This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2006 proceedings.

The Top Load Balanced Forest Routing in Mesh Networks

Yean-Fu Wen and Frank Yeong-Sung Lin
Department of Information Management
National Taiwan University (NTU)
{d89002; yslin}@im.ntu.edu.tw

Abstract—**Public wireless local area networks (PWLAN), which provide last-mile connectivity to the Internet, are popular worldwide, especially in heavily populated cities. Traditional ad hoc shortest path routing algorithms, such as AODV and DSR, focus on minimum hops that cause traffic to concentrate on some TAPs, while others are light. Thus, the major issue addressed this paper is how to cluster backbone mesh networks efficiently so that routing is concentrated on given gateways. We formulate the problem as an integer programming problem with minimal routing traffic as the objective function, subject to the top load balancing and link capacity. We propose a greedy algorithm, called Greedy Load Balancing Routing (GLBR), to solve this problem and evaluate it by the Lagrangian Relaxation approach to quantify the objective value correctly. The experimental results show that the algorithm achieves near-optimization, and obtains a gap smaller than 5% and 10% in grid-based and random-based architectures, respectively.**

I. INTRODUCTION

It is now possible to retrieve data services anywhere and anytime via public wireless local area networks (PWLAN), which are an inexpensive means of providing last-mile connectivity to the Internet. It is now possible to retrieve data services anywhere and anytime via public wireless local area networks (PWLAN), which are an inexpensive means of providing last-mile connectivity to the Internet. The use of mesh networks on the last-hop reduces the major deployment and maintenance costs of wired infrastructures, such as Hot Spot, and thereby cuts overall ISP costs. Thus, some Transit Access Points (TAPs) [8] are wired to the Internet, while the others access the Internet through these wire-connected nodes by forming a multi-hop wireless mesh network.

As the TAPs have fixed positions (e.g., on traffic lights), the routing complexity of mobility is reduced. However, a TAP not only forwards Mobile Host (MH) data, but also acts as a router to forward other TAPs' data. Traditional ad hoc shortest path routing algorithms, such as AODV and DSR, focus on minimum hops that cause traffic to concentrate on some TAPs, while others are light. Namely, the algorithms that aim to minimize the number of hops may ignore "fairness" in routing. For example, the shortest path routing is likely to use the same set of hops to relay packets for the same original destination (OD) pair. This will overload the nodes on the path, even though there are other feasible paths. Such an uneven use of the nodes may cause some TAPs to be busy forwarding packets, which will reduce the bandwidth available to their MHs.

The issue of fairness and end-to-end performance in multi-hop wireless mesh networks by removing spatial bias and maximizing spatial reuse were studied by [8] and [9]. R. Karrer et al. addressed the fairness problem, while V. Gambiroza et al. developed a simple layer-2 Inter-TAP Fairness Algorithm (IFA) to achieve the objective. They address the upstream fair issues by controlling the transmission time for a single branch to avoid the starvation problem of downstream TAPs. In this paper, we focus on the load balancing for a "load-balanced forest" with an apposite number of the gateways.

Load balancing helps avoid bottlenecks in a network and increases the network's resource utilization. Some load balancing routing research focuses on multi-path routing to cope with congestion and link breakage [5], [3]. Meanwhile, other studies focus on single gateway load balancing distributed on different interfaces [6], [2]. Here, we adopt the same fairness index equation and support multiple backhauls load balancing in a wireless mesh network.

The "top load-balanced forest" means there are many egresses as cluster heads in the mesh networks. A "top load-balanced forest" is a backbone forest for a set of load such that all the branches, which are the closest links to the gateways, carry near equivalent amounts of traffic.

We formulate the problem as an optimization problem to obtain minimal aggregated flow of each used links, subject to the number of egress nodes, the fairness index, and the capacity constraints. Our proposed routing algorithm, GLBR, is effective in dealing with this complex optimization problem. In further computational experiments, the LR approach is used to fulfill the timing and the quality requirements of optimal decisions. Through this approach, which has been successfully adopted to solve many famous NP-complete problems [4], we can derive the lower bound (LB) to evaluate the gap between the proposed algorithm value and the optimization solution.

The remainder of this paper is organized as follows. In Section II, we describe the top load-balancing forest problem by mathematical formulation. In Section III, the detailed procedures of the optimal top load-balancing forest routing algorithm are described. In Section IV, we describe the evaluations, which are bounded by the LR approach, and the experiment results to demonstrate that our approach finds a near-optimal solution. Finally, in Section V, we present our conclusions.

II. PROBLEM FORMULATION

1-4244-0086-4/06/$20.00 ©2006 IEEE.

The problem is modeled as a graph, $G(V, L)$, where V are vertices representing TAPs distributed on a two-dimensional plane (X_AXIS, Y_AXIS) and L denote links between any two TAPs within the transmission range. The number of egress nodes and positions are given, and each wireless node has a directional antenna. Accordingly, a mathematical model is then developed to deal with the problem as a top load-balanced forest routing problem in minimize the total number of flows in a mesh network. TABLEs 1 and 2 list the given parameters and the decision variables, respectively. The problem is formulated as the following integer programming problem.

Objective function

$$Z_{IP} = \min \sum_{l \in L} f_l \,, \tag{IP}$$

subject to

$$\sum_{g \in G} z_{gv} = 1 \qquad \forall v \in V \tag{1}$$

$$\sum_{l \in L} y_l = |V| - |G| \tag{2}$$

$$z_{gv} \le \sum_{p \in P_{gv}} x_{gpv} \qquad \forall g \in G, v \in V \tag{3}$$

$$\alpha \le \frac{(\sum_{l \in g^-} f_l)^2}{E_g \sum_{l \in g^-} f_l^2} \qquad \forall g \in G \tag{4}$$

$$\sum_{p \in P_{gv}} x_{gpv} \delta_{pl} \le y_l \qquad \forall g \in G, v \in V, l \in L \tag{5}$$

$$\sum_{l \in v^+} y_l = 1 \qquad v \in V - G \tag{6}$$

$$\sum_{l \in g^+} y_l = 0 \qquad \forall g \in G \tag{7}$$

$$\sum_{g \in G} \sum_{v \in V} \sum_{p \in P_{gv}} x_{gpv} \delta_{pl} \le f_l \qquad \forall l \in L \tag{8}$$

$$f_l \le C_l \qquad \forall l \in L \tag{9}$$

$$\sum_{g \in G} \sum_{p \in P_{gv}} x_{gpv} = 1 \qquad \forall v \in V \tag{10}$$

$$x_{gpv} = 0 \text{ or } 1 \qquad \forall p \in P_{gv}, g \in G, v \in V \tag{11}$$

$$y_l = 0 \text{ or } 1 \qquad \forall l \in L \tag{12}$$

$$z_{gv} = 0 \text{ or } 1 \qquad \forall g \in G, v \in V \tag{13}$$

$$f_l \ge 0 \qquad \forall l \in L \,. \tag{14}$$

The objective of this problem is to find the minimum total number of link flows, subject to:

Constraint (1): Every TAP, v, must select a gateway, g, as an egress point to connect to the Internet.

Constraint (2): This is a tree constraint, comprised of a set of TAPs, v, and gateways, g, which must be equal to the number of used links larger than the number of TAPs minus the number of gateways.

TABLE 1. NOTATIONS FOR THE GIVEN PARAMETERS

Notation	Descriptions
G	The set of candidate gateways, where $g \in G$.
L	The set of links, where $l \in L$.
V	The set of TAPs, where $v \in V$.
P_{gv}	The set of paths from the original node v to destination node g or vice versa.
C_l	The given capacity of a link l.
δ_{pl}	The indication function, which denotes link l on a path p.
E_g	The number of out-degrees of a gateway g.
α	The fairness index value, which is equal to 1 - ε, where ε is a small value.
g^-	The incoming link of a gateway g.
V^+, g^+	The outgoing link of a TAP node v and a gateway g.

TABLE 2. NOTATIONS FOR THE DECISION VARIABLES

Notation	Descriptions
x_{gpv}	1 if the path p from a node v to a gateway g is selected; otherwise, 0.
z_{gv}	The TAP v connected to the wired network via the gateway g.
y_l	1 if the link l is used to connect two nodes; otherwise, 0.
f_l	The amount of flow via link l to a gateway.

Constraint (3): Once a TAP, v, selects a gateway, g, as its egress point, a path from v to g must be determined.

Constraint (4): This constraint iterates the link flows into a fairness index equation to achieve the given fairness value α (i.e., an α value approaching 1 indicates the loads are most balancing distributed on each gateway g's adjacent links).

Constraint (5): Once the path p is selected and the link l is on the path, then the value of the decision variable, y_l, must be set to 1.

Constraint (6): A tree constraint that limits the number of out degrees to 1.

Constraint (7): The number of out degrees of a gateway is equal to 0.

Constraint (8): Aggregates the number of flows via gateway g's out-degree link l.

Constraint (9): The capacity constraint, which limits the aggregate flow not larger than the given capacity, C_l.

Constraint (10): The number of paths from each TAP v to the selected gateway g is equal to 1.

Constraints (11)~(13): The values of decision variables z_{gv}, x_{gpv}, and y_{gl} are limited to 0 or 1.

Constraint (14): The aggregate flow, f_l, of each link must larger than 0.

III. THE PROPOSED ALGORITHM

We propose a greedy approach, called Greedy Load-balanced routing (GLBR), which adds one link and connects one TAP to a given gateway per iteration. Initially, we set the cost of each link to 1 and mark all TAPs, $|V| - |G|$, as FALSE. We then select the node with the lowest cost, which also has the fewest out-degrees and the current total traffic flow of its candidate egress link is minimum, into the forest dominant set T. Once the link is selected, the cost of the out degree to the previous node's cost plus 1 in order to balance the traffic flow of each branch. The aggregate flow of the nodes, v, belonging to the selected path is also increased by 1. We repeat the above procedure until all v are marked and belong to the forest dominant set T.

```
Algorithm GLBR (B, V, L)
Input: G (V, L) (a directed graph, where v ∈ V, and (u, v) ∈ L) and a set of gateways G where g ∈ G. {All link costs (u, v) are assumed to be
        nonnegative.}
Output: The nodes of the mesh network routing tree are included in the dominant set T. the variable v.pred marks the previous node of each relay node v
        to a gateway b.
begin
   for all vertices v do
      v.mark := FALSE;      v.SP := INFINITE;
   end-for.
   for all gateway b do
      T := T ∪ b;           b.pred := NULL;
   end-for.
   while all nodes, v, are not in T do
      for each node, u, in T do
            find the minimum cost node v that is not included in the dominant set T. Here, we also check the capacity constraint (9) around the path p_bv. If
            more than two links (uv) have the same cost, we select the minimal branch aggregated flow of gateway u or the minimal number of out
            degrees of node v.
      end-for.
      T := T ∪ v;           v.pred ;= u;
      for node w, which is not in T, have the direct link to v do
         the previous link cost plus 1 to link cost(v, w);
      end-for.
      while v.pred is not a gateway do
         increase the link capacity to 1 along the selected path from node v.
      end-while.
   end-while.
end.
```

Figure 1. The pseudo code of our propose GLBR algorithm.

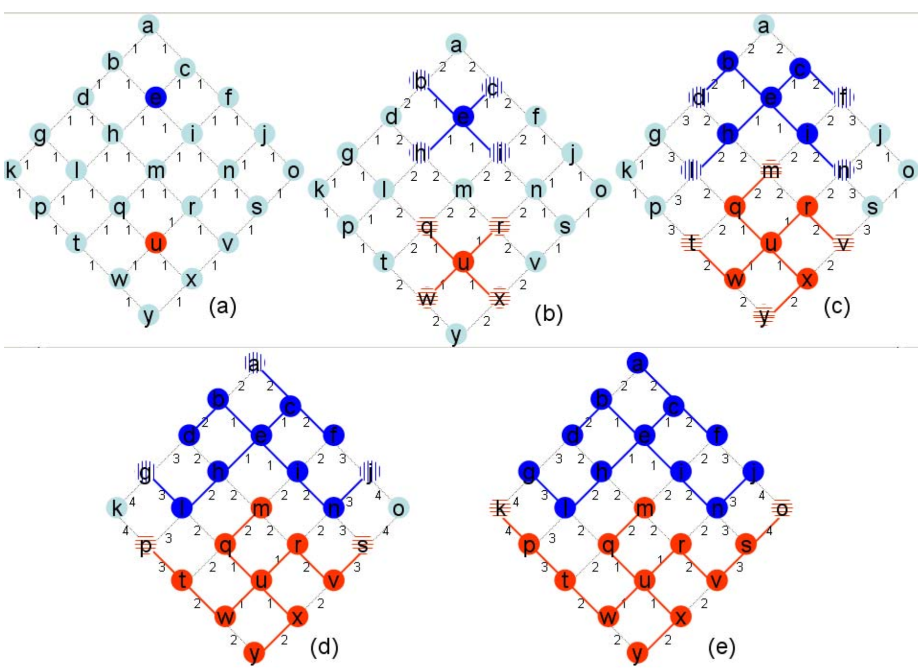

Figure 2. An example of GLBR algorithm with two gateways, e and u. Initially in (a), e and u are included in the set $T = \{e, u\}$ and all links are set to 1. (b) All links adjacent to gateways are selected and the adjacent link costs are increased by 1 per iteration. (c) Select the minimal link cost with minimal capacity of gateway branches. Thus, nodes y, v, t, m, n, l, f, and d are selected. Update the related link cost and capacity per iteration, too. $T = \{b, c, d, e, f, h, i, l, m, n, q, r, t, u, v, w, x, y\}$. (d) Execute the same procedures and select the links with the minimal cost (i.e., 3) and minimal branch capacity. Nodes a, g, j, p, and s are selected. The relative link costs and capacity are updated. Finally, the remaining nodes, k and o, are selected that shows in (e). The objective value is 46 and $FI = 0.93$.

Fig. 1 shows the pseudo code for the GLBR algorithm, which extends the concept of Prim's minimum cost spanning tree algorithm, to achieve the minimum flow of a top load balanced forest in a mesh networks. In the code, line 13 ("while-loop") ensures that each node not included in the gateways is selected to fulfill Constraints (1) and (2) that connect to a gateway. Lines 14-17 restrict a TAP to selecting one previous node to fulfill Constraints (3), (6), and (10), while line 14 fulfills Constraint (7). Lines 20-22 and 23-25 increase the link cost and capacity per iteration in order to fulfill the load balancing Constraint (4).

Fig. 2 shows a GLBR algorithm example with two gateways in the center of the graphs, to illustrate how the algorithm works. In this figure, the algorithm starts from the gateways and their out-degree links, e.g., link $\overline{eb}$, $\overline{ec}$, $\overline{eh}$, and $\overline{ei}$, and $\overline{uq}$, $\overline{ur}$, $\overline{uw}$, and $\overline{ux}$ for gateways *e* and *u*, respectively. After update the adjacent links cost capacity, e.g., increase link cost of $\overline{ba}$, and $\overline{bd}$ of node *b* and increase the capacity along path from each node to its selected gateway, e.g., path {*b*, *e*} with capacity 1 in (b). We get the result in (c) in the same way. We then select the remaining adjacent nodes with the minimal link cost and small capacity of a selected branch, such as link $\overline{ca}$ with capacity 2 and link cost equal to 2. But, we can not select link $\overline{fj}$ because its link cost is 3. Accordingly, we select nodes g, p, and s per iteration in (d). Finally, the routing path for two gateways is constructed iteration by iteration, as shown in (e).

IV. Evaluation and Experiment

For wireless mesh networks, the top load-balanced forest problem is solved by the algorithm described in the previous paragraph. An LR-based approach is adopted to obtain the lower bound (LB) and evaluate the proposed algorithm. In the following, illustrating the process by which the algorithms arrive at solutions for a top load-balanced routing forest

4.1. Lagrange Relaxation (LR)

An LR-based approach to solve large-scale integer programming problems, first used in the 1970s [4]. In brief, it is a flexible solution strategy that permits us to exploit the fundamental structure of possible optimization problems by relaxing complicated constraints into the objective function with Lagrangean multipliers [1], [4]. Accordingly, the primal optimization problem is transformed into a dual-mode problem. Furthermore, we decompose complex mathematical models into stand-alone sub-problems and use a proper algorithm to optimally solve each sub-problem. By the properties of decomposition, which proved in [1], [4], it can effectively reduce the complexities and difficulties compared to the original problem (IP).

Before transforming the above primal problem (IP) into a dual mode problem (D), let $\beta_g = \sum_{l \in g^-} f_{gl}$ and iterate it into Constraint (4). The denominator is exchanged to the left hand side and gives the following three equations.

$$\beta_g = \sum_{l \in g^-} f_l \qquad \forall g \in G \qquad (4\text{-}1)$$

$$\alpha E_g \sum_{l \in g^-} f_l^2 \le \beta_g^2 \qquad \forall g \in G \qquad (4\text{-}2)$$

$$0 < \beta_g < \sum_{l \in g^-} C_l \qquad \forall g \in G \qquad (4\text{-}3)$$

Then, Constraints (3), (4-1), (4-2), (5), and (8) are relaxed. For a vector of non-negative Lagrangean multipliers, μ_{gv}^1, μ_g^2, μ_g^3, μ_{gvl}^4, and μ_l^5, the problem (D) is given by:

Dual mode problem:

$$\begin{aligned} Z_D = \min & \sum_{l \in L} f_l \\ & + \sum_{g \in G} \sum_{v \in V} \mu_{gv}^1 \left(z_{gv} - \sum_{p \in P_{gv}} x_{gpv} \right) \\ & + \sum_{g \in G} \mu_g^2 \left(\alpha E_g \sum_{l \in g^-} f_l^2 - \beta_g^2 \right) \\ & + \sum_{g \in G} \mu_g^3 \left(\beta_g - \sum_{l \in g^-} f_l \right) \\ & + \sum_{g \in G} \sum_{v \in V} \sum_{l \in L} \mu_{gvl}^4 \left(\sum_{p \in P_{gv}} x_{gpv} \delta_{pl} - y_l \right) \\ & + \sum_{l \in L} \mu_l^5 \left(\sum_{g \in G} \sum_{v \in V} \sum_{p \in P_{gv}} x_{gpv} \delta_{pl} - f_l \right) \end{aligned} \qquad \text{(D)}$$

subject to: (1), (2), (6), (7), (9), (10), (11), (12), (13), and (14).

To solve this problem, we decompose (LR) into five independent and solvable sub-problems. The summation of the values of the five sub-problems is the lower bound (LB), while the value of our proposed algorithm is an upper bound (UB) of the problem (IP). The distance between LB and UB, computed by (*UB* - *LB*) / *LB* * 100%, illustrates the optimality of the problem solution.

The LR-based approach ensures the optimization results between the UB and LB, so we keep the gap as small as possible in order to enhance our solution quality and achieve near optimization. Fig. 3 shows a simple LR experimental result where the LB, obtained from the dual-mode problem (D), quickly reaches the expected near optimal value after about 1,300 iterations. The gap in this experiment is about 3.5%.

4.2. Experimental results

We distribute a set of nodes, *V,* in two cases: 1) grid-based graphs; and 2) in a uniformly random-based fashion with a density of one node per 1*1 area. Then, we uniformly assign the TAPs, *v*, as gateways, *g*. Each node has a maximum transmission range, *R* = 1. The relative parameter configurations are listed in TABLE 3. For uniformly random deployment, the network connectivity is only a function of the average number of neighbors. Here, we compare the following two conditions: (i) varying the numbers of nodes; and (ii) varying the number of gateways.

Fig. 4 shows grid-based experimental results for Case (1) with 1, 2, and 4 gateways. As the number of nodes increases, the normalized numbers of flows also increases. The UB-*x* curves (where *x* means the number of gateways) is calculated by the proposed algorithm to solve the primal problem (P), while the LB-*x* curves denote the LB value, which is generated by the dual mode problem (D) solved by the LR approach. The gap between UB and LB is less than 5%, which means the optimal solution is guarantee in between the UB and LB.

Fig. 5 shows the random-based experimental results for Case (2) with 1, 2, and 4 gateways. As the number of nodes increases, the normalized number of flows also increases. In this case, the gap is about 10%, which is larger than the grid-based case because the gateways have variance numbers of degrees and the fairness index constraint (4) is relaxed. However, this does show that our proposed algorithm gets the objective value within 10% of the optimization value.

V. CONCLUSIONS

In this paper, we have defined the "top load-balanced forest" and described the problem by mathematical formulation. We have also proposed a GLBR algorithm to handle the problem. The algorithm handles the fairness, capacity limit, and TAP assignment issues simultaneously. It not only achieves the minimum objective function value and top load balancing of the gateways' branch, but also gets equivalent number of nodes between gateways with shorter path. We evaluated the proposed algorithm by an LR-based approach. The experimental results show that the proposed algorithm gets a near optimal solution with a gap less than 5% and 10% for grid-based and random-based topologies, respectively. The time complexity is $O(|V|^2 \cdot E_v)$.

REFERENCES

[1] Ahuja R. K., T. L. Magnanti, and J. B. Orlin, *Network Flows:Theory, Algorithms, and Applications*, Ch. 4 and Ch 16, Prentice-Hall, 1993.

[2] H. Dai, R. Han, "A node-centric load balancing algorithm for wireless sensor networks," IEEE GLOBECOM - Wireless Communications 2003, V(1):1-5, pp. 548-552.

[3] L. Zhang, Z. Zhao, Y. Shu, L. Wang, O.W.W Yang, "Load balancing of multipath source routing in ad hoc networks," Communications, 2002. ICC 2002. IEEE International Conference on, 5(28) April-2 May 2002, pp. 3197-3201.

[4] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 27, no. 1, 1981, pp. pp. 1-18.

[5] M. Pearlman, Z. Haas, P. Sholander, and S. S. Tabrizi, "On the impact of alternate path routing for load balancing in Mobile Ad-Hoc Networks," MobiHoc'2000, Boston, USA.

[6] P.-H. Hsiao, A. Hwang, H.T.Kung, D. Vlah, "Load-Balancing Routing for Wireless Access Networks," IEEE Infocom, 2001.

[7] R. Jain, W. Hawe, D. Chiu, "A quantitative measure of fairness and discrimination for resource allocation in Shared Computer Systems," DEC Research Report TR-301, Sept. 1984.

[8] R Karrer, A. Sabharwal, E.W. Knightly, "Enabling large-scale wireless broadband: the case for TAPs," *Computer Communication Review* 34(1), 2004, pp. 27-32.

[9] V. Gambiroza, B. Sadeghi, and E. Knightly, "End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks," in Proceedings of ACM MobiCom 2004, Philadelphia, PA, 2004.

TABLE 3. THE RELATIVE EXPERIMENTAL PARAMETERS

Parameters	Value	Parameters	Value
MAX_LR_ITERATION	10000	X_AXIS	5-14
LR_OSC_DEGREE	100	Y_AXIS	5-14
LR_CVG_DEGREE	200	NUMBER_OF_GATEWAYS	1, 2, 4
MAX_NUM_NODES	400	FI	0.9

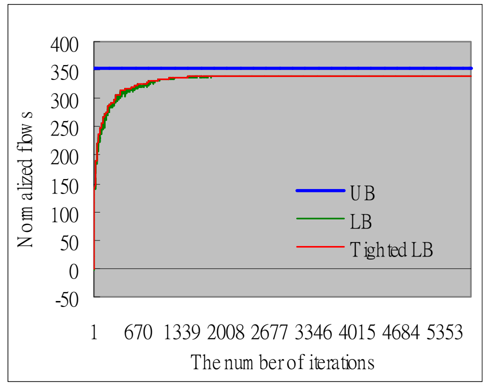

Figure 3. an example of LR experimental results with 4 gateways assignment 11*11 grid graphs. Here the gap is equal to 3.5% and FI = 0.92.

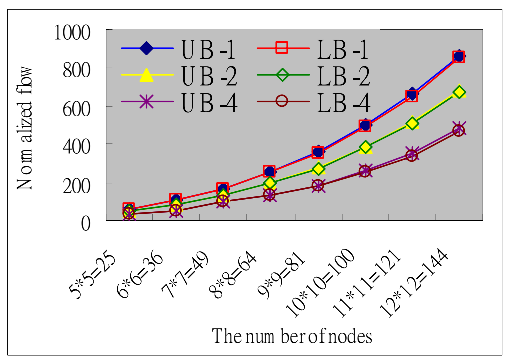

Figure 4. Case 1) grid-based experimental results where UB-*x* and LB-*x* denote the UB and LB with *x* gateways, respectively. The gap is less than 5%.

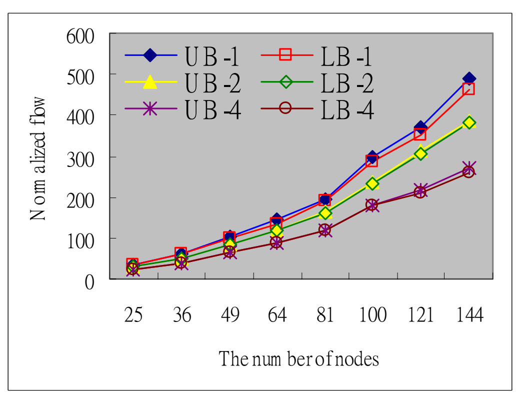

Figure 5. Case 2) random-based experimental results where UB-*x* and LB-*x* denote the UB and LB with *x* gateways, respectively. In this experiment, the gap is less than 10%.