I. J. Communications, Network and System Sciences, **2009, 1, 1-89**
Published Online February 2009 in SciRes (http://www.SciRP.org/journal/ijcns/).

Gateways Placement in Backbone Wireless Mesh Networks

Maolin TANG
Queensland University of Technology, Brisbane, QLD, Australia
Email: m.tang@qut.edu.au
Received October 6, 2008; revised November 5, 2008; accepted December 11, 2008

Abstract

This paper presents a novel algorithm for the gateway placement problem in Backbone Wireless Mesh Networks (BWMNs). Different from existing algorithms, the new algorithm incrementally identifies gateways and assigns mesh routers to identified gateways. The new algorithm can guarantee to find a feasible gateway placement satisfying Quality-of-Service (QoS) constraints, including delay constraint, relay load constraint and gateway capacity constraint. Experimental results show that its performance is as good as that of the best of existing algorithms for the gateway placement problem. But, the new algorithm can be used for BWMNs that do not form one connected component, and it is easy to implement and use.

Keywords: Gateway Placement, Backbone Wireless Mesh Networks

1. Introduction

A wireless mesh network is an ad hoc communication network that is made up of wireless communication nodes organized in a mesh topology. It allows for continuous connections and reconfiguration around broken or blocked paths by hopping from node to node until the destination is reached. In a wireless mesh network communication nodes can connect to each other via multiple hops and they are not mobile. The infrastructure that supports a mesh wireless network is a wireless router network, or backbone wireless mesh network (BWMN).

A BWMN consists of a collection of wireless mesh routers, each of which can communicate with other wireless mesh routers and clients. Each mesh router forwards packages on behalf of other mesh routers and clients. The wireless mesh routers are generally not mobile. The clients connect to the wireless network through a wireless mesh router and they do not have restriction on mobility.

Gateway placement is an important problem in the design of BWMNs. It determines network points, or gateways, through which a BWMN communicates with other networks. The objective is to minimize the total number of gateways subject to Quality-of-Service (QoS) constraints. There are three popular QoS constraints in the design of BWMNs: *delay constraint, relay load constraint and gateway capacity constraint.*

A BWMN is a multi-hop network where significant delay occurs at each hop due to contention for the wireless channel, pocket processing, and pocket queuing. The delay is therefore a function of the number of communication hops between the mesh router and its gateway [1]. The delay constraint requires that the maximal number of hops from any mesh router to a gateway is not greater than a given number R. In the placement of BWMN gateways, it is important to guarantee the throughput for individual traffic flows. Since it is assumed in this research that a BWMN has multiple communication channels, which allow interfering wireless links operate on different communication channels concurrently, the bottleneck on throughput is therefore reduced to the load on the link individual links between wireless routers as relay load L. In addition, the throughput of a BWMN depends on the bandwidth and processing speed of the gateways. Thus, a gateway has a capacity S, which is measured by the maximal number of mesh routers that it can serve.

This paper presents a new algorithm, namely incremental clustering algorithm, for the gateway placement problem. Compared with existing algorithms for the gateway placement problem, the new algorithm has the following advantages: first, it guarantees to find a gateway placement satisfying all the constraints; second, it has competitive performance; third, it can be used for the BWMNs that does not form a connected component; fourth, it is easy to implement and use.

The remaining paper is organized as follows. The following section formulates the BWMN gateway placement problem, which is followed by a discussion of related work. Then, we discuss our incremental clustering

algorithm and show our experimental results on our incremental clustering algorithm. Finally, we conclude the incremental clustering algorithm.

2. Problem Formulation

A BWMN can be represented by a directed graph $G = (V, E)$. Each node $v =< x, y, r > \in V$ represents a mesh router, where x and y are the x-coordinate and y-coordinate of the location of v and r is the radius of the circular transmission range of v. Arc $< v_i, v_j > \in$ E if and only if mesh router v_j is in the transmission range of mesh router v_i, or $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \le r_i$, where $v_i =< x_i, y_i, r_i >$ and $v_j =< x_j, y_j, r_j >$. Note that $< v_i, v_j > \in E$ does not implies $< v_j, v_i > \in E$ because the radiuses of their transmission range may be different.

A mesh cluster is a set of vertices $C \subseteq V$. A mesh cluster has a cluster head $h \in C$. The nodes in C and the arcs between them define a *cluster graph* $G_C = (C, E_C)$, where an arc $< v_i, v_j > \in E_C$ if and only if $v_i \in C$, $v_j \in$ C, and $< v_i, v_j > \in E$. A mesh cluster is connected if and if only the corresponding cluster graph is connected. The radius of a mesh cluster r_C is defined as the maximal shortest distance between from the mesh cluster head h and the nodes in C. The delay constraint is translated into an upper bound R on the mesh cluster radius.

The *shortest path spanning tree* is a spanning tree of G_C, $T(G_C)$, which is formed by composing the shortest paths from the cluster head h to all the other nodes in C. The nodes at i^{th} level of the shortest path spanning tree have i hops to the cluster head h. The depth of $T(G_C)$ is denoted $d(T(G_C))$. Let v be a node in $T(G_C)$. The number of nodes in the subtree rooted v is denoted $\pi(v)$.

Given a BWMN represented by a directed graph $G = (V, E)$, a delay constraint R, a relay load constraint L and a gateway capacity constraint S, the BWMN gateway placement problem is to find a set of connected clusters $\{C_1, C_2, \cdots C_n\}$ and their corresponding clusters' shortest path spanning threes such that n is minimal subject to

(a) $C_1 \bigcup C_2 \bigcup \cdots \bigcup C_n = V$;
(b) $|C_k| \le S$, where $1 \le k \le n$;
(c) $d(C_k) \le R$, where $1 \le k \le n$;
(d) $\forall v \in T(G_{C_k})$, $\pi(v) \le L$.

The shortest path spanning threes give a gateway placement solution where the roots represent the mesh router where a gateway is placed and the links specify the communication topology.

Condition (a) guarantees that a BWMN gateway placement solution covers all mesh routers; Condition (b) ensures that the gateway capacity constraint S is satisfied; Condition (c) enforces that the delay constraint R is met; Condition (d) makes sure that the relay load constraint L is respected.

3. Related Work

From the computational point of view, the gateway placement problem is conjectured as an NP-hard problem as it can be transformed into the minimum dominating set problem [1], which has been proven to be NP-complete [2]. Thus, optimal algorithms are not suitable for solving the problem as they would lead to combinatorial explosion in the search space when the problem size is large. Because of the reason, all existing algorithms for the gateway placement problem are heuristic or approximation ones.

In [3], Wong, *et al*. addressed two gateway placement problems: one is to optimize the communication delay and another is to optimize the communication cost. Although the algorithms can be extended to consider delay, relay load, and gateway capacity constraints, they can only be used for BWMNs that form a connected component and require at least two hops for communication between at least one pair of nodes.

The algorithm proposed by Bejerano in [4] uses a clustering technique and performs the gateway placement problem in four stages: select cluster heads, assign each node to an identified cluster satisfying the delay constraint, break down the clusters that do not satisfy the relay load constraint or the gateway capacity constraint, and finally select gateways to reduce the maximum relay load. However, the algorithm does not have competitive performance because of the following two reasons: first, when identifying cluster heads and assigning mesh routers to the identified cluster heads, the algorithm does not make use of global information about the BWMN; second, splitting a cluster without considering re-assigning those mesh routers to existing clusters may create some unnecessary clusters and therefore increases the number of clusters significantly.

In [5], Chandra, *et al*. explored the placement problem of Internet Transit Access Points (ITAPs) in wireless neighborhood networks under three wireless link models, and for each of the wireless link models, they developed algorithms for the placement problem based on neighborhood layouts, user demands, and wireless link characteristics. The placement problem is similar to the gateway place of BWMN. However, their algorithms consider only one constraint, that is, users' bandwidth requirements.

The work closest to ours is the algorithm proposed by Aoun, *et al*. in [1], which transforms the gateway placement problem into the minimum dominating set problem and adopts a recursive dominating set algorithm to tackle the minimum dominating set problem. The algorithm considers the delay, relay load and gateway constraints and has better performance than the Wong's algorithms, the Bejerano's algorithms, and the Chandra's algorithms. However, it has the following deficiencies: first, it can be used for those BWMNs that form a connected component; second, it needs to set the initial

radius size properly; otherwise, it would not create satisfactory results.

4. The Incremental Clustering Algorithm

4.1. Preliminaries

In this paper, the *transitive closure* of a directed graph $G = (V, E)$ is a directed graph $G^+ = (V, E^+)$ such that for $\forall < u,v > \in E^+$ if and only if there exists a non-null path from u to v. The *n-step transitive closure* of a directed graph $G = (V, E)$ is a directed graph $G^n = (V, E^n)$ such that for $\forall < u,v > \in E^n$ if and only if there exists a non-null path from u to v and the length of the path is less than or equals to n. Figure 1 shows a BWMN graph. The transitive closure and the 2-step transitive closure are displayed in Figure 2 and Figure 3 respectively.

A BWMN graph $G = (V, E)$ can be represented by an $n \times n$ adjacency matrix $A = [a_{ij}]_{n \times n}$, where

$$a_{ij} = \begin{cases} 1, \ if \ v_i, v_j \in V \text{ and } < v_i, v_j > \in E; \\ 0, \ \text{otherwise.} \end{cases} \quad \textbf{(1)}$$

For example, for the BWMN graph shown in Figure 1, its adjacency matrix is shown in Equation 2. The adjacent matrix representations for its transitive closure and its 2-step transitive closure are displayed in Equation 3 and Equation 4 respectively.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \textbf{(2)}$$

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \textbf{(3)}$$

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \textbf{(4)}$$

Figure 1. A BWMN graph G.

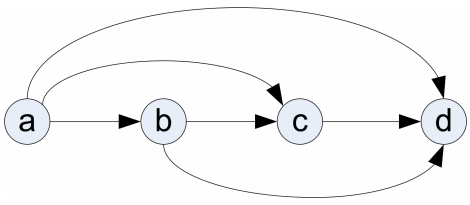

Figure 2. The transitive closure of G.

4.2. Algorithm Description

The incremental clustering algorithm solves the gateway placement problem by iteratively and incrementally identifying gateways and assigning mesh routers to identified gateways.

Algorithm 1 is the algorithm description.

Algorithm 1 Incremental clustering algorithm

while $U \neq \phi$ **do**
  construct a BWMN graph from U;
  build the R-step transitive closure from the BWMN graph;
  identify gateways from the R-step transitive closure;
  assign mesh routers in U to identified gateways subject to the R, L and S constraints;
  remove the assigned mesh routers from U.
end while

In Algorithm 1, U is the set of mesh routers; R, L and S represent the delay constraint, relay load constraint and the gateway capacity constraint, respectively.

The incremental clustering algorithm is an iterative one. In each iteration, it starts with constructing a BWMN graph from the current unassigned mesh router set U, and then builds the R-step transitive closure from the BWMN graph, and then identifies gateways based on the R-step transitive closure, and finally assigns mesh routers to the identified gateways and removes the assigned mesh routers from U. The process is repeated until U is empty. By the time U is empty, every mesh router has been assigned to a gateway. This algorithm is incremental as it incrementally identifies gateways and assigns mesh routers to identified gateways, rather than identifying all gateways and assigning all mesh routers to the gateways in one step. Since the construction of a BWMN graph has been already introduced in the previous subsection and the algorithm for building an R-step transitive closure is well-known, we focus on discussing how to identify gateways from the R-step transitive closure and how to assign mesh routers to identified gateways in the following.

It can be observed that the i^{th} row of the R-step transitive closure is a cluster, representing a set of mesh routers that can be covered by the i^{th} mesh router, where $1 \leq i \leq |U|$. The i^{th} mesh router is the head of the mesh cluster. The mesh router clusters can be classified into *covered clusters* and *uncovered clusters*. A mesh cluster is a uncovered one if there exists one mesh router in the mesh cluster that is not present in the other mesh clusters; Otherwise, the mesh cluster is a covered one. It can be observed that there is one and only one mesh router that cannot be covered by the other mesh cluster in a uncovered cluster, which is the head.

For each uncovered mesh cluster, at least one gateway is needed as the head of the mesh cluster cannot use any mesh router in other mesh clusters as its gateway because it cannot be covered by any other mesh routers

in the other clusters. Thus, we select the head of a uncovered cluster as a gateway. However, sometimes there is no uncovered mesh cluster (we will give an example when illustrating the algorithm later). If this is the case, we select the head of the mesh cluster that covers the maximal number of mesh routers as a gateway. Algorithm 2 is the algorithm for identifying gateway.

Algorithm 2 Identifying gateways

```
for i =1 to |U| do
    if the corresponding mesh router cluster of the i^th
    row of the R-step transitive closure is a uncovered
    mesh cluster
    then
        the head of the mesh router cluster is selected
        as a gateway;
    end if
end for
if no uncovered mesh cluster was found then
    find a mesh router cluster has the maximal size;
    the head of the mesh router cluster is selected as a
    gateway.
end if
```

Once gateways have been identified using the technique described above, we assign as many mash routers as possible to those identified gateways subject to the delay, relay load, and gateway capacity constraints to minimize the total number of gateways. Algorithm 3 is the algorithm for assigning mesh routers to identified gateways.

Algorithm 3 Assigning mesh routers to identified gateways

```
for each gateway g do
  for h =0 to R do
    for any mesh router that is covered by g and the
    shortest distance to g is h do
      if not violating any of the constraints then
        assign the mesh router to g;
        remove the mesh router from the other gateways,
        if any;
      end if
    end for
  end for
end for
```

4.3. Algorithm Analysis

The incremental clustering algorithm is iterative. In each iteration, the algorithm identifies at least one gateway, assigns at least one mesh router to an identified gateway and therefore the number of unassigned mesh routers decreases by one. Thus, the algorithm terminates after at most $n - 1$ iterations, where n is the total number of mesh routers. In addition, an assignment is accepted only when it does not violet the constraints. So, it is guaranteed that the algorithm generates a feasible solution when it terminates.

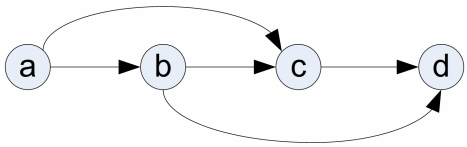

Figure 3. The 2-step transitive closure of G.

Assume that $G = (V, E)$ is the BWMN graph of a BWMN gateway placement problem. The computational complexity of the incremental clustering algorithm in the worst case is $O(R \times |V|^3 + |E| \times |V|^2)$. The following is the proof.

In the worst case, the algorithm iterates $|V|$ times. In each of the iterations, the algorithm identifies only one gateway and assigns only one mesh router (the mesh router at the gateway) to the gateway. Thus, the algorithm builds the BWMN graph $|V|$ times. It takes $O(|V|^2)$ time to build a BWMN graph that has $|V|$ nodes (it is assumed that the adjacent matrix representation is used.). Thus, the total computational complexity for building BWMN graphs is $O(|V|^3)$. It takes $O(R \times |V|^2)$ to construct an R-step transitive closure. In the worst case, the R-step transitive closure needs to be constructed $|V|$ times. Thus, the total computational complexity for constructing R-step transitive closures is $O(R \times |V|^3)$. In addition, given an R-step transitive closure, it takes $O(|E| \times |V|)$ time to identify a gateway in an iteration. Thus, the total computational complexity for identifying gateways is $O(|E| \times |V|^2)$. It takes at most $O(|V|)$ to assign a mesh router to a gateway (it needs to remove the assigned mesh router from the other mesh router clusters). Thus, the total computational complexity for assigning $|V|$ mesh routers is $O(|V|^2)$ in the worst case. Thus, the computational complexity in the worst case is $O(|V|^2 + R \times |V|^3 + |E| \times |V|^2 + |V|^2) = O(R \times |V|^3 + |E| \times |V|^2)$.

4.4. Algorithm Illustration

This section uses an example to illustrate how the incremental clustering algorithm works. The BWMN gateway placement problem is given in a BWMN graph shown in Figure 4. In the BWMN there are nine mesh routers that may have different coverage radiuses. For example, the coverage radius of mesh router v_8 is larger than that of mesh router v_9. As a result, mesh router v_8 can cover mesh router v_9, but not the other way around. Figure 5 is the matrix representation of the BWMN graph shown in Figure 4.

For this BWMN gateway placement problem, we assume that the delay constraint $R = 2$, the relay load constraint $L = 2$, the gateway capacity constraint $S = 3$. In other words, for this BWMN gateway placement problem we need to find a solution such that the maximum hop from any mesh router to its gateway must not exceed 2, every mesh router must not relay packets for more than 2 mesh routers, and each gateway must not serve for more than 3 mesh routers.

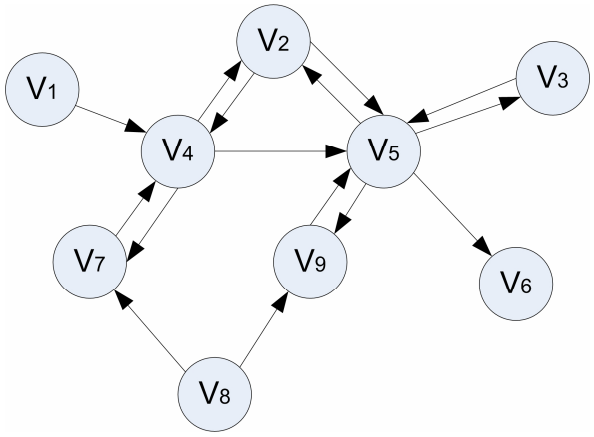

Figure 4. The BWMN graph.

The algorithm starts with finding the 2 transitive closure of the BWMN graph. Figure 6 displays the matrix representation of the 2-step transitive closure of the BWMN graph.

Then, the algorithm identifies gateways using the procedure described in Algorithm 2. Since the mesh router clusters corresponding to the 1th and the 8th rows of the 2-step transitive closure are the only uncovered mesh router clusters, v_1 and v_8 are identified as gateways. The algorithm then uses the procedure described in Algorithm 3 to assigns mesh routers in U to v_1 and v_8 as many as possible subject to the *R*, *L* and *S* constraints. The assigning procedure starts with v_1.

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 5. The matrix representation of the BWMN graph.

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Figure 6. The matrix representation of the 2-step transitive closure of the BWMN graph.

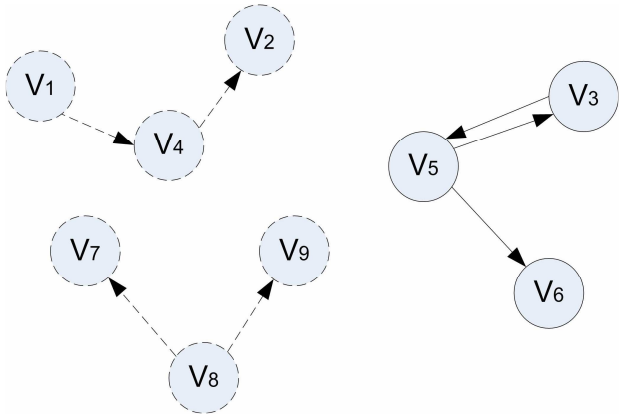

Figure 7. The intermediate state of the BWMN graph.

It considers all the mesh routers that can be covered by v_1 according to the information given in the 2-step transitive closure in Figure 6 in the descending order of the number of hops from the mesh router to v_1. As a result, v_1, v_4 and v_2 are assigned to gateway v_1 in the order. The assigning procedure then uses the same idea to assign mesh routers to v_8, v_7 and v_9 to gateway v_8. Figure 7 shows the state after this iteration of identifying gateways and assigning mesh routers. In the figure, the components drawn in broken lines represent the assigned mesh routers and the components drawn in solid lines represent the mesh routers that have not been assigned to any gateway.

Since there are still some mesh routers that have not been assigned to any gateway, the algorithm repeats the above process. It creates a BWMN graph for the remaining mesh routers and then generates a 2-step transitive closure of the BWMN graph. Figures 8 and 9 show the matrix representation of the BWMN and the 2-step transitive closure of the BWMN graph, respectively.

From the 2-step transitive closure of the BWMN graph, the algorithm identifies gateways using the procedure described in Algorithm 2. Since all the mesh router clusters are covered ones, the mesh router cluster that has the largest size, which is v_5, is selected as a gateway. The algorithm then assigns the rest mesh routers to gateway v_5. Figure 10 shows the final placement result. As displayed in the figure, three gateways are needed to be placed.

5. Experimental Results and Discussions

This section evaluates the performance of the incremental clustering algorithm by comparing it with three top algorithms for the gateway placement problem by simulation. The three top algorithms are the weighted recursive algorithm proposed by Aoun, *et al.* in [1], the iterative greedy algorithm proposed by Bejerano in [4], and an augmenting algorithm similar to those proposed by Wong, *et al.* in [3] and by Chabdra, *et al.* in [5]. The performance of the four algorithms are evaluated and

compared in terms of the delay constraint, the relay load constraint, and the gateway capacity constraint respectively.

We have developed a MATLAB program to randomly generate gateway placement problems. All the generated gateway placement problems have 200 mesh routers on a 10 × 10 plane. The connection radius is 1.0, and the minimum distance between any pair of mesh routers is 0.5. We have use the program to generate 30 instances for each of the set-ups, and have used the four algorithms to solve the gateway placement problems. The performance of the algorithms is evaluated by the average number of gateways of the 30 runs for each of the set-ups in each of the evaluations.

The implementations of the weighted recursive algorithm, the iterative greedy algorithm, and the augmenting algorithm used in the evaluations is the ones used in [1] and is kindly provided by Mr Bassam Aoun and Prof. Raouf Boutaba. However, the program used for randomly generating test problems is different from the one used in [1]. Given a parameter *n*, the test problem generator used in [1] randomly creates a test problem that contains up to *n* mesh routers, but the test problem generator used in our experiments randomly creates a test problem that has exactly *n* mesh routers, which makes the experimental results more accurate and reliable.

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 8. The matrix representation of the BWMN graph.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 9. The matrix representation of the 2 transitive closure of the BWMN graph.

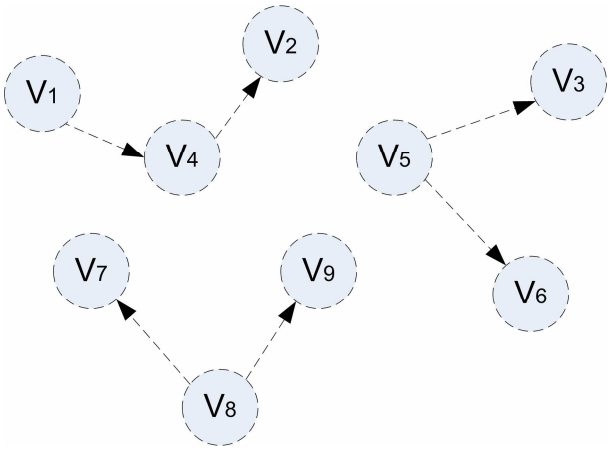

Figure 10. The solution.

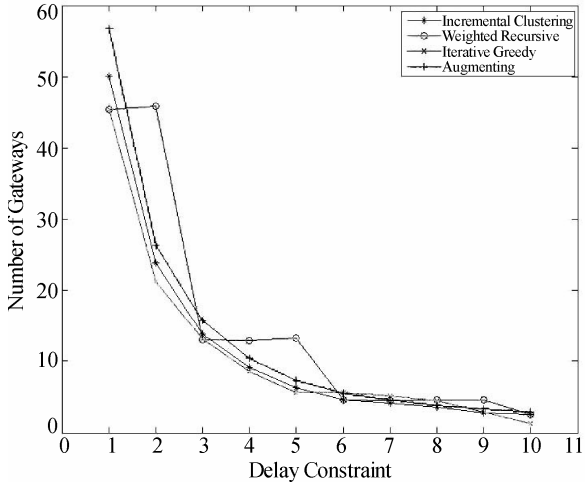

Figure 11. Comparison of the effects of the hop constraint on the four algorithms. L = NaN. S = NaN.

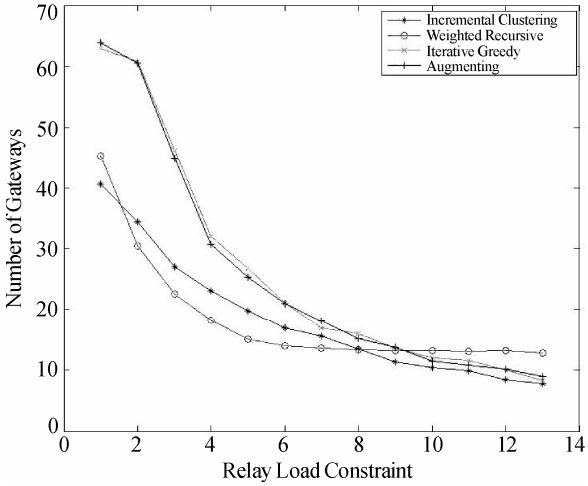

Figure 12. Comparison of the effects of the link capacity Constraint on the four algorithms. R = 8. S = NaN.

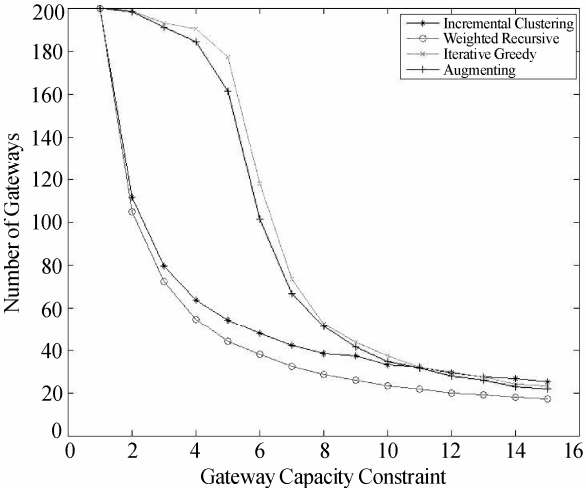

Figure 13. Comparison of the effects of the gateway constraint on the performance of the four algorithms. R = 8. L = NaN.

5.1. Effects of the Delay Constraint

The effects of the delay constraint on the performance of the four algorithms are evaluated in this section. In the evaluation, the relay load constraint and the gateway capacity constraint are relaxed. The values of the delay constraint vary from 1 to 10. Figure 11 shows the evaluation results.

It can be seen from the figure that the performance of the incremental clustering algorithm is similar to that of the iterative greedy algorithm and the augmenting algorithm, but it is better than that of the weighted recursive algorithm, under the delay constraints.

5.2. Effects of the Relay Load Constraint

This section evaluates the effects of the relay load constraint on the performance of the four algorithms. In this evaluation, the link capacity constraint varies from 1 to 13, the gateway capacity constraint is relaxed, and the delay constraint is fixed to 8. Figure 12 illustrates the evaluation results.

The evaluation results show that the performance of the incremental clustering algorithm is better than that of the iterative greedy algorithm and the augmenting algorithm. It also outperforms the weighted recursive algorithm when the relay load constraint is 1 and when the replay load constraint is greater than 8. But, it is not as good as that of the weighted recursive algorithm when the link capacity is between 2 and 8. In overall, the performance of the incremental clustering algorithm is as good as that of the weighted recursive algorithm, which is the best among the existing gateway placement algorithms, under the relay load constraints.

5.3. Effects of the Gateway Capacity Constraint

The effects of the gateway capacity constraint on the performance of the four algorithms are studied in this section. In this evaluation, we test the performance of the four algorithms when the gateway capacity constraint varies from 1 to 15. The delay constrain is set to 8 and the relay load constraint is relaxed. Figure 13 shows the performance of the four algorithms in relation to the gateway capacity constraint.

The figure shows that that the performance of the weighted recursive algorithm is the best among the four algorithms. The performance of the incremental clustering algorithm is similar to that of the weighted recursive algorithm, and it is better than that of the iterative algorithm and the augmenting algorithm when the gateway capacity constraint is tight. When the gateway capacity constraint is relaxed, the performances of the recursive clustering and assignment algorithm, the iterative greedy algorithm, and the augmenting algorithm are close to each other.

6. Conclusions

This paper has presented a new algorithm for the gateway placement problem. Different from existing algorithms for the gateway placement problem, this new algorithm incrementally identifies gateways and assigns remaining mesh routers to the identified gateways. By incrementally identifying gateways, the new algorithm can exploit the dynamically generated information about the distribution of unassigned mesh routers; By incrementally assigning mesh routers to a gateway, the new algorithm can fully explore mesh router assignment options and therefore benefit to reduce in the number of gateways. Experimental results have shown that in overall the performance of the new algorithm is as good as that of the best of the three top algorithms, and sometimes it outperforms the best algorithm.

In addition to its good performance, the new algorithm has the following advantages: first, it guarantees to find a gateway placement satisfying all the constraints; second, it has competitive performance; third, it can be used for the BWMNs that does not form a connected component; fourth, it is easy to implement and use.

7. Acknowledgement

The author would like to thank Mr Bassam Aoun and Prof. Raouf Boutaba for providing the source code used in their research on the gateway placement problem in [1].

8. References

[1] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, “Gateway placement optimization in wireless mesh networks with QoS constraints,” IEEE Journal on Selected Areas in Communications, Vol. 24, No. 11, pp. 2127– 2136, November 2006.

[2] M. R. Garey and D. S. Johnson, “Computers and intractability; A guide to the theory of NP-completeness,” New York, NY, USA: W. H. Freeman & Co., 1990.

[3] J. Wong, R. Jafari, and M. Potkonjak, “Gateway placement for latency and energy efficient data aggregation,” in Proceedings IEEE LCN, pp. 490–497, 2004.

[4] Y. Bejerano, “Efficient integration of multihop wireless and wired networks with QoS constraints,” IEEE/ACM Transactions on Networking, Vol. 12, No. 6, pp. 1064–1078, December 2004.

[5] R. Chandra, L. Qiu, K. Jain, and M. Mahdian, “Optimizing the placement of Internet TAPs in wireless neighborhood networks,” in Proceedings IEEE ICNP, pp. 271–282, 2004.

