

Copyright
by
Hrishikesh B. Acharya
2011

The K -Observer Problem in Computer Networks

by

Hrishikesh B. Acharya , B.Tech.(Hons.)

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2011

The Thesis Committee for Hrishikesh B. Acharya Certifies
that this is the approved version of the following thesis:

The K -Observer Problem in Computer Networks

APPROVED BY

SUPERVISING COMMITTEE:

Mohamed G. Gouda, Supervisor

Alyosius K. Mok

Acknowledgments

I would like to thank my advisor Dr Mohamed Gouda, my colleague Mr Taehwan Choi, and Dr Rida A Bazzi of the Arizona State University, who were kind enough to guide and collaborate with me in the execution of this research.

The K -Observer Problem in Computer Networks

Hrishikesh B. Acharya , M.S.

The University of Texas at Austin, 2011

Supervisor: Mohamed G. Gouda

For any non-negative integer K , a K -observer P of a network N is a set of nodes in N such that each message, that travels at least K hops in N , is handled (and so observed) by at least one node in P . A K -observer P of a network N is *minimum* iff the number of nodes in P is less than or equal the number of nodes in every K -observer of N . The nodes in a minimum K -observer of a network N can be used to monitor the message traffic in network N and collect traffic statistics, detect patterns of denial-of-service attacks when they occur in N , and act as firewalls to identify and discard attack messages from N . This thesis considers the problem of constructing a minimum K -observer for any given network. We show that the problem is NP-hard for general networks, and give linear time algorithms for constructing minimum or near-minimum K -observers for special classes of networks: trees, rings, L -rings, and large grids.

Table of Contents

Acknowledgments	iv
Abstract	v
List of Figures	vii
Chapter 1. Introduction	1
Chapter 2. The K -Observer Problem	5
Chapter 3. K -Observers of Tree Networks	11
Chapter 4. K -Observers of Ring Networks	16
Chapter 5. K -Observers of Large Grid Networks	19
Chapter 6. Applications	22
Chapter 7. Concluding Remarks	24
Bibliography	26
Vita	30

List of Figures

2.1	Example of constructing network N' from network N	6
3.1	Example of a tree network T	13
3.2	Computing minimum K -observers of T	15
4.1	A ring network of n nodes	16
5.1	A section of $(r + 1) \times (r + 1)$ nodes in a large grid network . .	19

Chapter 1

Introduction

Every node in a computer network performs a number of traditional tasks: generating messages, routing and forwarding messages, and consuming messages. Beside these traditional tasks, some nodes are designated, in a network, to perform additional tasks: observing and collecting statistics concerning the message traffic that goes through each designated node and filtering the message traffic that goes through each designated node. We refer to those nodes that are designated, in a computer network, to perform these additional tasks as *network observers*.

This thesis discusses the problem of how to select the nodes to be designated network observers in a computer network. We start the discussion by proposing a criterion, named minimum K -observers for some non-negative integer K , which can be used to identify the network observers in a computer network.

A K -observer P of a network N is a set of nodes in N such that each message, that travels at least K hops in N , is handled (and so observed) by at least one node in P .

Clearly, this definition of a K -observer P depends on the chosen value

of K . On one hand, if the chosen value of K is small (with respect to the total number of nodes n in N), then P is a large set containing most of the nodes in N . For example, if $K = 0$, then P is the set of all nodes in N . On the other hand, if the chosen value of K is large (with respect to the total number of nodes n in N), then P can be a small set containing few nodes in N . For example, if $K = n - 1$, then P can be a singleton containing only one node (any node) in N . Also, if $K \geq n$, then P can be the empty set.

Note that if a set P is a K -observer of a network N , then adding more nodes of N to P does not change the status of P of being a K -observer of N . This note suggests that we should be more interested in minimum K -observers, rather than in K -observers, as defined next.

A K -observer P of a network N is *minimum* iff the number of nodes in P is less than or equal the number of nodes in every K -observer of N .

Based on this discussion, in order to identify the network observers in a computer network N , one needs to construct a minimum K -observer P of network N , for some chosen K , and use the nodes in the constructed P as the network observers in N .

The problem of constructing a minimum K -observer in a network is related, though not identical, to several established problems in network design:

1. Constructing a node cover in a network

The *Node Cover Problem* is to find a minimum number of nodes such

that each link is incident to at least one node in the network [11]. The K -observer problem is more generalized than the *Node Cover Problem* such that the Node Cover Problem is the special case of the K -observer problem when $K = 1$. [4] discusses the Node Cover Problem for sources and destinations of all paths in the network whereas the K -observer problem generalizes the Node Cover Problem for a path of length K in the network.

2. Creating a backbone for communication

A very important problem for wireless networks, the *Connected Dominating Set Problem*, is to find a set of nodes that are connected, such that each node in the network either belongs to this set of nodes or is one hop from it [23]. A connected dominating set is used for virtual backbones in wireless networks. Although mobile networks do not have physical backbones, virtual backbones can be formed to help communication in wireless ad-hoc networks [3, 21, 16].

3. Placing guards in an art gallery

The *Art Gallery Problem* is to determine the number of guards necessary to cover an art gallery, such that every point in the art gallery is guarded by at least one observer [8]. This problem is equivalent to the *Coverage Problem* in the context of wireless sensor networks [13], wireless ad-hoc networks, and wireless sensor ad-hoc networks [17]. Moreover, this problem is equivalent to the *Dominating Set Problem* if guards must be placed on nodes, and only nodes need to be guarded [11].

4. The problem of facility location in a network

The *Facility Location Problem* is to find a place for a facility such that the distances from customers are minimized [22]. This problem can be regarded as the *Set Cover Problem* [14]; it has been discussed in various contexts such as the placement of monitoring nodes [5], web server replicas [18], and overlay nodes [20].

The rest of the thesis is organized as follows. We begin by proving that our K -observer problem is NP hard for general networks in Chapter 2. We then explore solutions for some special cases: tree networks in Chapter 3, ring networks in Chapter 4, and grid networks in Chapter 5. We go on to discuss some possible applications for the K -observer problem in Chapter 6, and conclude with a few remarks.

Chapter 2

The K -Observer Problem

A *network* N is an undirected graph (V, E) , where V is a nonempty set of *nodes* and E is a set of undirected *links*. Each *link* in E is a set of two distinct nodes in V . A link $\{u, v\}$ in a network N is said to be *incident* at nodes u and v in N .

A *path* in a network N is a nonempty sequence (u_1, u_2, \dots, u_r) of distinct nodes in N such that each pair $\{u_i, u_{i+1}\}$ of consecutive nodes in the sequence constitutes a link in network N .

The *length* of a path (u_1, u_2, \dots, u_r) in a network is $r - 1$. For example, the length of the path (u_1) is 0, the length of the path (u_1, u_2) is 1, and so on.

Let P denote a set of nodes in a network N and let K be a non-negative integer. Set P is called *K-observer* of N iff every path of length at least K in N has at least one node in P .

A K -observer P of a network N is called *minimal* iff for each node u in P , N has a path q of length at least K such that P and q share only one node and their shared node is u .

A minimal K -observer P of a network N is called *minimum* iff for every minimal K -observer Q of network N , the number of nodes in Q is at least the

number of nodes in P .

Let K be a non-negative integer. The K -observer problem is to design an algorithm that takes as input any network N and produces as output a minimum K -observer of network N .

An algorithm that solves the K -observer problem, when $K = 0$, can be designed as follows. This algorithm takes any network N as input and produces the set of all nodes in N as output. The correctness of this algorithm is based on the observation that the only 0-observer (and so the only minimum 0-observer) of a network N is the set of all nodes in N .

Unfortunately, the following theorem states that the K -observer problem, for any $K > 0$, is NP-hard, so any algorithm to solve this problem is very likely to be expensive.

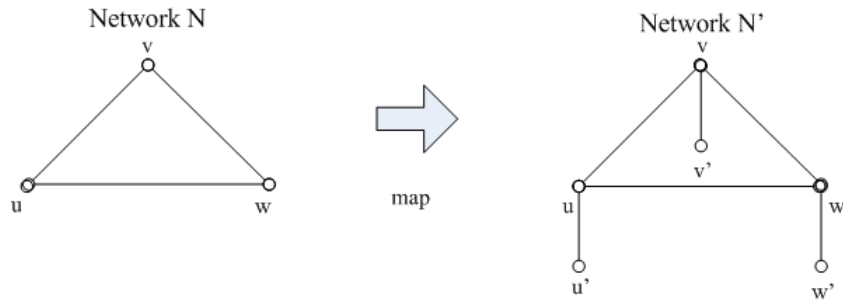


Figure 2.1: Example of constructing network N' from network N

Theorem 2.0.1. *The K -observer problem, for any value of K that is greater than 0, is NP hard.*

Proof. It is sufficient to prove that the following K -observer decision problem,

where $K > 0$, is NP hard. The K -observer decision problem is to design an algorithm that takes, as input, any network N and any positive integer m and produces, as output, a determination of whether N has a minimal K -observer P that has exactly m nodes. We refer to the pair (N, m) as an *instance* of the K -observer decision problem.

We prove that the K -observer decision problem, for any value of K greater than 0, is NP hard by induction on K .

Base Step $K = 1$:

The 1-observer decision problem is exactly the same as the node cover decision problem, which is known to be NP hard[9, 11].

Induction Step:

Assuming that the K -observer decision problem is NP hard, we show that the $(K + 1)$ -observer decision problem is NP hard. In particular, we show that any algorithm that solves the $(K + 1)$ -observer decision problem can be also used to solve the K -observer decision problem.

Let (N, m) denote an instance of the K -observer decision problem that we need to solve. We use (N, m) to construct a corresponding pair (N', m') which is an instance of the $(K + 1)$ -observer problem such that the following two conditions hold:

- (i) If N' has a minimal $(K + 1)$ -observer P' that has exactly m' nodes, then N has a minimal K -observer P that has exactly m nodes.

- (ii) If N has a minimal K -observer P that has exactly m nodes, then N' has a minimal $(K + 1)$ -observer P' that has exactly m' nodes.

The pair (N', m') is constructed from the pair (N, m) as follows:

1. N' is constructed from N as follows. First, for each node u in N , add two nodes u and u' in N' : node u is called an *original node* and node u' is called the *shadow node* corresponding to node u . Second, add an link $\{u, u'\}$ between each original node u and its corresponding shadow node u' in N' . Third, for each link $\{u, v\}$ between two nodes u and v in N , add a link $\{u, v\}$ between the two corresponding original nodes u and v in N' . (See the example in Figure 2.1.)
2. $m' = m$

Now we prove that the constructed pair (N', m') satisfies the two conditions (i) and (ii) mentioned above.

Proof of (i):

Assume that network N' has a minimal $(K + 1)$ -observer P' that consists of m' nodes. Note that because P' is a minimal $(K + 1)$ -observer, then P' cannot have both an original node u and its shadow node u' .

Construct from this P' a subset P of nodes in N as follows.

1. For every original node u in P' , add the corresponding node u in P .
2. For every shadow node u' in P' , add the corresponding node u in P .

Note that because P' cannot have both an original node and its shadow node, the number of nodes in P' equals the number of nodes in the constructed P . It remains to prove that (a) the constructed P is a K -observer of network N , and (b) the constructed P is minimal.

Proof of (a):

Consider any path q of length at least K in network N and assume that this path ends at node u in N . Now consider path q' in N' that corresponds to path q in N . Path q' ends at the original node u in N' . Extend path q' by one link to the shadow node u' . The extended path is of length at least $(K+1)$, and at least one node in the extended path is in the $(K+1)$ -observer P' . Hence, at least one node in q is in set P . This shows that P is a K -observer of network N .

Proof of (b)

Consider any node u in the K -observer P of network N . This node u corresponds to one node v which is either the original node u or the shadow node u' in the $(K+1)$ -observer P' . Be-

cause P' is minimal, network N' has a path q' of length $K + 1$ such that P' and q' share only node v . Thus, network N has a path q of length K such that P and q share only node u . This shows that P is a minimal K -observer of network N .

Proof of (ii):

Assume that network N has a minimal K -observer P that consists of m nodes. Construct from this P a subset P' of nodes in N' as follows. For every node u in P , add the corresponding node u in P' . Note that the number of nodes in P equals the number of nodes in the constructed P' .

It remains to prove that (a) the constructed P' is a $(K + 1)$ -observer of network N' , and (b) the constructed P' is minimal. The proofs of (a) and (b) in this case are similar to the proofs of (a) and (b), presented earlier.

□

Having shown that the K -observer problem is NP hard for general networks, we present next polynomial time (in fact linear time) algorithms for solving this problem for special classes of networks such as tree networks, ring networks, and grid networks.

Chapter 3

K -Observers of Tree Networks

In this chapter, we solve the K -observer problem for tree networks that have no cycles. Figure 3.1 shows an example of a tree network T that has 13 nodes, named node 0 to node 12, and 12 links.

Next, we describe an algorithm that computes a minimum K -observer for a tree network. This algorithm takes as input a tree network T and a positive integer K and returns as output a minimum K -observer P of network T . This algorithm consists of the following four steps:

Step 1:

Choose any node in T to be the root and add directions to the links in T to make T a directed tree where the root is a sink node.

Step 2:

Define for each node x in T , a variable named len_x whose range of values is $0 \dots K$.

Step 3:

For each node x in T , where the values of the len variables of all predecessor nodes of x have already been computed, compute the value of

len_x as follows:

$$\begin{array}{ll}
len_x &:= 0 & \text{if } x \text{ has no predecessor } y \text{ whose } len_y < K \\
&:= len_y + 1 & \text{if } x \text{ has exactly one predecessor } y \text{ whose } len_y < K \\
&:= len_y + 1 & \text{if } x \text{ has two or more predecessors } \{y, z, \dots\} \\
& & \text{whose } len's < K \text{ and } len_y \text{ is the maximum } len \\
& & \text{among those predecessors and } len_z \text{ is the second maximum } len \\
& & \text{among those predecessors and } (len_y + len_z + 2) < K \\
&:= K & \text{if } x \text{ has two or more predecessors } \{y, z, \dots\} \\
& & \text{whose } len's < K \text{ and } len_y \text{ is the maximum } len \\
& & \text{among those predecessors and } len_z \text{ is the second maximum among those predecessors} \\
& & \text{and } (len_y + len_z + 2) \geq K
\end{array}$$

Step 4:

A minimum K -observer of tree network T is the set of every node x in T where $len_x = K$.

Theorem 3.0.2. *If this algorithm is applied to a tree network T , and a positive integer K , then the computed set by this algorithm is a minimum K -observer of network T .*

To see how this algorithm works, consider the tree network T in Figure 3.1. In step 1, we choose node 5 to be the root and we make the root a sink node such that T becomes a directed tree as shown in Figure 3.2(a). In step 2, we define a variable named len_x for every node x in T . In step 3, we compute len_x for every node x in T . In step 4, we show that the 2-observer of minimal cardinality of T is $\{4, 0, 3, 5\}$ in Figure 3.2(b). In addition to that, we show that the 3-observer of minimal cardinality of T is $\{4, 3\}$ in Figure 3.2(c).

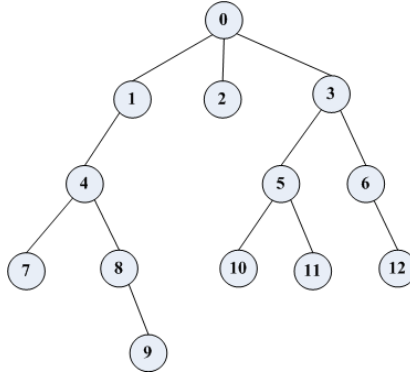


Figure 3.1: Example of a tree network T

Interestingly, the algorithm works irrespective of the choice of the root. Even if a leaf node is chosen as the root, the algorithm still correctly computes minimum K -observers in tree networks.

The time complexity of this algorithm is linear such that it is proportional to the number of nodes in the input tree network.

This algorithm as described above is centralized. But a distributed version of this algorithm can be described as follows:

Step 1:

Each node x , that has exactly one neighboring node y in the tree network T , knows that it is a leaf in T and so it assigns its variable len_x the value 0 and sends the value of its len_x to node y .

Step 2:

After a node y receives the value of len_x from every neighboring node x except one, say node z , then node y computes the value of its variable

len_y (as described in the centralized version of the algorithm) and sends the computed value of len_y to node z .

Step 3:

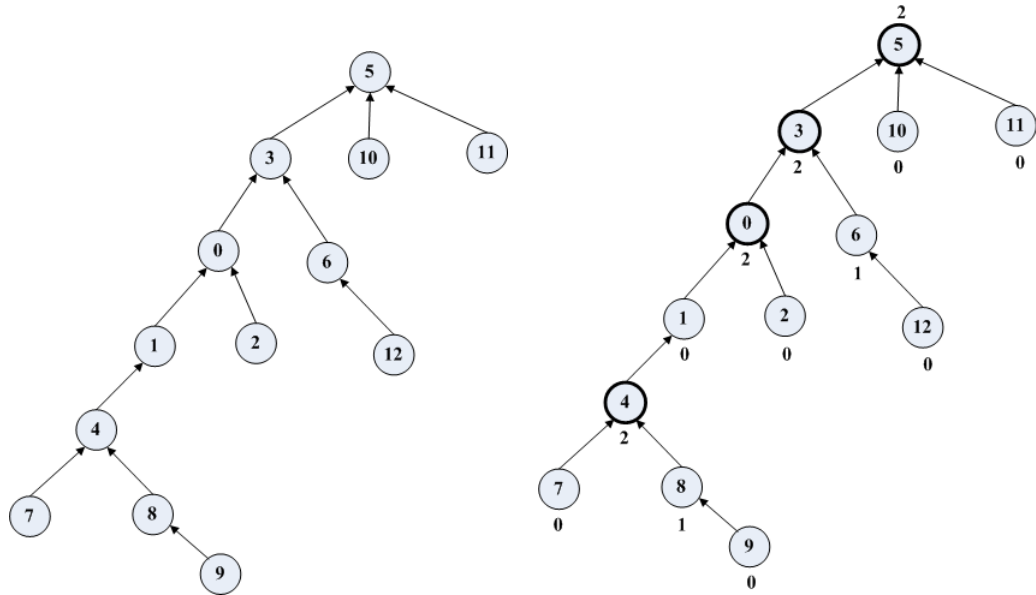
If a node z receives the value of len_y from every neighboring node y , then node z recognizes that it is the root and computes the value of its variable len_z (as described in the centralized version of the algorithm).

Step 4:

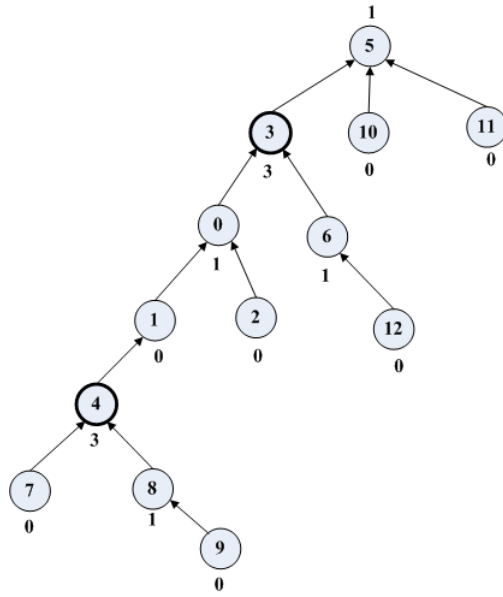
Every node x , where the value of len_x in K , knows that it is in the computed minimum K -observer P of the tree network T .

It is possible that during the execution of this distributed algorithm, two adjacent nodes y and z compute the values of their respective len variables and send their computed values to one another. Thus, each of these two nodes first sends its len value to the other node then receives the len value of the other node. In this case, the two nodes y and z behave differently depending on whether or not index y is larger than index z as follows:

- i. The node with the larger index, say node y , ignores the value of variable len_z that it receives from $node_z$ and keeps the value of its variable len_y unchanged.
- ii. The node with the smaller index, node z , recognizes that it is the root and uses the value of variable len_y received from node y to re-compute the value of its variable len_z .



(a) Making T a directed tree by choosing node 5 to be the root (b) A minimum 2-observer of T is $\{0, 3, 4, 5\}$



(c) A minimum 3-observer of T is $\{3, 4\}$

Figure 3.2: Computing minimum K -observers of T

Chapter 4

K -Observers of Ring Networks

In this chapter, we solve the K -observer problem for ring networks. Figure 4.1 shows a *ring network* with n nodes, named u_1, u_2, \dots, u_n , and n links.

Next, we describe an algorithm that computes a minimum K -observer for a ring network. This algorithm takes as input a ring network R and a positive integer K and returns as output a minimum K -observer P of network R . This algorithm consists of the following three steps:

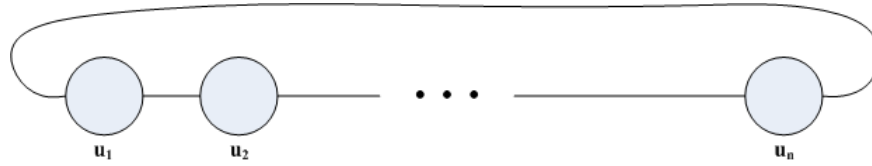


Figure 4.1: A ring network of n nodes

Step 1:

Initially, $P := \emptyset$ (the empty set)

Step 2:

if R has at most K nodes **then**

 return the empty K -observer P

```

    terminate the algorithm
else
    /*  $R$  has at least  $K + 1$  nodes */
    continue the algorithm
end if

```

Step 3:

```

    remove any node, say node  $u$ , and its two incident links from network
     $R$ 
    /* the resulting network is a tree */
    apply the algorithm in Chapter 3 to compute a minimum  $K$ -observer
     $Q$  of the resulting tree network
     $P := Q \cup \{u\}$ 
    return  $P$  and terminate the algorithm

```

Theorem 4.0.3. *If this algorithm is applied to a ring network R , and a positive integer K , then the computed set P by this algorithm is a minimum K -observer of network R .*

Note that the time complexity of this algorithm is linear in the number of nodes in the input ring network R .

Note also that if Step 2 is removed from the above algorithm, then the computed K -observer P of ring R may no longer be minimum. In this case, however, the number of nodes in P is no more than one over the number of nodes in a minimum K -observer of R . This suggests the following definition.

Let L be a non-negative integer. A K -observer P of a network N is called *L -bounded* iff the number of nodes in P is no more than L of the number of nodes in a minimum K -observer of network N . (Note that a minimum K -observer of a network N can now be regarded as a 0-bounded K -observer of N .)

Next, we describe a linear time algorithm that can be used to construct an L -bounded K -observer for a special class of networks called *L -rings*.

Let L be a positive integer. A network RR is called an *L -ring* iff RR has exactly L ring sub-networks. Note that if every ring sub-network in an L -ring is collapsed into a single super node then the resulting network is a tree.

An algorithm, for constructing an L -bounded K -observer P for any given L -ring RR , is as follows. First, identify a node in every ring sub-network in the given L -ring RR . Let u_1, \dots, u_x denote the identified nodes, where $x \leq L$. Second, remove the identified nodes and their incident links from the given L -ring RR . Note that the resulting network is a tree. Third, apply the algorithm in Chapter 3 to compute the minimum K -observer Q of the resulting tree. Compute the L -bounded K -observer P of the given L -ring RR as follows: $P := Q \cup \{u_1, \dots, u_L\}$

Chapter 5

K -Observers of Large Grid Networks

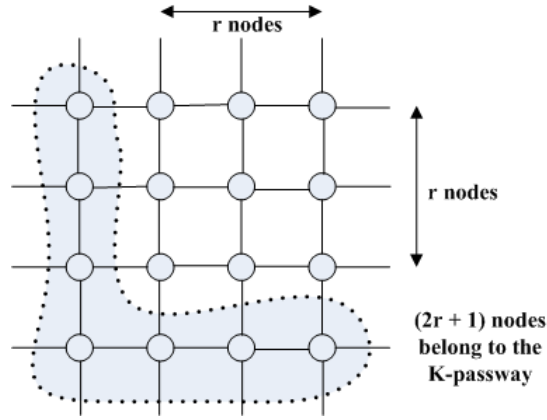


Figure 5.1: A section of $(r + 1) \times (r + 1)$ nodes in a large grid network

In this chapter, we discuss how to construct minimal K -observers for large grid networks.

Let d be a large positive integer. A d -large grid network is a network that has d^2 nodes partitioned into 4 *corner nodes*, $4d - 8$ *border nodes*, and *middle nodes*. Each corner node has 2 incident links, each border node has 3 incident links, and each middle node has 4 incident links.

Next, we present a construction of a minimal K -observer of a d -large grid network D . An interesting point about this construction is that the

number of nodes in the constructed K -observer is $O(d)$ when the value of K is $O(d^2)$.

Our construction takes as input a d -large grid network D and a positive integer K and produces as output a minimal K -observer P of network D . The number of nodes in the constructed P is $O(d^2/\lfloor\sqrt{K}\rfloor)$. The construction proceeds in the following three steps.

Step 1:

Initially, P is the empty set.

Step 2:

Partition the nodes in network D into sections, where each section has $(r + 1)^2$ nodes as shown in Figure 5.1. In Step 3 below, we argue that the value of r is $\lfloor\sqrt{K}\rfloor$. From each section, add the $2r + 1$ nodes, that form an L -shape in the section, into set P .

Step 3:

Any path in network D , whose nodes are not in set P , must be confined to a single section in D . Thus, the length of the longest path in D , whose nodes are not in P , is $r^2 - 1$. Therefore, in order to make P a K -observer of D , the value of r needs to be $\lfloor\sqrt{K}\rfloor$.

Theorem 5.0.4. *If this algorithm is applied to a d -large grid network D and a positive integer K , then the computed set P by this algorithm is a minimal K -observer of network D .*

Note that the time complexity of this algorithm is linear in the number of nodes in the input d -large grid network.

So far, we could only prove that the computed K -observer P of network D is minimal (not minimum). Luckily, we show next that the number of nodes in the computed P is relatively small when K is $O(d^2)$. But first we need to adopt the following notation.

$|P|$ is the number of nodes in computed set P .

$|D|$ is number of nodes in network D .

$section.P$ is the number of nodes from one section in computed P .

$section.D$ is the number of nodes in one section in network D .

Hence,

$$\begin{aligned}
|P| &= (section.P * |D|) / section.D \\
&= ((2r + 1) * d^2) / (r + 1)^2 \\
&\approx O(d^2 / r) \\
&\approx O(d^2 / \lfloor \sqrt{K} \rfloor)
\end{aligned}$$

Therefore, if K is $O(d^2)$, for example $K = d^2/10$, then $|P|$ is $O(d)$.

Chapter 6

Applications

The concept of minimum K -observers of a network is a fundamental idea, as it is a generalization of the concept of node cover. This is illustrated by the many applications for which this concept can be put to good use. We mention some of these applications here.

First, minimum K -observers can be used in constructing optimal connectivity paths in wireless sensor networks, wireless ad-hoc networks, and wireless ad-hoc sensor networks. For example, sensor nodes in wireless sensor networks need to maintain connectivity, but use as little energy as possible [7]. A solution to the problem would be to designate some particular nodes as targets, so the other nodes need only get a message to the nearest target node; these special nodes may have access to high power or bandwidth, and can ensure the message is rapidly delivered to its final destination. If we select nodes that constitute a minimum K -observer of the network, and deploy target nodes at these positions, we can guarantee that we choose an optimal number of target nodes to ensure that every "low-power" node has a nearby target (within K hops). Similarly, it may be interesting to choose nodes constituting the minimum K -observer of a content distribution network or of a disruption tolerant

network, and place caches at these nodes; each cache can serve the nodes in its “zone”, so with a small number of caches we ensure that every node in the network has access to a nearby cache.

Secondly, a minimum K -observer of a network can be used to measure and monitor the network traffic, as it can be considered to form a good number of well spread-out points at which to measure the parameters of the network, such as packet loss and packet delay. Such an arrangement of monitors can also be used for fine-tuning a network; for example, a service provider might deploy monitors at the nodes of a K -observer in order to identify usage accounting or bottlenecks.

Thirdly, minimum K -observers can be used to detect and prevent malicious attacks in the Internet. Thus, if we solve the K -observer problem for a computer network, it can help us to find the optimal places to place firewalls or filters in the network [12, 2, 1], to prevent IP prefix hijacking [19], or to defend against distributed denial of service (DDOS) attacks [4]. Placing “sentry” nodes forming a minimum K -observer ensures that no localized attack can cascade through the network without being detected. Determining the minimum K -observer of a network helps set up many countermeasures to limit the damage done by an attacker. Moreover, it can be used for digital forensics, to track down the adversary after an attack.

Chapter 7

Concluding Remarks

This thesis introduces the concept of minimum K -observers of computer networks. Our primary idea is that the nodes in a minimum K -observer of a network N can be employed as observers of network N . The problem of constructing a minimum K -observer for a general network is NP-hard (Chapter 2), but we demonstrate that this problem can be solved in linear-time for some special classes of networks: trees (in Chapter 3), rings (in Chapter 4), and large grids (in Chapter 5). Whether the K -observer problem can be solved for Internet-like networks, as specified in [10] and [6], remains an open, but important, question. The K -observer problem is useful in deciding the placement of firewalls, sentry nodes to detect attacks, and so on, and it would be an important contribution to develop algorithms to solve the K -observer problem (at least near-optimally) for networks in the Internet.

There is scope for considerable further work in identifying interesting classes of network for which the K -observer problem is tractable. An important open question that follows from our work is how to devise good heuristics and approximation algorithms for the K -observer problem. Furthermore, it may be noted that we have developed our theory for the K -observer problem

under the assumption that the network is stable in time and does not change; it would be interesting to investigate solutions to the problem – and, indeed, to check if the problem is solvable – for time-dependent networks, such as mobile and vehicular networks.

We find that the potential applications for the K -observer problem can be extended from the optimal connectivity for energy efficiency in wireless networks to the efficient node placement for traffic and to the detection and the prevention of network attacks in Chapter 6. We foresee that the K -observer problem will be helpful to understand and solve many problems in computer networks.

Bibliography

- [1] H. B. Acharya and Mohamed G. Gouda. Firewall modules and modular firewalls. In *Proceedings of the 18TH IEEE International Conference on Network Protocols (ICNP 2010)*, 2010.
- [2] Ehab Al-Shaer, Adel El-Atawy, and Taghrid Samak. Automated pseudo-live testing of firewall configuration enforcement. *IEEE J.Sel. A. Commun.*, 27:302–314, April 2009.
- [3] Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '02*, pages 157–164, New York, NY, USA, 2002. ACM.
- [4] Benjamin Armbruster, J. Cole Smith, and Kihong Park. A packet filter placement problem with application to defense against spoofed denial of service attacks. *European Journal of Operational Research*, 2005.
- [5] Lee Breslau, Ilias Diakonikolas, Nicholas Duffield, Yu Gu, Mohammad-Taghi Hajiaghayi, David S. Johnson, Howard Karloff, Mauricio Resende, and Subhabrata Sen. Disjoint-path facility location: Theory and prac-

- tice. In *Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2011.
- [6] Kenneth Calvert, Matthew B. Doar, Ascom Nexion, and Ellen W. Zegura. Modeling internet topology. *IEEE Communications Magazine*, 35:160–163, 1997.
 - [7] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wirel. Netw.*, 8:481–494, September 2002.
 - [8] Václav Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory*, Series B:39–41, 1975.
 - [9] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to algorithms, second edition, 2001.
 - [10] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '99, pages 251–262, New York, NY, USA, 1999. ACM.
 - [11] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

- [12] Mohamed G. Gouda and Alex X. Liu. A model of stateful firewalls and its properties. In *Proceedings of the 35th IEEE International Conference on Dependable Systems and Networks (DSN 2005)*, pages 320–327, 2005.
- [13] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, WSNA '03, pages 115–121, New York, NY, USA, 2003. ACM.
- [14] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
- [15] Dmitri Krioukov and Kevin Fall. Compact routing on internet-like graphs. In *Proceedings of IEEE INFOCOM*, pages 209–219, 2004.
- [16] Sung-Ju Lee, William Su, and Mario Gerla. Wireless ad hoc multicast routing with mobility prediction. *Mob. Netw. Appl.*, 6:351–360, August 2001.
- [17] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proceedings of IEEE INFOCOM*, pages 1380–1387, 2001.
- [18] Lili Qiu, Venkata N. Padmanabhan, and Geoffrey M. Voelker. On the placement of web server replicas. In *Proceedings of IEEE INFOCOM*, pages 1587–1596, 2001.

- [19] Tongqing Qiu, Lusheng Ji, Dan Pei, Jia Wang, and Jun Xu. Towerdefense: Deployment strategies for battling against ip prefix hijacking. In *Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP)*, October 2010.
- [20] Sabyasachi Roy, Himabindu Pucha, Zheng Zhang, Y. Charlie Hu, and Lili Qiu. Overlay node placement: Analysis, algorithms and impact on applications. *Distributed Computing Systems, International Conference on*, 0:53, 2007.
- [21] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Netw. Appl.*, 9:141–149, April 2004.
- [22] Alfred Weber. *Theory of the Location of Industries*. Chicago: The University of Chicago Press, 1909.
- [23] Jie Wu and Hailan Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '99, pages 7–14, New York, NY, USA, 1999. ACM.

Vita

Hrishikesh Bhatt Acharya was a member of the Indian team for the International Biology Olympiad in 2002, and won a silver medal. In 2006, he received the Bachelor of Technology degree in Computer Science and Engineering from the Indian Institute of Technology, Kharagpur, where he was the Aditya Birla Scholar. He then applied to the graduate program at the University of Texas, Austin, and was accepted for post-graduate study.

Permanent address: 2112 Guadalupe, Austin, Texas

This thesis was typeset with L^AT_EX[†] by the author.

[†]L^AT_EX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T_EX Program.