Improving Quality-of-Service in Ad hoc Wireless Networks with Adaptive Multi-path Routing

Sajal K. Das*, A. Mukherjee+, S. Bandyopadhyay+, Krishna Paul#, D. Saha^

*Department of Computer Science and Engineering, The University of Texas at Arlington,
Arlington, TX 76019-0015, USA. E-mail: das@cse.uta.edu
+Saltlake Technology Center, PricewaterhouseCoopers Ltd., Saltlake, Calcutta 700 091, India.
#Cognizant Technology Solutions, Sector V, Saltlake, Calcutta 700 091, India.
^Department of Computer Science, Jadavpur University, Calcutta 700 032, India.

Abstract. **The objective of this paper is to propose a mechanism for adaptive computation of multiple paths to transmit a large volume of data packets from a source s to a destination d in ad hoc wireless networks. We consider two aspects in this framework. The first aspect is to perform preemptive route rediscoveries before the occurrence of route errors while transmitting a large volume of data from s to d. Consequently, this helps find out dynamically a series of multiple paths in temporal domain to complete the data transfer. The second aspect is to select multiple paths in spatial domain for data transfer at any instant of time and to distribute the data packets in sequential blocks over those paths in order to reduce congestion and end-to-end delay. The performance of this approach has been evaluated to show the improvement in the quality of service. It has been observed that the mechanism allows any source to transmit a large volume of data to a destination without degradation of performance due to route-errors. Additionally, it would help reduce significantly the end-to-end delay and the number of route-rediscoveries needed in this process.**

I. INTRODUCTION

There has been a growing interest in ad hoc networks in recent years [1,2,3,4,5]. An important problem associated with routing in ad hoc networks is to employ methods that will ensure quality of service (QoS). The successful operation of an ad-hoc network will be disturbed, if an intermediate node, supporting a communication between a source-destination pair, moves out of range during data transfer. This interruption in communication results in a subsequent route re-discovery between that source-destination pair or invoking some path-maintenance algorithm that eventually increases the end-to-end delay.

Moreover, the routing schemes proposed so far in the context of ad hoc networks employ single-path routing which might not ensure optimal end-to-end delay. However, once a set of paths between s to d is discovered, in some cases, it is possible to improve end-to-end delay by splitting the volume of data into different blocks and sending them via selected multiple paths from s to d, which would eventually reduce congestion and end-to-end delay [6]. Utilization of multiple paths to provide improved performance as compared to a single path communication has been explored in the past in the context of wired network [9,10,11]. However, the use of multiple paths has not been explored in the domain of ad hoc wireless networks.

In this paper, we propose a mechanism for adaptive computation of multiple paths in temporal and spatial domain to transmit large volume of data from s to d in ad hoc wireless networks. We consider two aspects in this framework. The first aspect is to perform preemptive route re-discoveries before the occurrence of route errors while transmitting a large volume of data from s to d. This helps find out dynamically a series of multiple paths in the temporal domain to complete the data transfer. The second aspect is to select multiple paths in the spatial domain for data transfer at any instant of time and to distribute the data packets in sequential blocks over those paths in order to reduce congestion and end-to-end delay. A notion of link stability and path stability has been defined and a unified mechanism is proposed to address these two aspects that relies on evaluating a path based on link stability and path stability. The solution method uses Lagrangean relaxation and subgradient heuristics [7,8] to find out optimally the paths and data distribution into those paths both in temporal and spatial domains. The performance of this approach has been evaluated on a simulation environment. It has been observed that the use of temporal multi-paths allows any source to transmit a large volume of data to a destination without degradation of performance due to route-errors. Additionally, the use of spatial multi-paths would help reduce significantly the end-to-end delay and the number of route-rediscoveries needed in this process.

The organization of this paper is as follows. Section II describes a stability-based framework for QoS routing. Section III illustrates an adaptive mechanism for multipath routing using a solution method based on Lagrangean relaxation and subgradient heuristics. Section IV explains the simulation results followed by a conclusion in section V.

II. A STABILITY-BASED FRAMEWORK FOR QOS ROUTING

Before proceeding further, let us introduce the following notations used to describe the framework:

A. *Notations*

N: set of nodes

0-7803-6451-1/00/$10.00 © 2000 IEEE

L: set of directed links
s,d: source and destination , $s,d \in N$
l_{mn}: link from node m to node n, where $m,n \in N$ and $l_{mn} \in L$
P: set of a set of stable paths, i.e., $\{P_1, P_2, \ldots\} = \{P_i\}$, $i \in [1,\infty)$
P_i: ith set of stable paths, $P_i \in P$, $i \in [1,\infty)$
p_j: binary indication variable for jth path in the set P_i, where
each set $P_i \in P$ and $i \in [1,\infty)$, $j \in [1, |P_i|]$, where
$p_j = 1$, when path p_j is selected for the set P_i
$= 0$ otherwise
D: set of a set of data volume in packets
$=\{D_1, D_2, \ldots\} = \{D_i\}$, $i \in [1,\infty)$
D_i: ith set of data volume in packets, $D_i \in D$, $i \in [1,\infty)$
Δ_j: data distribution for the jth path in the set P_i
and $\sum_{j \in P_i} \Delta_j = D_i$ for $i \in [1,\infty)$
R: transmission range (assumed to be equal for all nodes)
M: average velocity of the nodes
a_{mn}: affinity of link l_{mn}, $l_{mn} \in L$
B: link bandwidth in packets/msec
H_j: number of hops traversed by j-th path in the set P_i
τ_j: average delay per hop per packet for j-th path in the set P_i
d_j: route discovery time for j-th path in the set P_i
q_j: average queuing delay per packet per node for j-th path
T_j: average path delay per packet for j-th path in the set P_i
and $T_j = H_j * \tau_j$
S_j: stability for j-th path in the set P_i
$d_{stab(i)}$: route discovery time for the most stable path in set P_i
$H_{stab(i)}$: number of hops traversed by most stable path in set P_i
$\tau_{stab(i)}$: average delay per hop per packet for most stable path in the set P_i
$T_{timeout}$: route-request-time-out for any set P_i
H_{avg}: average number of hops traversed by any j-th path in any set P_i

B. *System Description*

The network is modeled as a graph $G = (N, L)$ where N is a finite set of nodes and L is a finite set of directed links. Each node $n \in N$ is having a unique node identifier. Two nodes n and m are connected by two unidirectional links $l_{nm} \in L$ and $l_{mn} \in L$ such that n can send message to m via l_{nm} and m can send message to n via l_{mn}. However, in this study, we have assumed $l_{nm} = l_{mn}$ for simplicity.

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. 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. Given any path p from any node i to another node m as $p = (i, j, k, \ldots, l, m)$, the *stability of path p* 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 $\eta^p_{sd} = \min\,[_{\forall i,j}\, a^p_{ij}]$.

Even if we evaluate a path to find out its stability, it may not be sufficiently stable to carry a large volume of data in a highly dynamic environment of ad hoc wireless networks. In this situation, communication can not be initiated for a large volume of data because of low path stability. Even if the communication is initiated, some form of route maintenance scheme has to be employed to repair a path or to find out an alternative path in case a route error occurs. However, this interruption in service and its resumption after route re-discovery will eventually degrade the QoS. Instead, if it is possible to predict the life of a path from s to d and accordingly preempt the route re-discovery process and discover a new path from s to d before the existing path breaks, it is possible to provide uninterrupted data communication service from s to d.

Moreover, once a set of path between s to d is discovered, in some cases, it is possible to improve end-to-end delay by splitting the volume of data into different blocks and to send it via selected multiple paths from s to d which would eventually reduce congestion and end-to-end delay. Of course, depending on the topology, we need to decide dynamically whether multi-path is a preferred mode or not.

III. AN ADAPTIVE MECHANISM FOR MULTI-PATH ROUTING

This section describes an adaptive mechanism for multi-path routing using a solution method based on Lagrangean relaxation and subgradient heuristics.

A. *Problem Formulation*

We consider set N of n nodes which are interconnected in a wireless mobile environment. Now we consider that the volume D of data packets needs to be routed from a source to destination (s-d). If D is very large, then it can not always be possible to send data in a single path in the mobile environment. This is because of the fact that the stability of the path is not so high to keep the connectivity during the routing of the entire data volume. So, the total data volume D is divided into comparatively smaller data volume packets D_1, D_2,.... called as temporal data sets. The number of major-division sets, i.e., $D_1, D_2, \ldots$, depends on the stability of paths, obtained dynamically at different time interval. Clearly, temporal data sets $\{D_1, D_2, \ldots\}$ constitute the set D. Each temporal data set, say ith set D_i, corresponds to say ith temporal stable path set P_i, in which data packets are routed from s to d. It is also clear that corresponding temporal stable path sets $\{P_1, P_2, \ldots\}$ form the set P. That is, P and D are sets of temporal sets $\{P_1, P_2, \ldots\}$ and $\{D_1, D_2, \ldots\}$ respectively.

Each temporal stable path set, say ith set P_i, has a number of stable j-paths ($j=1,2..$). The jth stable path is selected when the value of p_j is 1 and the maximum number of paths selected in the set P_i is equivalent to $|P_i|$. We call these j-paths (j=1,2..) as spatial paths in the temporal path set P_i. On these j-spatial paths, the data D_i is distributed further in Λ_j (j=1,2...). We also call them as spatial data volume Λ_j (j=1,2...) for the set D_i and the sum of all Λ_j's is equivalent to the data volume of the temporal set D_i. The S_j, d_j and H_j for the path p_j are self-explained terms.

In order to perform a preemptive route rediscovery before the occurrence of route error, the source should be able to initiate the route rediscovery in such a manner that next set of paths are available before the completion of current data volume. Let us assume that $T_{timeout}$ is the route-request-time-out, i.e. the maximum time interval allocated from generating a route-request from a source and getting route replies back to source for any set P_i and H_{avg} is the average number of hops traversed by any jth path in any set P_i. For any set P_i with data volume to be communicated D_i, the route-rediscovery for the next set P_{i+1} would be initiated after transmitting (D_i – $T_{timeout}/(H_{avg}{*}\tau_{avg})$ amount of data packets which will ensure that the next set P_{i+1} would be available to source just before completion of transmitting D_i volume of data packets.

On the basis of the notations and terminology explained above, we describe the problem as follows.

Problem Definition: Minimize the average delay of a wireless mobile ad-hoc network by sending data packets dividing them into a set of several temporal paths for each source-destination (s-d) pair. Again each of these temporal paths is divided into a number of spatial paths through which smaller data blocks are sent. Several stabilized paths with route discovery time and a number of hops traversed into these paths have been taken as input to the problem.

Formally this optimization problem is defined as:

Minimize :

$Z = (\sum_{i\in[1,\infty)} \sum_{j\in P_i} (d_j + \Delta_j H_j \tau_j p_j))^{\alpha}$ for each s-d pair (1)

where α is the degree of the complexity of the optimization problem.

Subject to:

$\forall_{i\in[1,\infty)} [\sum_{j\in P_i} (d_j + \Delta_j H_j \tau_j p_j) < (d_{stab(i)} + |D_i| H_{stab(i)} \tau_{stab(i)})]$ (2)

$\forall_{i\in[1,\infty)} [\forall_{j\in P_i} ((d_j + \Delta_j H_j \tau_j p_j) < S_j - \sum_{k=1}^{j-1} (d_k + \Delta_k H_k \tau_k p_k))]$ (3)

$\forall_{i\in[1,\infty)} [\sum_{j\in P_i} \Delta_j = D_i]$ and $\sum_{i\in[1,\infty)} D_i = \mathbf{D}$ (4)

$\forall_{i\in[1,\infty)} [\forall_{j\in P_i} (0 \le p_j \le 1)$ and $\sum_{j\in P_i} \lceil p_j \rceil \le |P_i|\,]$ (5)

The objective function (1) will determine the average network delay for each source-destination pair. The constraint (2) guarantees the data distribution for each s-d pair in a set of paths must be less than that over a single, most stable path. The constraint (3) assures that the data distribution in multiple paths must reach the destination within the life of those paths. The constraint equation (4) mentions that the total data packets for each source-destination must be distributed in a set of paths. The constraint (5) indicates that when the value of the path p_j must lie between 0 and 1, the path will be selected for the set P_i, and the number of paths selected must not exceed the cardinality of the set P_i.

B. *Path Finding Algorithm*

In this scheme, a source initiates a route discovery request when it needs to send data to a destination. The source broadcast a route request packet to all neighboring nodes. Each route request packet contains source id, destination id, a request id with a locally maintained time-stamp, a route record to accumulate the sequence of hops through which the request is propagated during the route discovery, and a count max_hop (max_hop =4 is taken as an initial value in the simulation) which is decrement at each hop as it propagates. When max_hop =0, the search process terminates. The count max_hop limits the number of intermediate nodes in a path.

When any node receives a route request packet, it decrements max_hop by 1 and performs the following steps:

1. If the node is the destination node, a route reply packet is returned to the source along the selected route, as given in the route record that now contains the complete path information.
2. Otherwise, if max_hop =0, discard the route request.
3. Otherwise, if this node id is already listed in the route record in the request, discard the route request packet (to avoid looping).
4. Otherwise, append the node id to the route record in the route request packet and re-broadcast the request.

When any node receives a route reply packet, it performs the following steps:

1. If the node is the source node, it records the path to destination along with its time of arrival from locally maintained time. Thus, the time-delay between route request and route reply for a path is determined. This is the time required for the route discovery (d_j) between s and d. This is an indicator of the delay caused due to traffic congestion, packet transmission time and number of hops in the path.
2. If it is an intermediate node, it appends the value of affinity and propagates the packet to the next node listed in the route record to reach the source node.

C. *Multi-path Algorithm for Data Communication:*

The multi-path design algorithm is described as follows:

```
step I: initialize sets $\{D_i\}=\Phi$ and $\{P_i\}=\Phi$ where $i\in[1,\infty)$
step II: while $\sum_{i\in[1,\infty)} D_i \le \mathbf{D}$
step III: call the path-finding algorithm that gives a set of stable paths with route-discovery-time and number of hop-counts for those paths. This is the set of input variables to the optimization problem
step IV: call optimization problem Z
step V: the paths $p_j$ (j=1,2,...) are selected for the set $P_i$
step VI: $\Delta_j$'s are determined for all paths $p_j$ (j=1,2,...) for the set $P_i$ and $\sum_{j\in P_i} \Delta_j = D_i$ for $i\in[1,\infty)$
step VII: if $\sum_{i\in[1,\infty)} D_i = \mathbf{D}$ then go to step IX
step VIII: go to step III after transmitting
            $(|D_i| - T_{timeout}/(H_{avg} * \tau_{avg})$ packets
step IX: stop
```

D. *An Example*

The multi-path algorithm can be explained by a simple example. Let us assume that a total data volume of 5000 packets needs to be routed from a source (s) to a destination (d). We call a path-finding algorithm that finds initially a set of stable paths with route discovery time and number of hop-counts of those paths. This path-finding algorithm finds five paths having route discovery time 68 ms, 138 ms, 201ms, 262 ms and 273 ms respectively. These paths are assumed to be stable for 1122 ms, 1682 ms, 3216 ms, 4228 ms and 5011ms and have 2, 3, 4, 4 and 4 hops respectively. The optimization algorithm selects three appropriate paths. The path set $\boldsymbol{P}_1 = \{p_1, p_2 \text{ and } p_5\}$ is selected to route data packets from source-to-destination. The set of data packets $\boldsymbol{D}_1 = \{527\ (\Lambda_1),\ 140\ (\Lambda_2) \text{ and } 764\ (\Lambda_5)\}$ is distributed into paths p_1, p_2 and p_5 respectively by solving this optimization algorithm. That is, the total 1431 packets of data have routed in the 1st iteration. Now $\mathbf{D}$- $\boldsymbol{D}_1$ $(=\sum_i \Lambda_i)$ i.e., (5000-1431) will be distributed in the next few iterations. The path-finding algorithm is again called to find out a set of paths with route discovery time and number of hops traversed in those paths for the next iteration after transmitting (1431-300/2) i.e. 1281 packets where $|\boldsymbol{D}_1| = 1431$, $T_{timeout} = 300$ms, $\tau_{avg} = 1$ msec. and $H_{avg} = 2$.

The multi-path algorithm distributes the remaining amount of data packets from source to destination among the next few iterations in the same way described above. The optimization problem in multi-path algorithm has been solved by Lagrangean relaxation and sugradient heuristics [12].

IV. SIMULATION RESULTS

The proposed system is evaluated on a simulated environment under a variety of conditions [13]. The environment is assumed to be a closed area of (1000x1000) square meters in which mobile nodes are distributed randomly. We ran simulations for 10, 20, 30 and 40 mobile hosts, operating at transmission ranges varying from 150 meters to 400 meters. The bandwidth for transmitting data is assumed to be 1000 packets/sec. The packet size is dependent on the actual bandwidth of the system.

In order to study the delay, throughput and other time-related parameters, every simulated action is associated with a simulated clock. The clock period (time-tick) is assumed to be one millisecond (simulated). For example, if the bandwidth is assumed to be 1000 packets per second and the volume of data to be transmitted from one node to its neighbor is 100 packets, it will be assumed that 100 time-ticks (100 millisecond) would be required to complete the task. The size of both control and data packets are same and one packet per time-tick will be transmitted from a source to its neighbors. The speed of movement of individual node ranges from 5 m/sec to 20 m/sec. Each node starts from a home location, selects a random location as its destination and moves with a uniform, predetermined velocity towards the destination. Once it reaches the destination, it waits there for a pre-specified amount of time, selects randomly another location and moves towards that. However, in the present study, we have assumed zero waiting time to analyze worst-case scenario and a uniform node velocity of 10 m/sec.

The success of the scheme relies on the fact that nodes s and d are always connected through some intermediate nodes in spite of mobility. In other words, the intermediate nodes through which s-d are connected may change with time, but s-d should remain connected. In Figure 4, we have shown the stability of most stable path between two arbitrary nodes 14 and 5, sampled at each 5-second interval of time. At every 5 seconds, a route discovery process is initiated from node 14 and the paths obtained after route-request–time-out (300 msec) are evaluated to find out the most stable path. As shown in the figure, it has been observed that no single path is stable throughout the span of 30 sec. However, we are getting a sustained stability between node 14 and 5 through different intermediate nodes. This establishes the viability of our scheme. In other words, if we can perform preemptive route re-discoveries before the occurrence of route errors while transmitting large volume of data from s to d, it is possible to find out a series of multiple paths in temporal domain to complete a large volume of data transfer.

Table 1 shows an example case to illustrate the advantage of using temporal multi-path only, disregarding the spatial multi-path for the time being. So, the set $\boldsymbol{P}_t$ in this case consists of only one path which is the most stable path. The total data volume to be communicated is 10000 packets from source (node no.24) to destination (node no.5) in a 30-node system with an average mobility of 10m/sec. No single path is found to be sufficiently stable to complete this large volume of data transfer. Thus, the source 24 needs to perform preemptive route discovery six times at different time intervals to find out dynamically a series of multiple paths in temporal domain to complete the data transfer. Each route given in the table is the most stable route at that instant of time.

Table 2 shows the same example using both spatial and temporal multi-paths. It shows significant improvement over the first scheme that uses temporal multi-path alone. First, the number of route discovery required to complete the data transfer process has been reduced to four (from six in earlier case) which implies creation of less congestion due to control packet propagation. Second, the time required to complete the data transfer is 21114 msec. which is significantly less than that without spatial multi-path (30283 msec).Thus, Table 2 illustrates the efficacy of the proposed scheme in reducing end-to-end delay while transmitting large volume of data.

V. CONCLUSION

In this paper, we have introduced a notion of temporal and spatial multi-path routing in ad hoc wireless networks and described an adaptive framework to evaluate the suitability of using spatially multiple paths with an objective to minimize end-to-end delay. Assuming the average per-hop delay per packet (τ_i) to be 1 msec, the gain is almost 30% for 10000 packets. Currently, we are working on simulating the multi-path algorithm under different load situations to see the effect on end-to-end delay.

REFERENCES

[1] 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.

[2] S. Corson, J. Macker and S. Batsell, Architectural considerations for mobile mesh networking, Internet Draft RFC Version 2, May 1996.

[3] Z.J.Haas, A new routing protocol for the reconfigurable wireless networks, ICUPC'97, San Diego, CA, Oct. 1997.

[4] C-K Toh, A novel distributed routing protocol to support ad-hoc mobile computing, IEEE International Phoenix Conference on Computer & Communications (IPCCC'96).

[5] R. Dube, C.D. Rais, K. Wang and S.K. Tripathi, Signal stability based adaptive routing for ad hoc mobile networks, Technical Report CS-TR-3646, UMIACS-TR-96-34, University of Maryland, August, 1996.

[6] N.S.V.Rao and S.G. Batsell, QoS Routing via Multiple Paths Using Bandwidth Reservation, in Proc. of the IEEE INFOCOM98, 1998.

[7] M.L. Fisher, "An applications oriented guide to Langrangean relaxation", Interfaces, Vol 15, pp 10-21. 1985.

[8] H. Held, P. Wolfe, and H.P. Crowder, "Validation of subgradient optimization", Math. Program, vol 5, pp 62-68, 1974.

[9] S. Bahk and W. El-Zarki, Dynamic Multi-path Routing and how it Compares with other Dynamic Routing Algorithms for High Speed Wide-area Networks, in Proc. of the ACM SIGCOM, 1992

[10] H. Chu and K. Nahstedt, Dynamic Multi-path Communication for Video Traffic, Proc. Hawian Int. Conf. On System Science, 1997.

[11] S. Murthy and J.J. Garcia-Luna-Aceves. Congestion-oriented Shortest Multi-path Routing, in Proc. of the IEEE INFOCOM96, 1996.

[12] D. Saha and A. Mukherjee, "On the multidiversity gateway location problem for a multilevel highspeed network", Computer Communication, vol 20, pp576-87, 1997.

[13] K. Paul et. al.,"A Simulation Technique for Evaluating a Distributed Routing Scheme among Mobile Hosts in Ad Hoc Wireless Networks", Proc. ASTC'99, San Diego, California, April 1999.

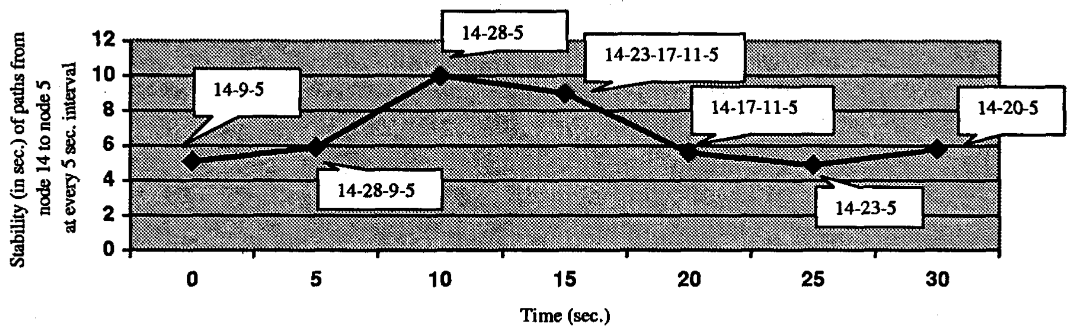

Fig 1. Stability of maximum stable path between node 14 and 5, sampled at each 5-second interval of time (the path is given in the boxes).

TABLE I

TOTAL TIME REQUIRED FOR SENDING 10000 DATA PACKETS USING TEMPORAL MULTI-PATHS ONLY.

D	P_i $\{d_j, S_j, p_j\}$	D_i	Time (in msec.) required for sending D_i packets of data (assuming $\tau_i = 1$ msec/packet)
10000 packets	{249, 11922, 24-31-16-5}	3891	11673
	{251, 4388, 24-32-39-5}	1379	4137
	{238, 2845, 24-25-5}	1303	2606
	{272, 6437, 24-31-37-33-5}	1541	6164
	{297, 2958, 24-11-9-5}	887	2661
	{247, 3358, 24-10-9-5}	999	2997
	TOTAL	**10000 packets**	**30238 msec.**

TABLE II

IMPROVEMENT IN TOTAL TIME REQUIRED FOR SENDING 10000 DATA PACKETS USING BOTH SPATIAL AND TEMPORAL MULTI-PATHS

D	$P_i\{d_j, S_j, p_j\}$			Λ_j			D_i	Time reqd. for sending D_i pkts data ($\tau_i = 1$ ms/pkt)
	p_1	p_2	p_3	Λ_1	Λ_2	Λ_3		
10000 pkts	{2, 3096, 24-5}	{56, 6864, 24-39-5}	249, 11922, 24-31-16-5}	3095	1058	2154	6037	11673
	{117, 2409, 24-25-5}	{251, 4388, 24-32-39-5}	x	1145	615	x	1760	4135
	{238, 2845, 24-25-5}	x	x	1303	x	x	1303	2606
	{289, 3200, 24-26-9-5}	x	x	900	x	x	900	2700
	TOTAL						**10000 pkts**	**21114 msec.**