Local Restoration for Bandwidth Guaranteed Connections in Mobile Optical Backbone Networks

Fangting Sun Mark Shayman
ftsun@glue.umd.edu shayman@glue.umd.edu
Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742

***Abstract*— We consider a mobile backbone network with free space optical point-to-point links. Each backbone node moves randomly but with low mobility, and future movement is not predictable. Requests for aggregate bandwidth between pairs of backbone nodes arrive one-by-one, and a bandwidth guaranteed connection is established if there are sufficient network resources (link bandwidth and node transceivers); otherwise, the request is rejected. Each request has a finite duration after which the bandwidth guaranteed connection is released. The connection may break due to the movement of the backbone nodes, thus necessitating a mechanism for connection restoration. We introduce a new type of mobile node, referred to as *controllable restoration agent* that may be used to restore broken connections. An online algorithm is developed to determine where to position the agents to provide maximum protection, given the locations of the nodes and the set of existing connections. It is shown that by using the algorithm to strategically place a limited number of agents, a significant improvement in network performance can be achieved.**

I. INTRODUCTION

Mobile ad hoc network (MANET) is a group of mobile nodes without centralized administration or fixed network infrastructure, in which the mobile nodes can communicate with other nodes directly or through cooperatively forwarding packets for each other. Since MANETs can be easily deployed and reconfigured, they become more and more attractive in a wide range of military and commercial applications. However, it has been shown that a flat ad hoc network has poor scalability [1]. To build a large-scale MANET, a promising solution is to organize nodes in a hierarchical way [2], [3]. That is, some nodes are selected to form a higher level network called the backbone network, in which the backbone nodes can establish links among themselves. By recursively creating backbone networks, a MANET with multi-level hierarchies can be created.

In this paper, we focus on the optical backbone network in a MANET where each backbone node can build point-to-point free space optical (FSO) links with other backbone nodes. However, similar results would be expected for highly directional radio frequency (RF) links. Since in a MANET nodes with low mobility are more eligible to act as backbone nodes, in this paper we assume that the backbone nodes have limited mobility. Consequently, it is reasonable to set up bandwidth guaranteed connections (e.g., MPLS label switched paths) to carry aggregate traffic, which has the advantage that QoS provision can be supported on a scalable basis.

This research was partially supported by AFOSR under grant F496200210217.

Due to the connection-oriented nature, bandwidth guaranteed connections make networks more vulnerable to path failures. And here we focus on the path failures due to nodes movement. In the literature, there has been considerable work on protecting and restoring bandwidth guaranteed connections in wireline networks, and most of the focus is on reserving two disjoint paths: an active path and a backup path together, either end-to-end [4], [5] or locally [6], [7], where the backup path is used for restoring the connectivity when the active path fails. However, in mobile ad hoc networks, when the active path breaks, the backup path may also become broken with high probability due to node mobility, hence, any pre-computed backup path is not appropriate. Furthermore, reserving two paths is not bandwidth-efficient.

Another way to alleviate the effects of connection failure is to repair the broken connection upon path failure, either locally or end-to-end. The disadvantage is that extra delay is introduced. In this paper, a broken connection that cannot be restored in real time is called an *interruption*. For bandwidth guaranteed connections, an important criterion of QoS provision is to minimize the number of interruptions. There are two ways to restore the bandwidth guaranteed connection caused by path failures: the end-to-end restoration, in which a new path is discovered and set up between the ingress node and the egress node, and the local restoration, in which a new sub-route is discovered and set up to connect the nodes at the sides of broken links. In general, both methods need to introduce extra delay to discover and set up the new path upon path breakage. The end-to-end restoration usually introduces more delay than the local restoration, while the local restoration may generate a sub-optimal path.

In this paper, we introduce a new type of mobile node, *controllable restoration agent*, to restore the broken connections in real time, and correspondingly reduce the number of interruptions. The agents have the same transmission capability as the general backbone nodes. The difference lies in two aspects. First, the agents are not used to take traffic under normal circumstances, but are only used to temporarily reroute traffic upon path failure. Second, the movement of the agents is controllable, and they can be adaptively relocated based on the network status.

The following scenario is considered in this paper. There is a set of general mobile backbone nodes and agents, which are geographically distributed in an area. The backbone nodes may be the clusterheads, and they will move when their clusters relocate. Then the backbone nodes move randomly inside the area with low mobility. Each backbone node may collect the traffic demands within its cluster and issue the requests for aggregate bandwidth to other backbone nodes, and the requests arrive in a random way. For each request, if sufficient resources are available, a bandwidth guaranteed connection is established between the source and destination; otherwise, the request is rejected. Since an established connection may fail due to path breakage, restoration mechanisms should be applied to restore the broken connection. In this paper, the following restoration scheme is used: When a link becomes broken, we first check whether there exists an agent which can temporarily take the traffic passing through the broken link. If this broken path can be repaired by agents, then the traffic on the broken link can be immediately rerouted through those agents, and no interruption happens; otherwise, an end-to-end path should be discovered and a new bandwidth guaranteed connection is to be established, which causes an interruption.

In order to reduce the number of interruptions, the mobile agents should be allocated in an optimal way, and be periodically relocated based on the dynamically changed network status. In this paper we define a weight map that gives a measure of the potential gain by placing an agent at a particular location. Based on the current network status and traffic pattern, the weight is calculated using the proposed distributed algorithm. When we relocate an agent, we should put it in the position corresponding to the highest potential gain in the weight map so that it can provide maximum protection to the network. It is shown that by using the algorithm to strategically place a limited number of agents, a significant improvement in network performance can be achieved.

The rest of paper is organized as follows: Section II describes the system model and formulates agent-assisted connection restoration problem. The weight map calculation and the detailed connection restoration scheme are described in section III. Section IV presents the simulation methodology and simulation results. Finally, section V concludes this paper and presents future work.

II. System model and Problem Formulation

In this paper, the following system model is considered. There are two types of nodes in a wireless optical backbone network: general mobile backbone node and controllable restoration agent[1]. Each node is equipped with a number of FSO transmitters and receivers, and an RF transceiver. The FSO transceivers are used to transmit high bandwidth data traffic in the backbone network, and the RF transceiver is used to collect the traffic within the cluster and transmit control messages in the backbone network, which has lower bandwidth. A unidirectional FSO link can be set up between a pair of FSO transmitter and FSO receiver within each other's transmission range. In the wireless optical backbone network, the backbone nodes can move randomly inside a certain area with low mobility, and the movement of agent nodes can be controlled to reduce the number of interruptions. We say there exists a *potential link* from node A to node B if they are in each other's transmission range, and we say there exists an *actual link* from node A to node B if they are in each other's transmission range and the FSO link has been set up from A to B. Furthermore, each transmitter or receiver can only participate in one actual link. Physically, to set up an actual FSO link between two nodes, the following procedures are needed: pointing, acquisition and tracking [8]. For a graphical model of the system, refer to [9].

In the network, each backbone node can collect the traffic demands within its cluster and issue the requests for aggregate traffic to other backbone nodes. In this paper we assume that the requests arrive in a random way. For each request, if sufficient resources are available, a bandwidth guaranteed connection is established between the source and destination; otherwise, the request is rejected. Specifically, to set up a bandwidth guaranteed connection for a request, we need two steps. In the first step, the source of a request performs route discovery to find a valid route with enough resources to the destination of the request. This means that (1) for each actual link in the route, the available link bandwidth is at least as great as the bandwidth requested; (2) for each potential link in the route, there is an available transmitter at the head node of the link and an available receiver at the tail node of the link. The route discovery can be achieved by using the existing mobile ad hoc network routing protocols, such as DSR [10] with some modification, and is usually executed through RF links. During this step, the positions of each node in the route are also returned to the source. After the route has been successfully discovered, the second step is to set up the FSO links on the route and reserve necessary bandwidth. If a link in the route is a potential link, then the actual link should be set up through necessary pointing, acquisition, and tracking. Each request has a random duration after which it leaves the network. When this happens, the bandwidth reservation for each link in the route is released. Any link for which there is no remaining bandwidth reservation is torn down and the corresponding transmitter and receiver are freed up. Thus, the actual link becomes a potential link.

Since a connection may fail due to path breakage, connection restoration mechanisms are applied to reduce the number of interruptions, where an interruption is used to indicate the situation that the broken connection cannot be restored in real time. In this paper, the connection restoration scheme is a combination of agent-assisted local restoration and end-to-end restoration. That is, once a node finds the link to another node is broken, it first checks whether the traffic on this link has been protected by some controllable agents. If the traffic is protected by some agents, that node reroutes broken traffic through the corresponding agents temporarily, which introduces negligible delay since that node does not need to

[1]In the paper, we use "general (backbone) node" and "agent" to distinguish the two type nodes, and use "nodes" to denote both of them.

do route discovery. If no such agents are available, the end-to-end restoration is applied, that is, a route is discovered and set up from the source to the destination and the corresponding bandwidth is reserved on the route. In this paper, the local restoration without using agents is not used, since the local restoration still needs to perform route discovery which introduces extra delay, and the new route is sub-optimal in general.

The problem is formulated as follows. For a FSO backbone MANET, let $\mathcal{N}$ denote the set of general backbone nodes, $\mathcal{A}$ denote the set of controllable restoration agents, and $\mathcal{E}$ denote the set of unidirectional links (including both actual and potential links) to connect all nodes. Each node keeps the record of its available interfaces (the number of FSO transmitters and receivers), and the residual bandwidth for actual link is kept with the corresponding edge. Let $\mathcal{P}$ denote the set of requests, where each request $P \in \mathcal{P}$ is defined by a triple (s_P, d_P, b_P) with s_P being the ingress node, d_P being the egress node, and b_P being the amount of bandwidth required for the request P. Let I_P denote the total number of interruptions corresponding to request P, then the objective of the system is to dynamically relocate the controllable mobile agents such that the total number of interruptions is minimized, that is,

$$\min \sum_{P \in \mathcal{P}} I_P \tag{1}$$

III. Description of Agent-Assisted Connection Restoration

At a high level, the basic idea of the agent-assisted connection restoration scheme can be described as follows. For a given bandwidth guaranteed connection request, once a path for the request has been successfully discovered and the corresponding resources have been reserved, the reservation will be maintained until the request expires or some links on the path break. For the first case, the request has succeeded, and the reserved resources are released. For the second case, the node on the head of the broken link checks whether the connections passing through it have been protected by some mobile agents. If a connection has not been protected by any mobile agents, that is, an interruption happens, then all the resources reserved by this connection are released, and a new route discovery will be issued by the source of the connection to find a path and reserve corresponding resources. If a connection is protected by certain mobile agent, the traffic on this connection will be temporarily rerouted through the mobile agent, and the source will simultaneously perform route discovery to find a new route without including mobile agents. Once the new path has been discovered, all the traffic corresponding to this connection will be rerouted through this new path, and the path with mobile agents will be released, as well as the resources reserved by the path.

A. Computation of Weight Map

In order to make the proposed agent-assisted connection restoration scheme effective, the controllable mobile agents should be located in an optimal way, and should be dynamically relocated according to the network status. In the proposed scheme, the decision on each mobile agent's location is made with the help of a weight map, where each position in the network has an associated weight in the weight map and this weight value indicates the protection degree that a controllable mobile agent can achieve if the agent is located at this position. By locating a mobile agent at a position with the largest value in the weight map, we expect it can best protect the existing network connections, and correspondingly minimize the number of interruptions.

Since the network topology changes dynamically, and the future connection patterns are usually unknown, the calculation of the weight map can only be based on the current network topology and connection patterns. In the proposed scheme, the following method is used to calculate the weight map. Each backbone node is associated with a sub-weight map whose values are determined by the connections passing through this node. For each mobile agent A, let $\mathcal{N}_A$ be the set of general backbone nodes in its neighborhood (e.g. its RF transmission range). For each backbone node $N_A \in \mathcal{N}_A$, let $Map(N_A)$ denote the sub-weight map calculated based on the connections passing through node N_A. Then the weight map of agent A's neighborhood is calculated as

$$Map(A) = \sum_{N_A \in \mathcal{N}_A} Map(N_A) \tag{2}$$

The sub-weight map associated with each general backbone node can be either calculated by the backbone node itself, and then submitted to the mobile agent, or it can be calculated by the mobile agent based on the information submitted by the backbone node, which includes the actual links connected to this node, the associated connections on the actual links, and the positions of its neighbors that have actual links with this backbone node.

We use Fig. 1 to illustrate how to calculate the sub-weight map associated with a general backbone node. Fig. 1 shows five general mobile backbone nodes and one mobile agent, and we consider the calculation of the sub-weight map associated with node C. There are 3 aggregate flows passing through node C: flow $B \rightarrow C \rightarrow D$ with 6 units of demand, flow $E \rightarrow C \rightarrow D$ with 8 units of demand and flow $E \rightarrow C \rightarrow F$ with 10 units of demand. The *aggregate flow* is the aggregate of all bandwidth guaranteed connections passing through the corresponding path. For example, aggregate flow $B \rightarrow C \rightarrow D$ is the aggregate of all bandwidth guaranteed connections whose paths includes the subpath $B \rightarrow C \rightarrow D$. The left-slanted-line area I is the intersection of the FSO transmission ranges of nodes B and D. The right-slanted-line area II is the intersection of the transmission areas of nodes D and E. And the dotted area III is the intersection of the transmission ranges of nodes E and F.

Given the general backbone node C, each aggregate flow passing through C corresponds to an area which may contribute to the sub-weight map associated with this node. Based on the geographical relationship among the nodes on them, the flows passing C can be partitioned into three types:

1) Type 1: flow $B \rightarrow C \rightarrow D$. For this type of flow, the

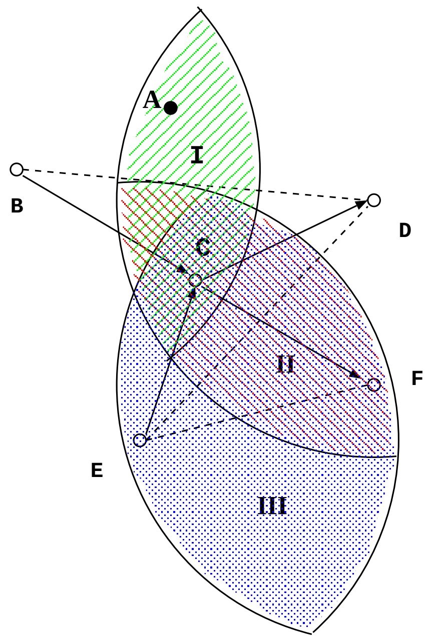

Fig. 1. Computation of Weight Map

node before C (node B) and the node after C (node D) are not in each other's FSO transmission range, but there is an agent A which lies in the intersection area (area I) of B and D's FSO transmission range. If node C moves out of B's or D's FSO transmission range, the flow can be rerouted through $B \rightarrow A \rightarrow D$ without introducing interruptions. Since the agents just take traffic temporarily, and the traffic will switch to new path after the source find the new path to release the agent, we assume the agents always have enough resources (transmitters, receivers and bandwidth) to help reroute the broken traffic when necessary. There is an agent in area I already, so there is no gain by placing an additional agent in area I. Let $w_{B\rightarrow C\rightarrow D}$ denote the contribution of flow $B \rightarrow C \rightarrow D$ to the sub-weight map associated with node C. Then we have

$$w_{B\rightarrow C\rightarrow D} = 0 \tag{3}$$

2) Type 2: flow $E \rightarrow C \rightarrow D$. For this type of flow, the node before C (node E) and the node after C (node D) are not in each other's FSO transmission range, and no mobile agent lies in the intersection area (area II) of E and D's FSO transmission range. If an agent is put into area II, it can potentially protect the flow $E \rightarrow C \rightarrow D$ when node C moves out of E's or D's transmission range, and the expected gain by putting an agent in this area should be proportional to the demand of the flow, which is 6 units. Let $w_{E\rightarrow C\rightarrow D}$ denote the contribution of flow $E \rightarrow C \rightarrow D$ to the sub-weight map associated with node C. Then we have

$$w_{E\rightarrow C\rightarrow D} = \begin{cases} 8 & \text{inside area II} \\ 0 & \text{outside area II} \end{cases} \tag{4}$$

3) Type 3: flow $E \rightarrow C \rightarrow F$. For this type of flow, the node before C (node E) and the node after C (node F) have built an actual link. If node C moves out of E's or F's FSO transmission range, the flow can be either totally rerouted through $E \rightarrow F$ if there is enough bandwidth, or can be partly rerouted $E \rightarrow F$ if there is not enough bandwidth, or can be fully protected by an agent inside the area III. On the other hand, if there is no actual link between node E and node F or agent in area III, the flow $E \rightarrow C \rightarrow F$ has no protection at all. Let $w_{E\rightarrow C\rightarrow F}$ denote the contribution of flow $E \rightarrow C \rightarrow F$ to the sub-weight map associated with node C, and let w_{left} denote the available bandwidth on the link $E \rightarrow F$. Then we have:

outside area III

$$w_{E\rightarrow C\rightarrow F} = 0 \tag{5}$$

and inside area III

$$w_{E\rightarrow C\rightarrow F} = \begin{cases} 0 & \text{if an agent lies in area III} \\ y & \text{if E and F have an actual link} \\ 10 & \text{if E and F have no actual link} \end{cases} \tag{6}$$

where

$$y = \max\{10 - w_{left}, 0\}.$$

For a general backbone node C, let $\mathcal{F}$ be the set of aggregate flows passing through C. After the contributions from all the aggregate flows passing through it have been calculated, the sub-weight map associated with node C can be calculated as

$$Map(C) = \sum_{f \in \mathcal{F}} w_f \tag{7}$$

where w_f is the contribution of aggregate flow f. That is, the sub-weight map of a general backbone node is simply the sum of the contributions from all the aggregate flows passing through it.

If a general backbone node has k transmitter interfaces and l receiver interfaces, then the maximum number of aggregate flows passing through that node is kl. Computing the weight function for each aggregate flow is constant time, so the computational complexity of the weight map for a backbone node is $O(kl)$. Since k and l are both small constants, the computational complexity of each general backbone node's sub-weight map can be regarded as a constant.

B. Dynamic Relocation of Agents

In the proposed scheme, the mobile agents are randomly allocated inside the area during the initialization procedure. Since the network topology and connection patterns change randomly with the time, to minimize the number of interruptions, the mobile agents should also be dynamically relocated according to the network status. After being allocated to the network, each mobile agent periodically collects the necessary information from the general backbone nodes which lie inside its RF transmission range, where that information is collected through RF channels. When calculating the weight map corresponding to this mobile agent, the effects of the mobile agent itself should be discarded. After calculating the weight map around its neighborhood, the mobile agent picks the position with the largest value in the weight map, and then moves to the new position.

When a mobile agent wants to move to a new position, it first notifies those backbone nodes with the aggregate flows that have been protected by this agent previously and cannot be protected further after movement. After a mobile agent has moved to a new position, it also notifies those backbone nodes with the aggregate flows that are not protected by this agent previously and now can be protected. Based on the notification, the general backbone nodes know which flows are protected by agents, thus can immediately reroute the broken connections through mobile agents upon link breakage, and correspondingly reduce the number of interruptions.

Since the weight map computational complexity for each general backbone node can be regarded as constant, and each mobile agent needs to collect information from $N_{neighbour}$ general backbone nodes, where $N_{neighbour}$ denotes the number of backbone nodes inside the mobile agent's RF transmission range, the overall computational complexity to calculate the weight map for each mobile agent is bounded by $O(N_{neighbour})$. If the sub-weight maps are calculated by each backbone node and submitted to the mobile agent through RF channel, the computational complexity is then distributed among the backbone nodes, while the drawback is that the extra bandwidth is needed to transmit the weight map. If the sub-weight maps are calculated by the mobile agent, and the backbone nodes only need to submit some necessary position and aggregate flows information, then the needed extra bandwidth can be very small, but the computational complexity at the mobile agent becomes $O(N_{neighbour})$. Based on the available RF bandwidth and computational capability, either computation method can be used.

IV. Performance Studies

A. Simulation Methodology

The simulation parameters are listed in Table I. We use a rectangular space of size 10000m × 10000m. The total number of general backbone nodes is 100, and the maximum FSO transmission range is 2000m. Each node has 4 FSO transmitters and 4 FSO receivers, and the FSO link bandwidth is 20 units. There are 40 source-destination traffic pairs randomly generated for each simulation. For each traffic pair, the bandwidth guaranteed connection request arrival time is modelled as a Poisson process, and the average request inter-arrival time is chosen uniformly between 500 to 1000 sec. For each request, the duration is modelled as an exponentially distributed random variable with mean 2500 sec, and the bandwidth demand is uniformly distributed between 2 to 6 units. When performing route discovery for a request, dynamic source routing (DSR) [10] is used, and the minimum-hop path including only general backbone nodes and satisfying the bandwidth and interface constraints is returned.

TABLE I

Simulation Parameters

Dimensions of Space	10km × 10km
Number of backbone nodes	100
Mobility model	Random waypoint
Maximum Velocity (v_{max})	20 m/s
Minimum Velocity (v_{min})	5 m/s
Number of FSO transmitters(receivers)	4
Maximum Transmission Range	2000 m
Link Bandwidth	20 units
Number of Traffic Pairs	40
Average request Inter-Arrival Time	500-1000 seconds
Average request duration	2500 seconds
Bandwidth demand	2-6 units

In the simulations, each backbone node moves randomly according to the *random waypoint model* [10]: a node starts at a random position in the network, waits for a duration called the *pause time*, which is modelled as a random variable with exponential distribution, then randomly chooses a destination location and moves towards it with a velocity uniformly chosen between v_{min} and v_{max}. When it arrives at that location, it waits for another random pause time and repeats the process. In the simulations, we set v_{min} to be 5m/s, v_{max} to be 20m/s, and use different average pause time.

In the simulations, the number of agents varies in different tests, and are put into the network in a controllable way. For each agent, after staying at the same position for 100 seconds, it will recalculate the weight map around its neighborhood, and move to the position with the highest weight value. In the simulation, the RF transmission range is also 2000m. All experiments are run for 20000 seconds. For each simulation setup, the results are averaged over 10 rounds, and at each round the random seed is changed to generate different mobility pattern and traffic pattern. Since random waypoint model is not stable at the beginning [11], [12], we add 5000 seconds extra warmup time to each experiment.

B. Simulation Evaluation

We first examine the average interruption number under different situations. Fig. 2 shows the simulation results for the average number of interruptions per request under various number of agents and different average pause time. From Fig. 2 we can see that the average number of interruptions per request decreases quickly with the increase of agent number, especially when the number of agents is small. For example, when using 5 agents, the number of interruptions per request is decreased by 40.5%, 40.0% and 40.0% when average pause time is 5000, 7500 and 10000 seconds respectively compared with the situation of no agent. Fig. 2 shows that the decrease speed of the average interruption number per request is higher when the mobility is relative higher. For example, for the average pause time of 5000 seconds, by adding 10 agents to the network, the average interruption number per request is reduced from 6.2 to 2.4. We can also see that the requests are interrupted much less frequently when some agents are added into the network compared with no agents. For example, for the average pause time of 5000 seconds, by adding 15 agents to the network, the request is interrupted every 20 minutes on the average; while the request is interrupted every 6.5 minutes on the average without agents.

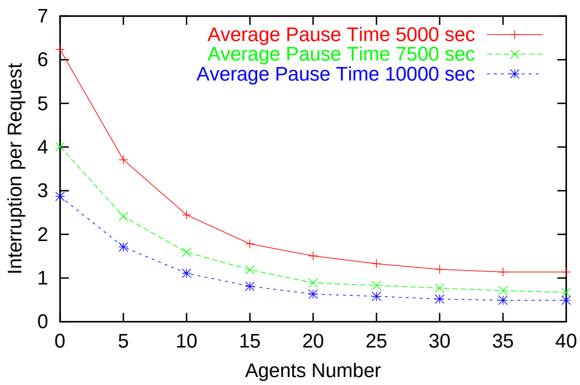

Fig. 2. Number of interruptions per request using different number of agent

From Fig. 2 it is also easy to see that there is little benefit from additional agents once the number of agents reaches 15. Furthermore, the number of interruption per request keeps the same when using 35 and 40 agents. The reason is that the current scheme only permits two-hop repair of the broken paths using agents, but some breaks require more than two-hop to repair them. For example, when both backbone nodes that associate to the same link move simultaneously, with a high probability there exists no agent to repair such link. The higher the mobility (smaller average pause time), the higher the probability two neighbor backbone nodes move simultaneously, so the higher the fraction of broken paths that cannot be repaired in two hops, which then leads to the increase of interruption number. In summary, the simulation results in Fig. 2 show that when adding more agents into the network, the number of interruptions per request decreases with smaller and smaller slope, but the interruption may not go to zero.

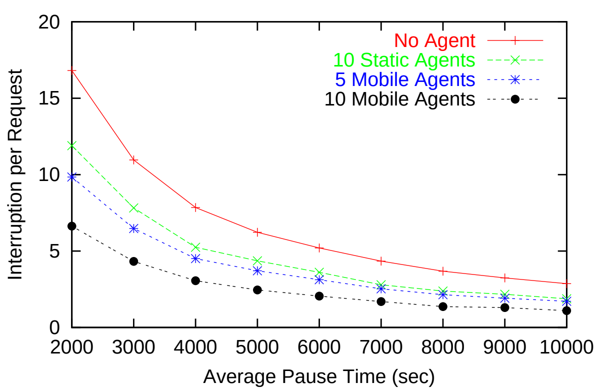

Fig. 3. Difference between using static agents and mobile agents

Fig. 3 shows the comparison results under different agent setup. In Fig. 3, "mobile agents" denotes the setup where agents can be dynamically relocated, while "static agents" denotes the setup where agents are randomly distributed inside the area according to a uniform distribution and will not be relocated during the whole simulation time. From Fig. 3 we can see that the effect of 10 static agents is much worse than 10 mobile agents and even worse than 5 mobile agents. For example, when the average pause time is 5000 seconds, compared with the situation that there is only 100 general backbone nodes, the number of interruption per request is reduced by 60.7% when using 10 mobile agents, but by only 30% when using 10 static agents. Thus, network status-dependent relocation of agents performs much better than random location of agents.

V. Conclusion

The primary contributions of this paper are the proposal of a new type of mobile node, the controllable restoration agent, and the development of a scheme for local restoration using these agents in mobile wireless optical backbone networks. The agents are allocated into the network uniformly at first and relocated periodically according to the weight map computed by the agent itself. Their function is to help repair broken connections locally. The bandwidth guaranteed connection is setup when an aggregate request arrives, and it may break during the lifetime of the request due to node mobility. When a connection breaks, local restoration is used to repair it using agents when possible. Through extensive simulation experiments, we show that the proposed scheme works well.

References

[1] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. IT-46, no. 2, pp. 388–404, March 2000.

[2] K. Xu, X. Hong, and M. Gerla, "An Ad Hoc Network with Mobile Backbones," in *IEEE International Conference on Communications (ICC'02)*, New York, NY, April 2002.

[3] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," in *INFOCOM*, April 2001.

[4] M. S. Kodialam and T. V. Lakshman, "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration," in *INFOCOM*, 2000, pp. 902–911.

[5] K. Kar, M. Kodialam, and T. V. Lakshman, "Routing Restorable Bandwidth Guaranteed Connections using Maximum 2-Route Flows," in *INFOCOM*, 2002.

[6] M. S. Kodialam and T. V. Lakshman, "Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels Using Aggregated Link Usage Information," in *INFOCOM*, 2001, pp. 376–385.

[7] L. Mélon, F. Blanchy, and G. Leduc, "Decentralized Local Backup LSP Calculation with Efficient Bandwidth Sharing," in *Proc. of 10th IEEE International Conference on Telecommunications (ICT'2003)*, Papeete, Tahiti, Feb 2003, pp. 929–937.

[8] N. Riza, Ed., *Optics in Information Systems*. Special Issue: Optical Wireless Communications, Oct. 2001, vol. 12, no. 2.

[9] F. Sun and M. Shayman, "Minimum Interference Algorithm for Integrated Topology Control and Routing in Wireless Optical Backbone Networks," in *IEEE International Conference on Communications*, Paris, France, June 2004.

[10] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing," In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[11] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *INFOCOM*, San Francisco, CA, 2003, pp. 1312–1321.

[12] ——, "Sound mobility models," in *ACM MobiCom*, San Diego, CA, September 2003.