4 in our case[5].

F. Path Evaluation and Data Communication Protocol

By adjusting the transmission range, a node-pair will try to maintain their neighborhood relationship i.e. protect the link between them, if that link is involved in a communication process at that instant of time. In the worst case, if two nodes are moving away from each other, both of them have to progressively increase their transmission range in order to protect the link between them. In order to protect a path during data communication, all of the node-pairs in the path have to protect their links by adjusting their transmission range

Let us assume that a source s wants to send NUM number of packets to a destination d. The source s initiates a route discovery request and it waits for the route reply until time-out. After time-out, the source evaluates all the paths in order to find out the "best" path for data communication. The best path is a path where the path protection mechanism during data communication will consume the least battery power during the adjustments of their transmission range.

Suppose that source s and destination d are connected by two intermediate nodes j and k. The current transmission range of s, j, k and d are R_s, R_j, R_k and R_d and the current distances between nodes are d_{sj}, d_{jk} and d_{kd}. The following steps are performed to evaluate a path :

- Compute the stability required : If NUM number of packets need to be communicated from s to d, then the stability η = (number of hops * NUM *t_p) / f , where t_p is the average hop-delay per packet and f is the correction factor to take care of congestion (typically, f=0.7 to 0.9, depending on traffic volume).
- Compute the affinity required for each link ; affinity required for each link should be at least equal to η. So, the final transmission range of each node should be sufficient to deliver this affinity value.
- Compute the final transmission range required by each node to achieve this affinity :
  If M is the average velocity, then
  $NewR_s = \eta*M + d_{sj}$
  $NewR_j = \max((\eta*M + d_{sj}), (\eta*M + d_{jk}))$
  $NewR_k = \max((\eta*M + d_{jk}), (\eta*M + d_{kd}))$
  $NewR_d = \eta*M + d_{kd}$
- If any one of ($NewR_s$, $NewR_j$, $NewR_k$, $NewR_d$) > Maximum transmission range Max_Trans, then reject the path and go to the next path.

- For all selected paths, Compute the sum of deviations from Max_Trans and average :

δR_p = ((Max_Trans – $NewR_s$) + (Max_Trans – $NewR_j$) + (Max_Trans – $NewR_k$) +(Max_Trans –$NewR_d$))/4

The best path is the path with max(δR_p).

The data communication from source to destination starts with a Start-Data-Comm packet. The purpose of this packet is to set protect flags in the NTs of intermediate nodes including s and d. On receiving the acknowledgment from designation, the source will start the data communication. Due to mobility of nodes, it may so happen that the Start-Data-Comm packet will not reach the destination and a path-reservation-failure occurs. In that case, no acknowledgment will come back to source via the intermediate nodes in the path. Thus, if no acknowledgement is received within a specified time-out period, the protect flags in NTs of all the nodes in that path for that communication-id will be reset and the source starts the same procedure with the second best path. On completion of the data communication, an End-Data-Comm packet will reset the protect flags, thus releasing the path.

III. PERFORMANCE EVALUATION

A. *Simulation Set up*

The proposed system is evaluated on a simulated environment [5,6] under a variety of conditions. It has been observed that, if the transmission range is adjusted to keep the average number of neighbors of a node between 6 to 8, the number of control packets generated would be uniform. Hence, the congestion generated due to control packet propagation would also be uniform. Consequently, the average end-to-end delay is also does not get affected by control packets.

Therefore, Service_Efficiency, which is defined as the ratio of the average number of data communication events successful per minute and the average number of route request generated per minute, would improve drastically in the proposed scheme as compared to any other routing scheme proposed in the context of ad hoc network. This is shown in Table 1.

IV. CONCLUSION

The preliminary results indicate significant advantages of the proposed protocol over existing routing protocols proposed in the context of ad-hoc networks. With the advent of radios with more sophisticated controls on transmission parameters, self-adjusting transmission range control is going to be a viable solution in near future.

REFERENCES

[1] E. M. Royer and C-K Toh, “A Review of Current Routing Protocols for Ad hoc Wireless Networks”, IEEE Personal Communication, April 1999, pp. 46-55.

[2] D. B. Johnson and D. Maltz, Dynamic source routing in ad hoc wireless networks, T. Imielinski and H. Korth, eds., *Mobile computing*, Kluwer Academic Publ. 1996.

[3] H.Takagi and L.Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Trans. Commun. vol COM-32, pp246-257, Mar.1984.

[4] Y.C.Cheng and T.G.Robertazzi, Critical connectivity phenomena in multihop radio network, IEEE Trans. Commun. , 37(1989), pp 770-777.

[5] Krishna Paul, Romit Roychoudhuri and S. Bandyopadhyay, “Survivability Analysis of Ad Hoc Wireless Network Architecture”, Proc. of the Second International Workshop on Mobile and Wireless Communication Networks, Paris, France, May 2000. Published in Lecture Notes in Computer Science LNCS 1818, Springer Verlag.

[6] Krishna Paul, S. Bandyopadhyay, D. Saha and A. Mukherjee, “Evaluating the Performance of Affinity-Based Routing Schemes among Mobile Hosts in Ad-Hoc Wireless Networks”, Accepted in the 7th International Conference on Telecommunication Systems, Modelling and Analysis, Nashville, USA, March 1999.

No. of Nodes	**Shortest_Path Algorithm with Fixed Transmission Range (R)**					**Variable_Trans_Range Algorithm with Adaptive Transmission Range (R_{adpt})**			
	R	**Num. of Comm. Initiated**	**Num. of Comm. Successful**	**Service -Eff.**		**R_{adpt}**	**Num. of Comm. Initiated**	**Num. of Comm. Successful**	**Service-Eff.**
20	**300**	**10**	**7**	**70%**		**323.7**	**10**	**10**	**100%**
30	**300**	**10**	**5**	**50%**		**295.7**	**10**	**9**	**90%**
40	**300**	**10**	**5**	**50%**		**269.9**	**10**	**10**	**100%**

Table 1. A comparison at d*ata volume = 1000 packets, mobility = 20 m/sec* and frequency of communication = *10 per min.*