

MultiServ: A Service-Oriented Framework for Multihop Wireless Networks

Xinyan Zhang, *Student Member, IEEE*, Qian Zhang, *Member, IEEE*, Bo Li, *Senior Member, IEEE*, Wenwu Zhu, *Senior Member, IEEE*, and Tak-Shing Peter Yum

Abstract—In order to enable fast deployment of new emerging services over multihop wireless networks, it is important to design an efficient service-based platform with the necessary traffic management capabilities. In this paper, we propose a new distributed service-oriented framework for wireless multihop networks, called MultiServ, in which it adopts a quantitative approach toward optimal traffic distribution. Under Multiserv framework, an efficient overlay network can be easily constructed that can greatly facilitate the deployment of new services. We use media streaming and application level multicast as examples to illustrate how the services can be supported. The performance results demonstrate that MultiServ can substantially outperform the conventional approach and achieves comparable performance obtained by a centralized scheme.

Index Terms—Overlay networks, traffic engineering, wireless multihop networks.

I. INTRODUCTION

MUCH OF THE recent work in wireless multihop networks, including ad hoc networks and wireless mesh networks, has been focusing on routing protocols dealing with nodes' mobility, changing topologies, and scalability. Less attention has been paid to the service deployment and quality-of-service (QoS) guarantee in the presence of noisy wireless links. In this paper, we address the problem of traffic provisioning in wireless multihop networks with the objective of facilitating new wireless service deployment; specifically, we focus on the problem of how to manage route large volume of traffic from different applications. We propose an overlay framework that can efficiently support multiple traffic streams.

Manuscript received February 15, 2004; revised January 15, 2004. The work of X. Zhang and B. Li was done while with Microsoft Research Asia. This work was supported in part by grants from RGC under Contract HKUST6402/03E and Contract HKUST6104/04E, in part by a grant from the National Science Foundation (NSF), China, under the Contract 60429202, in part by a NSFC/RGC joint grant under Contract N_HKUST605/02, and in part by a grant from Microsoft Research under Contract MCCL02/03.EG01.

X. Zhang is with the Wireless and Networking Group, Microsoft Research Asia, Beijing 100080, China (e-mail: xyzhang2@ie.cuhk.edu.hk).

Q. Zhang is with the Beijing Sigma Center and the Wireless and Networking Group, Microsoft Research Asia, Beijing 100080, China (e-mail: qianz@microsoft.com).

B. Li is with the Wireless and Networking Group, Microsoft Research Asia, Beijing 100080, China, and also with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: bli@cs.ust.hk).

W. Zhu was with the Wireless and Networking Group, Microsoft Research Asia, Beijing 100080, China. He is now with Intel Communication Technology Laboratory, China, Beijing 100080, China (e-mail: Wenwu.zhu@intel.com).

T.-S. P. Yum is with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: tsyum@ie.cuhk.edu.hk).
Digital Object Identifier 10.1109/JSAC.2005.845624

Generally speaking, our solution provides a distributed traffic provisioning strategy, which facilitates the dissemination of traffic along multiple paths so that high transmission efficiency can be achieved.

The motivations for this study are: first, the capacity of a wireless multihop network is not only constrained by the scarce wireless bandwidth, but also limited due to the contention caused by multihop relay; thus routing remains as a key problem; second, packet transmission is subject to possibly link failure, in which a multipath routing scheme becomes essential for applications such as media streaming and multicast; third, in order to facilitate the deployment of multiple services over wireless networks, each node must be capable of handling multiple traffic streams with potentially different QoS requirements.

We propose a service-oriented framework called *MultiServ* that builds upon the overlay concept to provide enhanced QoS for multiple services in wireless networks. The realization of the MultiServ architecture requires cooperation of multiple nodes. We construct an overlay network, in which each node behaves as a *sender*, as well as a *receiver*. There are two critical issues that need to be addressed in this framework: 1) overlay network construction that deals with the neighbor selection and 2) multipath routing in the overlay network. In addition, this needs to incorporate a number of factors such as link utilization, load balancing, and forwarding performance. One of the key novelties in the proposed framework is a distributed rate-based forwarding algorithm that can reduce the burstiness of different traffic stream and hence reduce the service fluctuation.

A. Related Work

Before we describe the MultiServ framework, we briefly review existing related works. To the best of our knowledge, MultiServ is the first quantitative approach toward optimal traffic distribution for an overlay network in wireless environment. The prior works related to this mainly fall into two categories. The first is the Internet protocol (IP) network traffic engineering, and the second is the QoS based routing in wireless multihop networks.

Traffic engineering is used to manage traffic in order to optimize the usage of network resources such as to balance traffic distribution across the network and to avoid congestion. An overview of IP traffic engineering can be found in [4]. One of the problems in IP network traffic engineering is the cost in the deployment since this often requires the upgrade across networks. MultiServ also targets to balance traffic, and it performs at the application level on top of current network protocol stacks.

Comparing with many centralized optimization in traffic engineering [2], [3], the distributed nature of our design makes it more scalable and practical, especially suitable for wireless network environment.

It has been shown in [10] that the per node capacity in an n -node random ad hoc network is $\Theta(1/\sqrt{n \log n})$, using a geometric analysis. [26] shows that the long term per node throughput can stay constant in a network where node movement process is ergodic with a stationary distribution uniform over the network. In this work, we took a rather different approach in trying to optimize the traffic relaying to facilitate the service deployment.

There have been many QoS algorithms proposed for wireless networks in particular medium access control (MAC) level scheduling [15], [24], in which it relies on datalink layer to provide bandwidth and delay information. None of the approaches however explicitly consider link failure or address the issue of service deployment. Such approaches can complement MultiServ.

The remainder of the paper is organized as follows. Section II presents the general framework of MultiServ platform. Section III introduces the mechanism to construct the MultiServ overlay network. Section IV discusses the traffic management strategy and implementation. Section V verifies the model using simulation and experiments. In Section VI, we conclude the paper and highlight the possible avenues for further study.

II. OVERVIEW

The service deployment in wireless networks encounters several new problems such as link errors, delay variations, heterogeneity in the end devices, handoffs, and the lack of potential infrastructure support. Proper traffic engineering scheme can help to alleviate this, in particular using multipath transmissions. In this section, we present the general framework for service deployment, and discuss the potential applicability.

A. General Framework

To facilitate users to access services in a wireless environment without or with limited infrastructure support, we assume that some of the nodes are capable of serving as application-layer routers. Such nodes have the capability of connecting to the Internet through various interfaces. These can be considered as the application-layer router for traffic relay, thus, to form a hybrid wireless network [29]). An example of such a network is illustrated in Fig. 1.

In Fig. 1, two WLAN networks are constructed using 802.11 ad hoc mode with various devices such as laptops, PDAs, and smart phones. To enable nodes to connect to outside world, some of the nodes are equipped with multiple network interfaces, for example, node A and D have satellite dishes to communicate with satellite, node B and F have WMAN antennas, node C and E have networking cable connecting Internet.

This hybrid network can further be extended to hierarchical network for scalability. In Fig. 1, the application-layer routers (A-F) can form a second level overlay network to relay the traffic more efficiently. As the network becomes larger, higher level

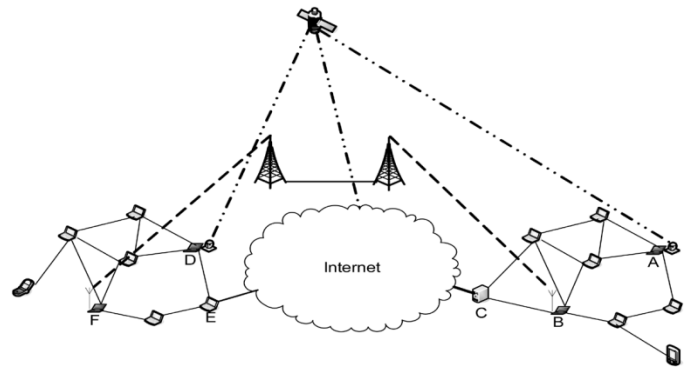


Fig. 1. Example of hybrid wireless networks.

overlay network can also be formed. The formulation of a higher level overlay network can also exploit the locality information in underlying wireless network similar to the way in the Internet environment [27], [28].

B. Service Deployment and Application Scenario

For simplicity, we adopt IP packets for data transfer, though the idea is also applicable to the networks that use other identifications for addressing. Each node has an IP address, normal transmission control protocol (TCP) connections can be established between a pair of nodes. In MultiServ platform, we can encapsulate IP packets to perform special operations such as application-layer routing with rate information. The information will be used by each node to calculate how to relay the packet. Detailed will be presented in Section IV.

Traffic can be divided into two types: intradomain and inter-domain traffic. For intradomain traffic, since the destination is in local network, the traffic will simply be forwarded. For inter-domain traffic, the application-layer router will be responsible for forwarding. Our proposal tries to increase the transmission efficiency by avoiding lossy links and reducing the hop count between a pair of nodes; this will be introduced in Section III.

Multimedia streaming is one of the possible applications. If this is within the same network, the proposed MultiServ platform can support the streaming from the source to the destination with lower data loss and higher reliability using multiple available (disjoint) paths. If this is between two nodes in a different network, the traffic will be handled by application router(s). This will be closely examined using simulation in Section V-B.

Another possible application is multicast. In multicast service, a user who would like join the multicast group will broadcast the request to other nodes with a time-to-live (TTL) limit, the node will increase TTL limit until it finds a group member, which has the nearest hop-distance. The node then will retrieve data from the member. If this fails in the local network, the application-layer router will be responsible for locating the members in another network. This will be studied in Section V-C.

III. MULTISERV OVERLAY CONSTRUCTION

Overlay network has been proposed as a general approach to facilitate service deployment in the Internet [30], [31]. The link in the IP based overlay network is a unicast tunnel

in the physical IP network, in which IP layer implements a minimal functionality—the best-effort datagram service, while the overlay network can implement a rich set of functionalities, like path resilience, QoS support, etc. There have been various proposals for overlay network construction, which in different degree try to exploit the topological information, particularly the locality in the underlying network. For example, application-layer multicast and CDN network are of this kind, where each node keeps nearby neighbors in the underlying network in order to transfer data efficiently.

In wireless multihop networks, the situation is different in that there is limited infrastructure; each node behaves as a router, as well as an end host. In fact, the structure of wireless network can only represent the relative position information of nodes. Overlay network can be built on top of the physical network, but the overlay structure must be closely coupled with the structure of underlying network. The rationale is that the transmission in wireless network not only consumes energy, but also is subject to link failure. Thus, tunneling between two distant nodes can be expensive and impractical.

The MultiServ platform is designed to facilitate the service deployment and to enhance the traffic delivery performance in wireless multihop networks. However, using the physical topology to perform traffic delivery suffers serious drawbacks because of the unpredictable nature of the wireless links. Therefore, building an overlay on top of wireless network becomes a necessary yet challenging problem. The purposes of building an overlay network are two-folded. The first is to maintain better “virtual” link quality among neighboring nodes and enhance the network connectivity; the second is to provide underlay information for the traffic management algorithms. We next describe the overlay network construction algorithm.

A. Neighbor Selection Criterion

The characteristic of a wireless link is quite different from that of a wired link. In fact, a link exists between two wireless nodes if and only if the receiving node has a higher signal-to-interference-plus-noise-ratio (SINR) than a given threshold. Two nodes can establish a wireless connection as long as they are either sufficiently close with each other, or with sufficient transmission power. Thus, it is not feasible to build an arbitrary topology due to the constraints such as power limitation, interferences and channel fading. The construction of an overlay network has to carefully balance several factors in order to maintain reasonable capacity. These include the interference along a multihop path, the loss ratio of a wireless link, and the failure of a link or node.

The interference and failure are usually handled by the physical networks; we focus our study on the loss ratio of a wireless link. Specifically, the neighbor selection algorithm in overlay construction aims to find links with low-loss ratio. Intuitively, a link with lower loss ratio can transmit more packets comparing with a higher loss ratio link during the same period of time. To select links with low-loss ratio, we use a probing packet technique similar to the one used in [6] to perform a measurement of the loss ratio of links. Each node periodically broadcasts a packet, say every p s. To avoid accidental synchronization, up

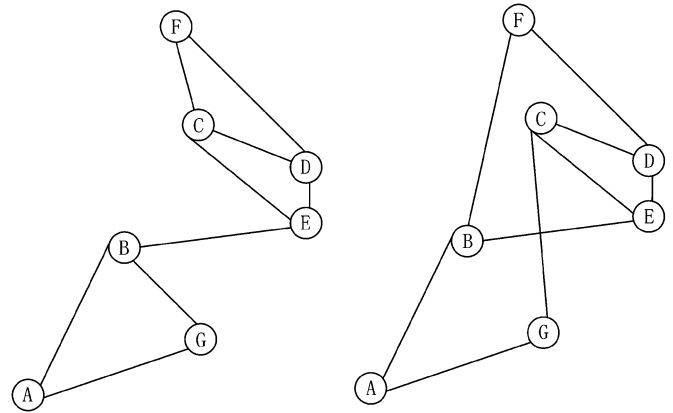


Fig. 2. Neighbor selection illustration.

to $0.1 p$ variation is introduced for each probing. Every node records the probes it receives during the last w s, allowing it to calculate the delivery ratio from the sender at any time t as

$$r(t) = \frac{\text{count}(t-w, t)}{\frac{w}{p}}.$$

$\text{Count}(t-w, t)$ is the number of probes received during the window w , and w/p is the number of probes that should have been received. Therefore, a node Y can calculate the link loss ratio for link X - Y and a node X can calculate that for link Y - X .

Knowing the loss ratio information, a node can determine the link quality and select nodes with lower loss ratio as neighbors. When finding a neighbor and forming a logical link to the neighbor, the capacity of the link will also be reported.

B. Neighbor Selection

The construction of an overlay network has to maintain high aggregate throughput in order to support large volume of traffic, and to provide high connectivity, i.e., to minimize the average hop count in order to improve the traffic relay efficiency. Thus, in an ideal overlay network, each link has high throughput and each pair of nodes have short hop distances. These are not always feasible. High throughput can be achieved by using links between closely nodes, but this often results in higher hop count. An example of neighbor selection is given in Fig. 2, simply by exchanging link C-F, B-G by B-F, C-G, the average hop distance decreases from 2 to 1.71. If each link has a fixed capacity and each node has a similar traffic demand, the traffic delivery efficiency will be increased by $(2 - 1.71)/1.71 = 17\%$. Furthermore, we can see that more disjoint paths are available in the network, such as A-B-E-D and A-G-C-D, which leads to improvement in handling failures.

We next formally describe the problem formulation. Given an overlay network represented by a directed graph $G = (V, E)$. Each node in the overlay network will be allocated the same total outbound bandwidth b for fairness and simplicity; the bandwidth can be arbitrarily allocated to transfer data to the neighbors. However, each links may have different delivery ratios due to different environment, that is, link x - y has a delivery R_{xy} .

Suppose node pair x - y has traffic t_{xy} , so the total traffic T generated in the network can be calculated as

$$T = \sum_{x \in V} \sum_{y \in S_x} \text{Distance}(x, y) \cdot t_{xy}.$$

Suppose each pair has equal probability to transfer traffic between, so we can use a value t to represent all t_{xy} . Therefore, the upper bound of traffic the network can transfer for each node is

$$\begin{aligned} t &= \frac{T}{\sum_{x \in V} \sum_{y \in S_x} \text{Distance}(x, y)} \\ &\leq \frac{\sum_{l \in E} \text{Capacity}(l)}{\sum_{x \in V} \sum_{y \in S_x} \text{Distance}(x, y)} \\ &= \frac{\sum_{(x,y) \in E} \text{Allocated_Bandwidth}(x, y) \cdot R_{xy}}{\sum_{x \in V} \sum_{y \in S_x} \text{Distance}(x, y)}. \end{aligned} \quad (1)$$

The value t is preferred to be as large as possible. A natural way is to maximize the upper bound, the right side of (1). In building the overlay network, each node x has a set of node S_x where nodes in S_x can be a neighbor of node x . We can build an overlay network to have a maximum traffic relaying efficiency by selecting appropriate neighbors from the candidate set for each node to increase the right side of (1), for example, to increase capacity of links or to decrease the average hop distance between all pairs.

In an extreme case if all the delivery ratios are similar, such as a situation that all links are zero loss links, the optimization problem can be simplified to build an overlay network with the shortest average hop distances. Such an overlay network can be constructed by connecting all available potential neighbors. However, the solution is not feasible due to the energy consumption and computational costs by relaying traffic to many neighbors. So practically, the maximum number of neighbors that each node can have must be limited, in this paper, we use up to four neighbors.

Generally speaking, the optimal solution to minimize right side of (1) is not an easy problem even in centralized environment. We use a distributed algorithm to refine neighbor selection to provide better delivery performance, which is given in Fig. 3.

The neighbor refinement algorithm finds an appropriate neighbor to replace current neighbor in order to decrease the total hop count. In a refinement process, we test all combinations for a potential neighbor X see if replacing any neighbor with X can have lower total distance vector, if so the algorithm performs the neighbor replacement.

C. Distance Vector Maintenance

To manage traffic in overlay network, we define a *distance vector* for each node. We adopt the similar concept used in destination sequenced distance vector (DSDV) protocol [23] to maintain the distance vector. A distance vector is defined as the shortest hop distance from node i to node j in the overlay network, namely D_{ij} . Unlike the DSDV protocol, the routing is done differently in our scheme.

Algorithm: Neighbor Refinement

1. Find a potential neighbor X with a satisfied delivery ratio to current node A .
2. Fetching X 's distance vector V_x , A 's neighbors' distance vectors V_i and X 's neighbors' distance vectors V_j . And also other necessary information to calculate distance vectors of A and X and their neighbors, such as A 's neighbors' neighbors' distance vectors.
3. Calculate distance vectors on all possible combination by exchanging neighbors from A-B and X-Y to A-X and B-Y. V_{ij} is the distances vector in node N_{ai} with neighbor N_{sj} , and V_{ji} is the distances vector in node N_{sj} with neighbor N_{ai} . V_{Ai} is the distance vector in node A when replacing neighbor N_{ai} with X and V_{Xj} is the distance vector in node X when replacing neighbor N_{sj} with A .
4.
$$\text{Min}V |_{A_i, X_j} = \min(\text{sum}(V_{Ai} + V_{Xj} + V_{ij} + V_{ji}) - \text{sum}(V_A + V_X + V_i + V_j));$$
5. If ($\text{Min}V < 0$) replace neighbor A_i with X , and X replace neighbor X_j with A .

Fig. 3. Neighbor refinement algorithm.

In the routing protocol, each node maintains a routing table entry for every known destination. The entry contains three fields: destination's identifier (location or address), the latest sequence number for that destination, and the metric, typically, the shortest hop distance between the node and the destination. Each node periodically broadcasts a route advertisement packet containing its complete route entry. When a node receives another node's route advertisement, it updates its own route entries if the routing advertisement contains more updated route information.

IV. MULTISERV TRAFFIC MANAGEMENT

In this section, we first highlight the motivations for traffic management in overlay networks; we next briefly examine the traffic management schemes used in IP network. We then present the traffic management in MultiServ architecture and discuss the implementations.

A. Motivations for Traffic Management

A traffic management strategy is desired in overlay network, in particular, in MultiServ platform. We illustrate this using an example shown in Fig. 4.

In Fig. 4, each link represents a logical connection between two nodes, which has a capacity of 6 units and each traffic demand requires bandwidth of 4 units. We assume the link capacity is bidirectional here for simplicity. To satisfy the traffic demands, there are multiple ways. For example, using fixed single shortest paths to forward data, some of the demand may not be satisfied if two demands share one common link. If we use the shortest path forwarding where the traffic are equally divided into each shortest paths, that is, demand A to G uses paths A-F-G and A-D-G, demand A to C uses paths A-D-C and A-B-C, demand A to E uses paths A-F-G-E, A-D-G-E, A-D-C-E and A-B-C-E, we can obtain the link utilization in Table I.

From Table I, we see that A-D link is heavily loaded, thus ideally some of the traffic on A-D should be moved to links A-B and A-F in order to achieve high aggregate throughput.

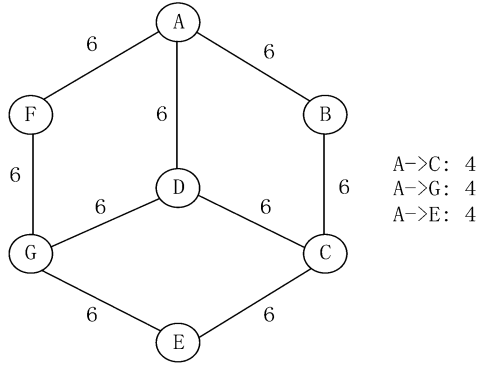


Fig. 4. Example of traffic management strategy.

TABLE I
LINK UTILIZATION OF THE EXAMPLE

Link	A-B	A-D	A-F	B-C	D-C	D-G	F-G	G-E	C-E
BW	3	6	3	3	3	3	3	2	2
Util	0.5	1.0	0.5	0.5	0.5	0.5	0.5	0.33	0.33

B. Multiserv Traffic Management

Traffic engineering has been used in the Internet to achieve better resource utilization and to optimize network performance. The output of traffic engineering is an “optimal” set of paths and link loads that produce the best possible performance given the available resources. The problem can be formulated into the linear programming problem. By solving the optimization problem, each link can have a weight so that the routing protocol will select the data forwarding paths in order optimally deliver the traffic. Reference [2] solved the dual problem by a computational simplified solution of the linear programming. Reference [3] achieved a near-optimal solution using destination-based aggregation of traffic and approximating unequal split of traffic using heuristics for traffic splitting to deal with the problem of unequally splitting the traffic.

Overlay network offers more flexibility in that it does not specify the particular communication protocol, but encounters several new problems: 1) each node also serves as the relay node that is less stable and 2) the link capacity in the overlay network varies due to the interference and other factors. As a result, the above described solution for the Internet can not be directly applied. In addition, a distributed traffic management is preferred.

The overlay network can be modeled as a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of links. Let c_{ij} be the capacity of logical link $(i, j) \in E$, M_{ik} is the traffic demand to node k , I_{jk} is the traffic input from node j , where $(j, i) \in E$ and the destination is node k , D_{ik} is the shortest hop distance to node k , and D_j is the distance vector of the neighbors transferred to node i for data forwarding in node j , where $(i, j) \in E$.

Given the above information for each node, it is not possible to obtain a global optimization. However, each node may obtain its local optimization. The objective of the optimization is to minimize total traffic generated and balance the utilization of logical links, while satisfying as much traffic demand

as possible. For node i , a linear programming formulation can be obtained

$$\min \quad \alpha + \varepsilon \sum_{k \in V} \sum_{(i,j) \in E} (D_{jk} - D_{ik} + 1) X_{jk} \quad (2)$$

s.t.

$$\sum_{(i,j) \in E} X_{jk} = M_{ik}, \quad k \in V \quad (3)$$

$$\sum_{k \in V} X_{jk} \leq \alpha c_{ij}, \quad (i, j) \in E \quad (4)$$

$$\begin{cases} X_{jk} = 0, & I_{jk} > 0, (i, j) \in E, (j, i) \in E \\ X_{jk} \geq 0, & \text{otherwise} \end{cases} \quad (5)$$

where X_{jk} is the traffic going through link (i, j) with the destination k , α is the maximal utilization of the links from i to its neighbors, ε is a small positive number introduced so that the optimization not only minimizes the link utilization, but also T_i and ensure that the minimization of link utilization takes higher priority. Equation (3) means that the total flow rate to node k is M_{ik} . Constraint (4) indicates that the utilization of each logical link will be less than or equal to α . Equation (5) prevents sending back the traffic to the node that it receives from. The objective function (2) is to minimize the link utilization so that the traffic can be relayed properly. Notice that if all traffic goes through the shortest path to the destination, that is through node j where $D_{jk} = D_{ik} - 1$, the generated traffic will be minimized. However, not always the traffic demand can be transmitted through shortest path in the overlay networks; therefore, traffic demand that may not go through the shortest path will introduce extra traffic. Specifically, in node i the extra traffic introduced is

$$T_i = \sum_{k \in V} \sum_{(i,j) \in E} (D_{jk} - D_{ik} + 1) X_{jk}. \quad (6)$$

C. MultiServ With Bounded Complexity

The traffic demand may vary from time to time, thus, it is necessary to calculate the bandwidth allocation dynamically. Solving linear programming problem can be time consuming and it may not be possible, especially for devices with low computation power in mobile network. Therefore, next we propose the following complexity-bounded heuristic method. Notice that to avoid generating extra traffic; the data forwarding should mainly use the shortest path. Under this principle, the heuristic method basically uses the shortest path first scheduling. In order to forward as much traffic as possible using shortest path forwarding, we can formulate a maximum flow problem as follows.

To maximize the traffic going through the shortest path, we use a virtual graph (G_s, E_s) illustrated in Fig. 5. In the graph, S node is a virtual source which launches entire traffic request in node i . T node denotes traffic demand where the capacity from the source to T_k is M_{ik} , and N node denotes the neighbor. The capacity from N_j to the E node is c_{ij} corresponding to the capacity of neighbors. A link (T_k, N_j) is present if traffic to node k can be forwarded through node j as a shortest path, where $D_{jk} - D_{ik} + 1 = 0$. M is a large value so that the bottleneck of any path from S to E will not be the link which has capacity M .

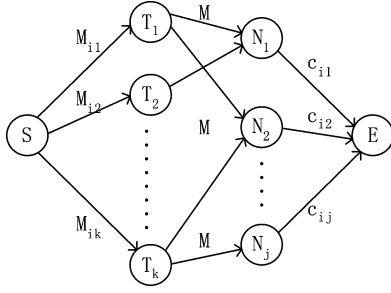


Fig. 5. Shortest path maximum flow.

By solving the maximum-flow problem from S node to E node, we can transfer maximal traffic through the shortest path where the flow in (T_k, N_j) represents the traffic going through link (i, j) which has the destination k . However, there might be traffic demand that is not satisfied. Fortunately, the rest part of the traffic that cannot be forwarded using the shortest path can be also scheduled using the similar graph in Fig. 5. The satisfied part of traffic demand and consumed capacity should be removed from the graph and the structure of the graph also need small modification, where $(T_k, N_j) \in E_s$ for every $I_{jk} = 0$.

The complexity of the best known maximum-flow algorithm is around $O(|V||E|)$. Suppose each node has n neighbors and we have totally m nodes in the overlay network, the graph in Fig. 5 has $m + n + 2$ nodes and less than $(n + 1)m + n$ edges. Therefore, the complexity of our algorithm can be bounded under $O((m + n)mn)$.

The maximum-flow method does not optimize the link utilization. Therefore, a load balance algorithm can be further applied over each link. Let X_{jk} be the traffic going through link (i, j) which has the destination k . Define U_j as utilization of link (i, j) , where $U_j = \sum_{k \in V} X_{jk} / c_{ij}$. The objective is to minimize the variance of link utilization, which is defined in the following:

$$V_i = \sum_{(i,j) \in E} \sum_{(i,k) \in E} (U_j - U_k)^2. \quad (7)$$

Therefore, we propose a heuristic method to decrease the variance iteratively. The algorithm in Fig. 6 tries to find unbalanced traffic in links and move traffic with the same destination from high utilized link to low utilized link, while keeping the extra traffic T_i unchanged. The reroute of traffic will decrease the objective function V_i each time. The execution of the algorithm continues until no more traffic could be rerouted.

D. Data Transmission Control

In our architecture, the node can use TCP connections to communicate with neighbors, but this incur several difficulties: 1) packet can be transmitted out of sequence; 2) the rate of TCP connections cannot be easily controlled, as a result the performance is unpredictable and uncontrollable; 3) multiple TCP connections may involve in the burst of traffic so that the relayed nodes' buffer may be filled and aggravates the congestion; and 4) some applications, such as streaming, are not suitable to be carried by TCP.

To address this problem, we propose a rate-based congestion control algorithm based on the idea of congestion management

```

Decrease_Pair_Variance( $a, b, U$ )
{
  For each Destination  $x$ 
  if ( $D_{ax} = D_{bx}$ ) and ( $X_{ax} > 0$ ) {
     $t = \min(X_{ax}, \sum_{k \in V} X_{ak} - Uc_{ia}, Uc_{ib} - \sum_{k \in V} X_{bk})$ ;
     $X_{ax} = X_{ax} - t$ ;
     $X_{bx} = X_{bx} + t$ ;
    if ( $\sum_{k \in V} X_{ak} = Uc_{ia}$  or  $\sum_{k \in V} X_{bk} = Uc_{ib}$ )
      return;
  }
}
    
```

```

Decrease_Variance()
{
   $U = \sum_{(i,j) \in E} \sum_{k \in V} X_{jk} / \sum_{(i,j) \in E} c_{ij}$ ;
  for all neighbor  $a$ 
  for all neighbor  $b$ 
  if ( $\sum_{k \in V} X_{ak} > U * c_{ia}$  and  $\sum_{k \in V} X_{bk} < U * c_{ib}$ )
    Decrease_Pair_Variance( $a, b, U$ );
}
    
```

Fig. 6. Heuristic algorithm to decrease variance of link utilization.

(CM) [1]. The basic idea is to use flow aggregation. The aggregated flow to one neighbor can use an additive-increase multiplicative-decrease (AIMD) congestion control in order to be friendly to background TCP flows if any. The sending rate increases when there is no packet loss. Upon a packet loss, the rate will be halved. When persistent congestion occurs, the rate drops to a small value forcing slow start to occur. An automatic repeat request (ARQ)-based mechanism can be adopted. The sender will retransmit the packet until receiving the acknowledgment.

In MultiServ architecture, congestion still can occur. Flows from a node are aggregated while sending to its neighbors; each flow may have different transmission rate. Through aggregation, we can easily control the rates by adjusting the weights in the aggregate flow. Specifically, we control the aggregate rates instead of individual flow sending rate. In transmission, the rate to neighbor i can be measured, say r_i . The sender will use a unified sending rate of $\sum r_i$ instead of individual sending rate r_i . Flow i can be controlled using a weight w_i , where $w_i = r_i / \sum r$. A round-robin scheduler is used for transmission. Apparently, more sophisticated schedulers can also be used.

An illustration is presented in Fig. 7. Suppose the rate with AIMD control for the four destinations are 400, 300, 200, and 100 kb/s, respectively, and the link bandwidth is 1 Mb/s. On average, four of ten packets will be sent to neighbor 1, 3 will send to neighbor 2, and, etc. Packets transmission from a node can be controlled and, thus, smoothed using aggregation. The interval of packets through each node along the path will be approximately equivalent under no congestion. Each node including relayed nodes in the overlay network will use the same scheduling algorithm to control the packet transmissions. The relayed nodes perform additional tasks such as buffering and status feedback for the sender to adjust the rates. The out of sequence packets are handled at the destination.

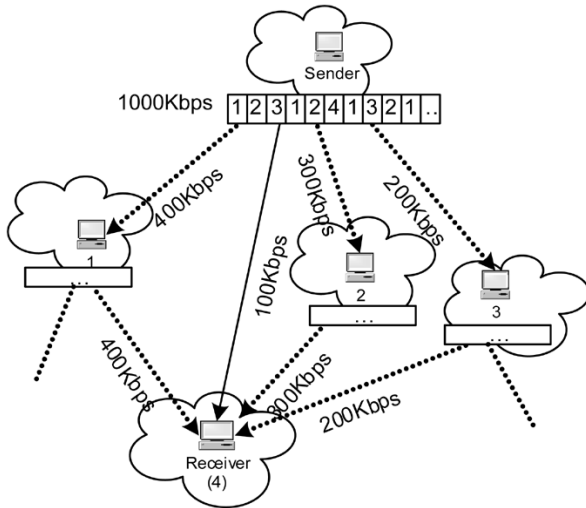


Fig. 7. Example of joint congestion control.

There are several unique advantages using aggregated rate control: 1) it enables more flexibility in rate control; 2) it minimizes the burstiness of traffic; and 3) the rate of individual flow can be easily controlled by adjusting the aggregated rates or/and the associated weight.

V. PERFORMANCE EVALUATION

In order to evaluate the effectiveness of our approach, we conducted simulations to verify the effectiveness of the proposed traffic management scheme. The purpose of simulation is to compare different traffic management strategies for different applications and under dynamic environment.

A. Overlay Construction

We use simulation to evaluate the overlay construction, mainly on the neighbor refinement algorithm. In our simulation, we constructed the overlay network in a 1 km * 1 km space, the nodes are randomly placed. We generate m nodes and each node has up to four neighbors. We compare our refinements algorithm with *random neighbor* method, where the neighbors are selected randomly from the satisfied nearby nodes. Fig. 8 illustrated a sample refinement overlay network. In the original random network [Fig. 8(a)], 20 nodes are spreading randomly in the 1000 m * 1000 m space, each node select up to four neighbors randomly from the nodes less than 500 m away. The resulting overlay network has an average hop distance of 2.66. We use the neighbor refinement algorithm described in Section II-B to refine the neighbor set for each node. After 32 effective replacements of neighbors, Fig. 8(b) shows the result. The refined overlay network has a decreased hop distance of 2.17, which is 82% of the original one. That means, the refined overlay network may carry 22% more traffic than the original one under similar link capacities.

To further evaluate the performance of neighbor refinement, we test the algorithm in network different sizes (from 20 nodes to 50 nodes, each node with up to four neighbors), we generate 100 overlay networks using random neighbor method for each size, and then use the algorithm to refine the neighbor set. The average hop distance is illustrated in Fig. 9. We can see that in

each sized overlay network (except very small overlay with ten nodes), the average hop distance decreased more than 15% of the original one.

B. End-to-End Streaming

The overlay network is constructed for a wireless multihop network, where the topology resembles the underlying network. In this simulation, we constructed the overlay network in a 1 km * 1 km space, the nodes are randomly placed. Specifically, we generate m nodes and each node has up to four neighbors, where the neighbors are selected randomly from the nearby nodes. We consider two kinds of network, one with *fixed capacity*. All logical links are considered as symmetric and we set the capacity as 1 Mb/s for simplicity. Another one is a asymmetric network with *varied capacity* set as $500 + \text{rand}(1000)$ kb/s.

We first study the performance of a media streaming application. Consider all users provide streaming service and streaming demands are generated from selected nodes. We assume that the traffic are different quality of constant-bit rate (CBR) video clips and use $r = 100 + \text{rand}(300)$ kb/s to generate traffic rates. The source-destination pairs are selected so that the rate of traffic demands in each node does not exceed its capacity. We generate 100 topologies for each m and use the average data to reflect the performance.

We add two traffic management strategies for comparison. The first one we use is a fixed shortest path between each source and destination and forward all traffic through this path, named *shortest path*. The other is that in each node we try to distribute the traffic to the next hop equally among all possible shortest paths, named *equal loading multipath*. For example, if the node has two possible neighbors which can lead the traffic to the destination through shortest path, then both the neighbors will deliver half of the traffic. In this method, the traffic can be divided multiple times in the intermediate nodes to distribute the traffic to as many links as possible. In our experiments, the optimal traffic management is denoted by *optimal*, the MultiServ traffic management strategy is denoted by *MultiServ*, and the MultiServ with bounded complexity is denoted by *bounded MultiServ*.

Since the operations in MultiServ are performed in each node without knowledge of global traffic status in the overlay network, this encounters difficulty in the experiment in that input traffic needs to be known for each node. Fortunately, this can be obtained from the traffic output from a neighboring node. Specifically, in our experiments, we derive this under a simplifying assumption that links have similar delay, thus, the traffic can be delivered in multiple steps. The first step, the traffic flow out of the source and to its neighbor, named the first forwarder; the second step, the traffic will flow to the first forwarder's neighbor, the second forwarder, and so on until the traffic reaches its destination. At each step, the algorithm is running in each node according to the traffic input. We stop running the algorithm until the system enters a balanced state where the traffic flow in each node becomes stable and all the traffic reaches its destination.

We use *data loss rate* to compare the performance of different traffic management strategies. Data loss rate is defined as the proportion of data loss to total traffic. For example, if at a moment in the network the total traffic rate is 10 Mb/s and the total

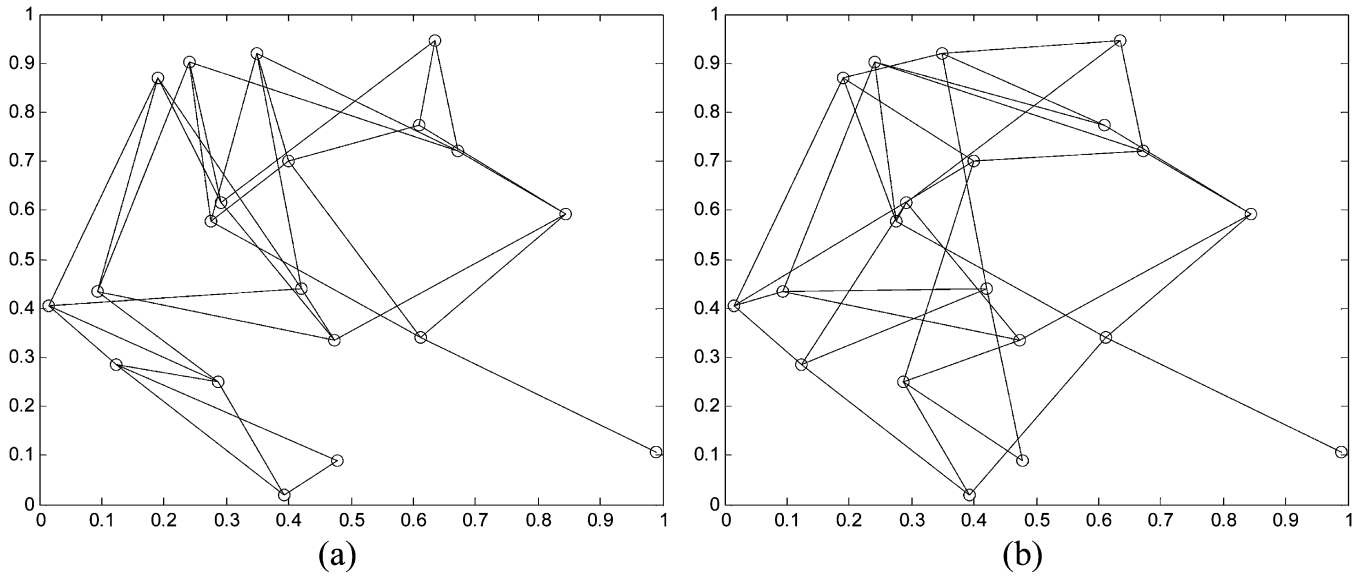


Fig. 8. Neighbor refinement in a 20 nodes graph in a 1000 * 1000 m space with 500 m radio range, after 32 refinements the average hop distance between pairs decrease from 2.66 to 2.17.

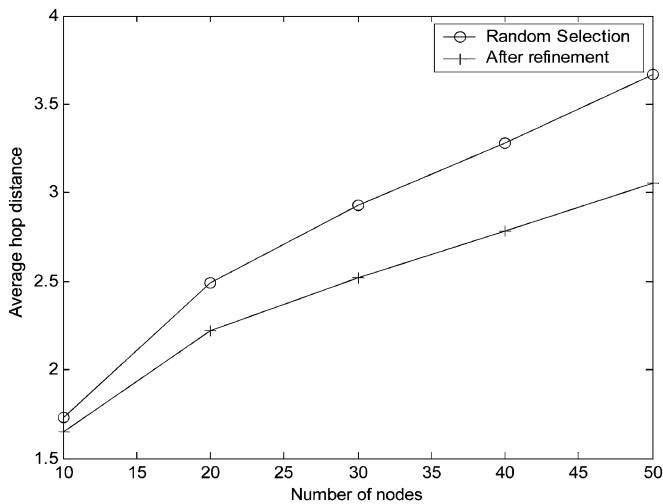


Fig. 9. Average hop distance: random neighbors and after refinements.

delivered traffic rate is 9 Mb/s, 1 Mb/s traffic is lost due to the lack of bandwidth, so the data loss rate is 10%.

We use Matlab V6.5 to simulate the overlay network and solve the optimization problems. In Figs. 10 and 11, we plot the data loss rate with various traffic demands for different traffic management strategies on a 20-node 80-link fixed capacity and varied capacity network, respectively. In these figures, traffic demands are labeled with P , the proportion of traffic to capacity, which is calculated as follows:

$$P = \frac{\sum \text{traffic rate} * \text{distance}(\text{source}, \text{destination})}{\sum \text{link capacity}}$$

It can be observed under a fixed capacity network, when the traffic increases, the data loss of equal loading multipath and shortest path method increases sharply, up to 20% using shortest path when the traffic load is 50%. While the MultiServ and bounded MultiServ method achieves similar result comparing with optimal method, the loss is less than 10%. Under a varied capacity network, the data loss is higher than that in

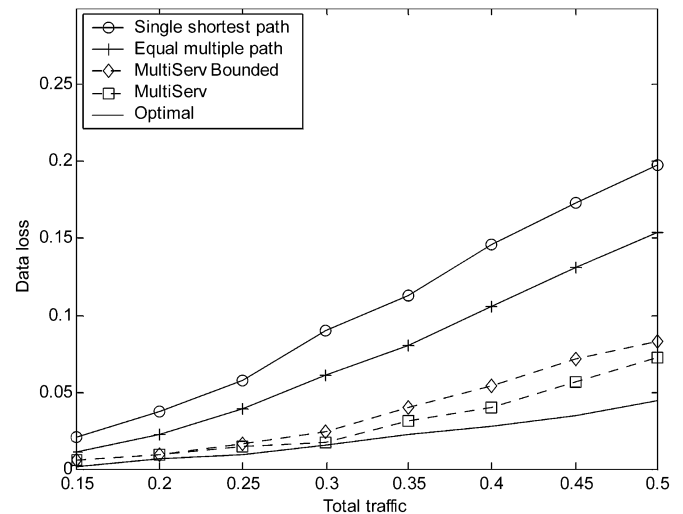


Fig. 10. Data loss rate with various traffic demands on a 20-node 80-link fixed capacity network.

a fixed capacity network with the same traffic load. MultiServ still demonstrates significantly performance gain with less than 1/2 of data loss rate compared with the shortest path and equal loading multipath schemes.

In Figs. 12 and 13, we plot the data loss rate with 40% traffic demands for different traffic management strategies on 10–50 nodes network with fixed capacities and varied capacities, respectively. From the figure, MultiServ and bounded MultiServ method has less than 10% data loss; while equal loading multipath and shortest path method has typically more than two times of data loss.

C. Application-Layer Multicast

In this simulation, we consider application-layer multicasting. There are g multicast groups where each group i has one source with u_i users and the streaming rate is r kb/s. For simplicity, each group forms a binary multicast tree. Each newly

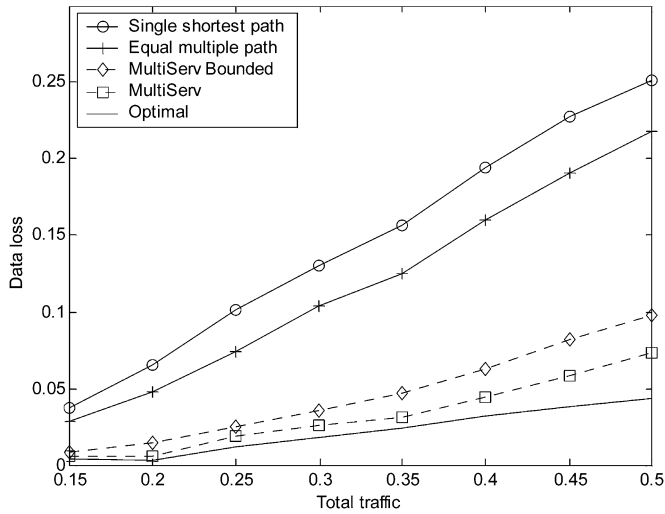


Fig. 11. Data loss rate with various traffic demands on a 20-node 80-link varied capacity network.

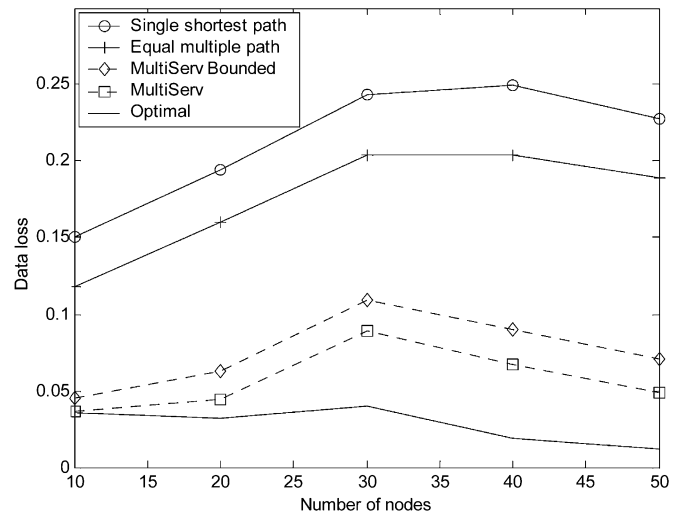


Fig. 13. Data loss rate with 40% traffic demands on 10-50 nodes network with varied capacities.

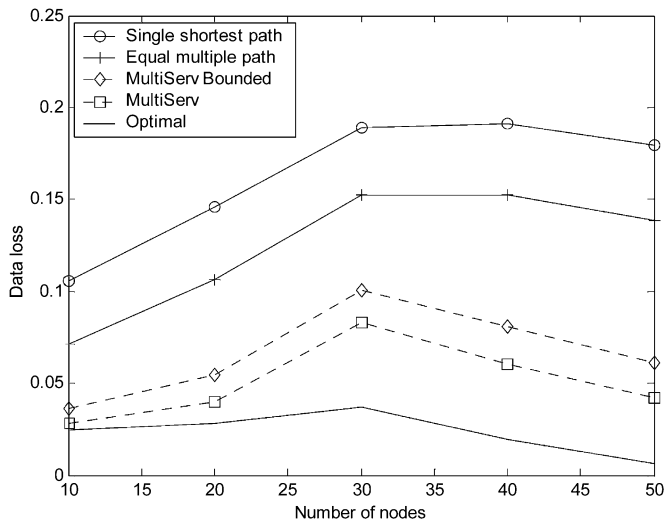


Fig. 12. Data loss rate with 40% traffic demands on 10-50 nodes network with fixed capacities.

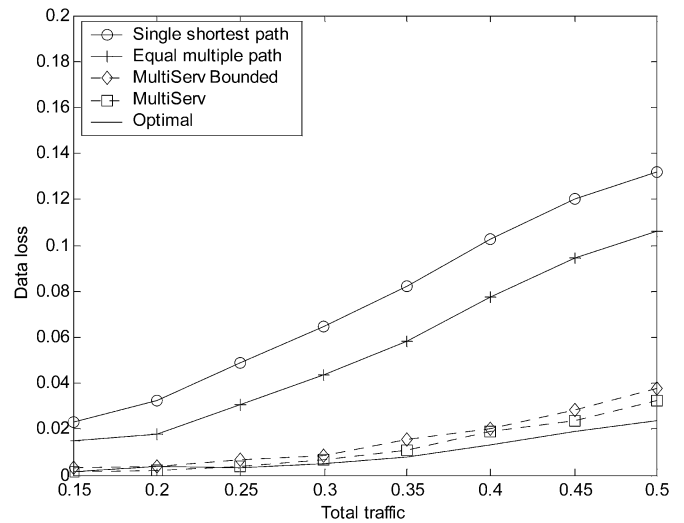


Fig. 14. Data loss rate with various multicast traffic demands on a 20-node 80-link fixed capacity network.

joined user selects an existing user in the tree where the user has less than two children and the hop distance is minimal in order to generate less traffic in the overlay network.

In this simulation, large overlay network is formed in order to see the performance of different bandwidth strategies in large scale. The formation of overlay network is still the same. Twenty multicast sources are randomly selected in the overlays, where the rates are set as $100 + \text{rand}(300)$ kb/s, the traffic demands are constructed using the following ways: a random user is selected and it will pick a random multicast group which will add a traffic request for the overlay network. It will be interested to consider the performance when the multicast content can be retrieved from multiple sources, but due to the limit of space, it will not be discussed in this paper.

In Figs. 14 and 15, we plot the data loss rate with various multicast traffic demands for different traffic management strategies on a 20-node 80-link fixed capacity and varied capacity network, respectively. The traffic load calculation is the same with previous one. In Figs. 16 and 17, we plot the data loss rate

with 40% multicast traffic demands for different traffic management strategies on 10-50 nodes network with fixed capacities and varied capacities, respectively. It can be seen that multicast traffic has less data loss than streaming traffic given the same traffic load. The reason is that the application-layer multicast has consideration of locality by finding nearby nodes as data source, which decrease traffic relaying. From Figs. 16 and 17, MultiServ strategies still has less than 1/2 of data loss compared with shortest path and equal loading multipath strategies.

It can be seen from all the above results that the MultiServ and bounded MultiServ methods achieve similar performance in all simulations, which are much better than equal loading multipath and shortest path methods. We have the data loss as follows.

- Data loss (Optimal).
- < Data loss (MultiServ).
- < Data loss (MultiServ Bounded).
- < Data loss (Equal Loading Multipath).
- < Data loss (Shortest Path).

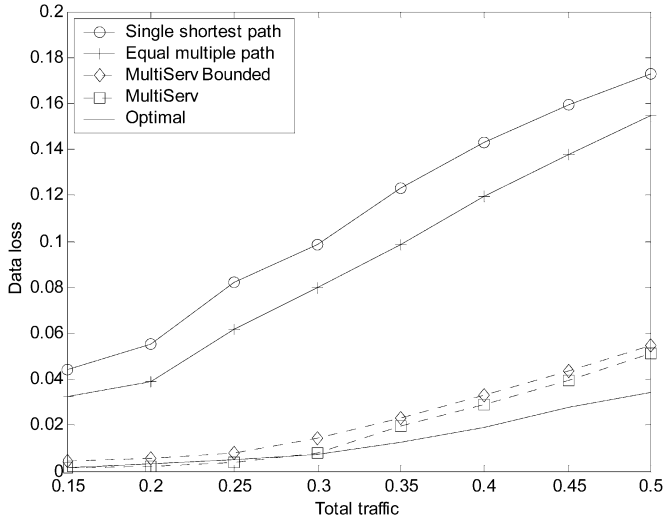


Fig. 15. Data loss rate with various multicast traffic demands on a 20-node 80-link varied capacity network.

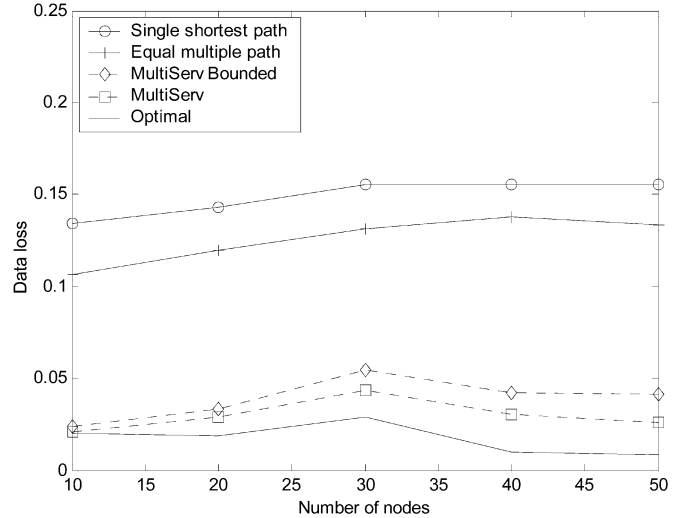


Fig. 17. Data loss rate with 40% multicast traffic demands on 10–50 nodes network with varied capacities.

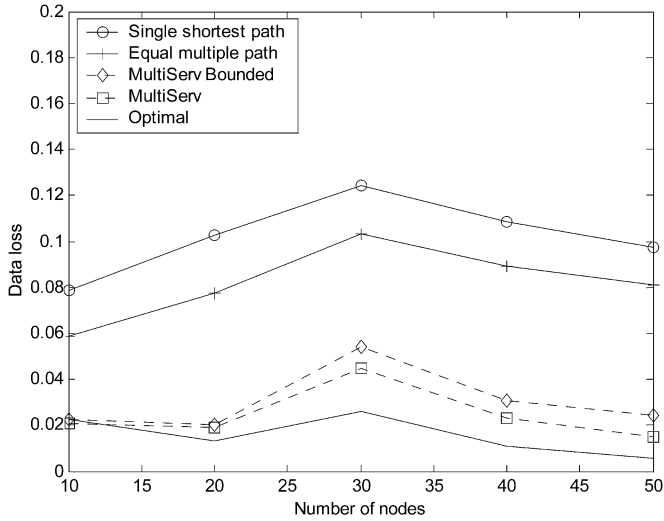


Fig. 16. Data loss rate with 40% multicast traffic demands on 10–50 nodes network with fixed capacities.

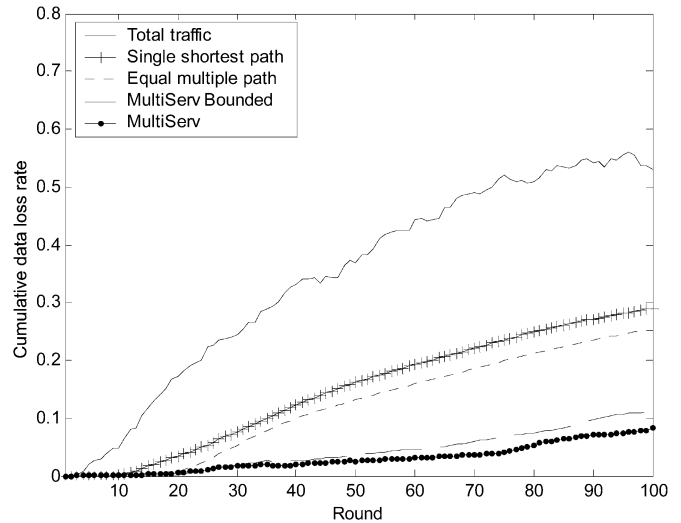


Fig. 18. Cumulative data loss rate with dynamic traffic (20 nodes, 80 links, varied capacities, 300 traffic volumes).

D. Dynamic Traffic

We also generate dynamic traffic to evaluate the proposed traffic management strategies. In dynamic traffic case, each traffic demand has a starting point and end point. For example, if a traffic demand with 100 kb/s rate from node 1 to node 10 has a starting point of 10 and end point of 20, the traffic will be launched in the tenth round in node 1, and it will continue and at the 20th round, the traffic stops. Fig. 18 plots the cumulative data loss rate. The solid line represents the proportion of traffic to total capacity. In Fig. 18, we can see the cumulative data loss of equal multiple path and shortest path strategies is higher than 20% with traffic increasing and in MultiServ the data loss is less than 10%.

E. Dynamic Capacities

The logical link capacities may change due to the interference or mobility in underlying network. The performance of

our traffic strategies are also been evaluated under such situation. In a 20-node 80-link overlay network, the link capacity is reset using $c = 500 + \text{rand}(1000)$ kb/s every five steps. The cumulative data loss rate is illustrated in Fig. 19. We can see that our traffic management strategies can adapt the varied capacity and perform the data forwarding quite well, it is observed that no links are overloaded and most traffic will safely reach its destination.

F. Combination Effects of Neighbor Refinement and MultiServ Traffic Management

Finally, we combine the techniques described in this paper and see how much performance gain we can achieve. We constructed a 20-node 80-link overlay network with random neighbors, say overlay A . The neighbor refinement algorithm is used in network A to form a new overlay B . To simplify, fixed capacity (1 Mb/s) is used for each link. We then use

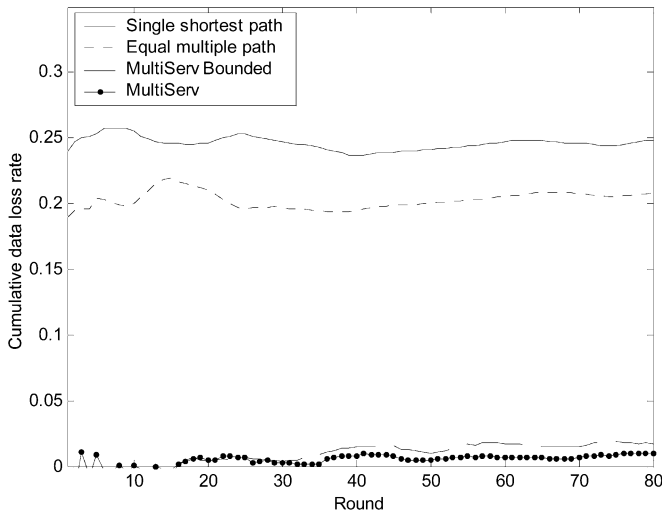


Fig. 19. Cumulative data loss rate with dynamic capacities (20 nodes, 80 links, 100 traffic volumes).

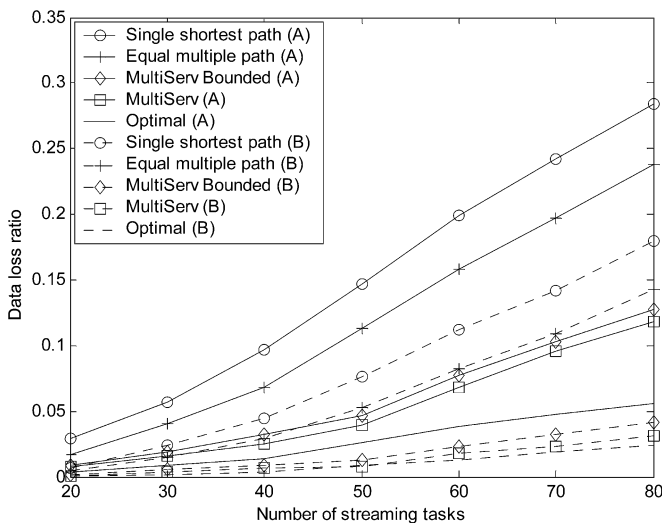


Fig. 20. Data loss rate using different traffic management strategy with different number of traffic demands on a 20-node 80-link with/without neighbor refinement.

our MultiServ traffic management methods in both network *A* and *B* to deliver streaming traffic described in Section V-B comparing with the data loss with shortest path and equal multiple path. We generate 100 overlay networks and calculate the average. The result is shown in Fig. 20.

From Fig. 20, we can see, given the same 80 streaming tasks, using MultiServ traffic management strategy with neighbor refinement, the data loss is less than 5%, comparing with more than 25% of data loss in shortest path routing. On the other hand, given data loss threshold of 5%, neighbor refinement + MultiServ can carry more than 80 streaming tasks comparing with less than 30 streaming tasks using simple shortest path routing. Clearly, the neighbor refinement + MultiServ techniques significantly improve the traffic relaying ability of the network.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a new architecture called MultiServ using the concept of overlay network, which can facilitate the deployment of new services in wireless multihop networks. We investigate traffic management problem, in particular, how to route large volume of traffic, and we show from extensive performance studies that the proposed solution can significantly outperform the existing approaches. There are several unique advantages in MultiServ architecture.

Scalability and stability: The overlay network uses a construction algorithm that provides more stability and can be easily extended to large scale; the proposed traffic scheme can also work in wireless networks with different size and varied link capacities.

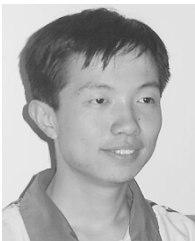
High performance and better service deployment: The aggregate flow scheme enables better control on traffic flows. MultiServ facilitates the deployment of services that may not be easy to deploy or have poor performance in wireless networks, such as media streaming and multicasting.

We are currently carrying out experiments using a wireless multihop testbed in Microsoft, where the main focus is to investigate the performance under a more realistic environment. Another possible avenue for further study is to examine the application-layer multicast based on the proposed MultiServ architecture.

REFERENCES

- [1] H. Balakrishnan, H. Rahul, and S. Seshan, "An integrated congestion management architecture for Internet hosts," in *Proc. SIGCOMM*, 1999, pp. 175–187.
- [2] Y. Wang, Z. Wang, and L. Zhang, "Internet traffic engineering without full mesh overlaying," in *Proc. IEEE INFOCOM*, 2001, pp. 565–571.
- [3] A. Sridharan, R. Guerin, and C. Diot, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," in *Proc. IEEE INFOCOM*, 2003, pp. 1167–1177.
- [4] D. Awduche *et al.*, "Overview and principles of Internet traffic engineering," IETF, RFC3272, May 2002.
- [5] A. Srinivas and E. Modiano, "Minimum energy disjoint path routing in wireless ad hoc networks," in *Proc. ACM MobiCom*, 2003, pp. 122–133.
- [6] D. S. J. De Douglas, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multihop wireless routing," in *Proc. ACM MobiCom*, 2003, pp. 134–146.
- [7] S. Yi, Y. Pei, and S. Kalyanaraman, "On the capacity improvement of ad hoc wireless networks using directional antennas," in *Proc. ACM MobiHoc'03*, pp. 108–116.
- [8] C. Peraki and S. D. Servetto, "On the maximum stable throughput problem in random networks with directional antennas," in *Proc. ACM MobiHoc*, 2003, pp. 76–87.
- [9] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. ACM MobiCom*, Jul. 2001, pp. 61–69.
- [10] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, pp. 388–404, Mar. 2000.
- [11] H. Luo, S. Lu, and V. Bharghavan, "A new model for packet scheduling in multihop wireless networks," in *Proc. ACM MobiCom*, 2000, pp. 76–86.
- [12] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE Trans. Netw.*, vol. 10, no. 4, pp. 477–486, 2002.
- [13] M. Gastpar and M. Vetterli, "On the capacity of wireless networks: The relay case," in *Proc. IEEE INFOCOM*, Jun. 2002, pp. 1577–1586.

- [14] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1488–1505, Aug. 1999.
- [15] C. R. Lin, "On-demand QoS routing in multihop mobile networks," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 22–26, 2001, pp. 1735–1744.
- [16] W. H. Liao, Y. C. Tseng, S. L. Wang, and J. P. Sheu, "A multipath QoS routing protocol in a wireless mobile ad hoc network," *Telecommun. Syst.*, vol. 19, no. 3–4, pp. 329–347, 2002.
- [17] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Pre-emptive routing in ad hoc networks," in *Proc. ACM/IEEE MobiCom*, Jul. 2001, pp. 43–52.
- [18] C. Zhu and M. S. Corson, "QoS routing for mobile ad hoc networks," in *Proc. IEEE INFOCOM*, Jun. 2002, pp. 958–967.
- [19] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, pp. 153–181, 1996.
- [20] V. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. INFOCOM*, Kobe, Japan, Apr. 1997, pp. 1405–1413.
- [21] C. Perkins, E. M. Royer, and S. R. Das, "Ad hoc on-demand distance vector routing," IETF, RFC 3561.
- [22] C. Gui and P. Mohapatra, "Efficient overlay multicast for mobile ad hoc networks," in *Proc. WCNC*, 2003, pp. 1118–1123.
- [23] C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM*, 1994, pp. 234–244.
- [24] Y.-C. Hsu, T.-C. Tsai, Y.-D. Lin, and M. Gerla, "Bandwidth routing in multihop packet radio environment," in *Proc. 3rd Int. Mobile Comput. Workshop*, 1997.
- [25] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: A core-extraction distributed ad hoc routing algorithm," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 202–209.
- [26] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1360–1369.
- [27] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically aware overlay construction and server selection," in *Proc. IEEE INFOCOM*, New York, 2002, pp. 1190–1199.
- [28] X. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu, "A construction of locality-aware overlay network: mOverlay and its performance," in *IEEE J. Sel. Areas Commun. (Special Issue on Recent Advances on Service Overlay Networks)*, Jan. 2004, pp. 18–28.
- [29] Meshdynamics. [Online]. Available: <http://www.meshdynamics.com/>
- [30] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. 18th ACM SOSP*, Banff, Canada, Oct. 2001, pp. 131–145.
- [31] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz, "OverQoS: An overlay based architecture for enhancing Internet QoS," in *Proc. 1st Symp. Networked Syst. Design Implem. (NSDI)*, San Francisco, CA, Mar. 2004, pp. 71–84.



Xinyan Zhang (S'03) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 2001 and the M.Phil. degree from the Department of Information Engineering, Chinese University of Hong Kong, Shatin, in 2004. He is currently working towards the Ph.D. degree at the Hong Kong University of Science and Technology, Kowloon.

His research interests include modeling, analysis and measurement on Internet, especially overlay network.



Qian Zhang (M'00) received the B.S., M.S., and Ph.D. degrees from Wuhan University, Wuhan, China, in 1994, 1996, and 1999, respectively, all in computer science.

She joined Microsoft Research Asia, Beijing, China, in July 1999, as an Associate Researcher in the Internet Media Group and now is a Researcher of the Wireless and Networking Group. She has published over 60 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, and overlay networking. She is the inventor of more than a dozen pending patents. Her current research interest includes multimedia delivery over wireless Internet, next-generation wireless networks, P2P network/ad hoc network. Currently, she is participating many activities in the IETF Robust Header Compression (ROHC) WG group and is the principal contributor of the IETF ROHC WG draft on TCP/IP header compression.



Bo Li (S'89–M'92–SM'99) received the B.Eng. (*summa cum laude*) and M.Eng. degrees in the computer science from Tsinghua University, Beijing, China, in 1987 and 1989, respectively, and the Ph.D. degree in the electrical and computer engineering from the University of Massachusetts at Amherst, in 1993.

From 1993 to 1996, he worked on high-performance routers and ATM switches at IBM Networking System Division, Research Triangle Park, NC. Since 1996, he has been with the Department of Computer Science, Hong Kong University of Science and Technology, Kowloon. Since 1999, he has also held an Adjunct Researcher position at Microsoft Research Asia (MSRA), Beijing, China. His current research interests are on adaptive video multicast, packet scheduling and dynamic routing in optical networks, resource management in mobile wireless systems, scheduling and energy efficient routing in ad hoc networks, across layer design for sensor networks, and content distribution and replication. He has published 80 journal papers and held several patents in above areas.

Dr. Li has been on the Editorial Board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *ACM/Kluwer Journal of Wireless Networks (WINET)*, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC)—Wireless Communications Series, *ACM Mobile Computing and Communications Review (MC2R)*, *Elsevier Ad Hoc Networks*, *SPIE/Kluwer Optical Networking Magazine (ONM)*, and *KICS/IEEE JOURNAL ON COMMUNICATIONS AND NETWORKS (JCN)*. He served as a Guest Editor for the *IEEE Communications Magazine* (Special Issue on Active, Programmable, and Mobile Code Networking, April 2000), *ACM Performance Evaluation Review* (Special Issue on Mobile Computing, December 2000), *SPIE/Kluwer Optical Networks Magazine* (Special Issue in Wavelength Routed Networks: Architecture, Protocols and Experiments, January/February 2002), the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Special Issues on Protocols for Next-Generation Optical WDM Networks, October 2000, Recent Advances in Service-Overlay Network, January 2004, Quality-of-Service Delivery in Variable Topology Networks, September 2004), and *ACM/Kluwer Mobile Networks and Applications (MONET)* (Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks, 2nd Quarter of 2005). In addition, he has been involved in organizing over 40 conferences, especially, IEEE INFOCOM since 1996. He was the Co-TPC Chair for IEEE INFOCOM 2004.



Wenwu Zhu (S'92–M'97–SM'01) received the B.E. and M.E. degrees from National University of Science and Technology, Hefei, China, in 1985 and 1988, respectively, the M.S. degree from Illinois Institute of Technology, Chicago, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 1993 and 1996, respectively, all in electrical engineering.

From 1988 to 1990, he was with the Graduate School, University of Science and Technology of China (USTC), and Chinese Academy of Sciences (Institute of Electronics), Beijing, China. Since September 2004, he has been with the Communication Technology Laboratory, China, as Co-Director. Prior to his current post, he was with Microsoft Research Asia first as a Researcher in the Internet Media Group and later as Research Manager of the Wireless and Networking Group. From 1996 to 1999, he was with Bell Laboratories, Lucent Technologies, NJ, as a Member of Technical Staff during 1996–1999. He has published over 180 refereed papers in the areas of wireless/Internet multimedia delivery, and wireless communications and networking. He participated activity in the IETF ROHC WG on robust TCP/IP header compression over wireless links. He is co-inventor of over 20 pending patents. His current research interest is in the area of wireless communication and networking, and wireless/Internet multimedia communication and networking.

Dr. Zhu is a member of Eta Kappa Nu, the Multimedia System and Application Technical Committee and Life Science Committee of the IEEE Circuits and Systems Society, and the Multimedia Communication Technical Committee of the IEEE Communications Society. He received the Best Paper Award in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2001. He has been on various editorial boards of IEEE journals such as Guest Editor for the PROCEEDINGS OF THE IEEE, Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. Currently, he is also the Chairman of the IEEE Circuits and System Society Beijing Chapter and the Secretary of Visual Signal Processing and Communication Technical Committee.



Tak-Shing Peter Yum was born in Shanghai. He received the B.S., M.S., M.Ph., and Ph.D. degrees from Columbia University, New York, in 1974, 1975, 1977, and 1978, respectively.

He joined Bell Telephone Laboratories in April 1978 working on switching and signaling systems. Two and a half years later, he accepted a teaching appointment at the National Chiao Tung University, Hsinchu, Taiwan. He stayed there for two years before joining the Chinese University of Hong Kong, Shatin, in 1982, where he is now Dean of Engineering and Professor of Information Engineering. His diverse industrial experience includes Bell Labs, Bellcore (now Telcordia), IBM Research, Motorola Semiconductors and ITRI of Taiwan, SmarTone Communications and Radio-Television, Ltd., Hong Kong. He has also lectured extensively in major universities in China and was appointed Adjunct Professors at South East University and Huazhong University of Science and Technology. He has published widely in Internet research with contributions to routing, buffer management, deadlock handling, message resequencing, and multiaccess protocols. He then branched out to work on cellular network, lightwave networks, and video distribution networks. His recent works are on 3G and IP networks. He enjoys doing research with students. Eight of his graduates are now professors at local and overseas universities.

Dr. Yum is on the editorial board of six international journals on communications and information science, including the IEEE TRANSACTIONS ON COMMUNICATIONS. He was formerly an Editor of the IEEE TRANSACTIONS ON MULTIMEDIA.