Optimal Inspection Points for Malicious Attack Detection in Smart Grids

Subhankar Mishra[1], Thang N. Dinh[2], My T. Thai[1], and Incheol Shin[3] *

[1] Dept. of Comp. and Info. Sci. and Eng.,
University of Florida, Gainesville, Florida 32611, USA
{mishra, mythai}@cise.ufl.edu
[2] Dept. of Comp. Sci.,
Virginia Comm. University, Richmond, VA 23284, USA
tndinh@vcu.edu
[3] Info. Security Dept.,
Mokpo National University Muan, Rep. of Korea
ishin@mokpo.ac.kr

Abstract. In this paper, we study the Optimal Inspection Points (OIP) problem, which asks us to find a subset of vertices in a given network to perform the Deep Packet Inspection so as to maximize the number of scanned packets while satisfying the delay constraints. This problem finds many applications for malicious attack detection, especially those where packet scanning is a must. Accordingly, we prove OIP is NP-complete and provide an FPTAS in the case of single path routing. For the multiple path routings, we design an FPTAS when the routing graph takes a form of series-parallel graphs, which is commonly used to model electric networks.

Keywords: Malicious Attacks Detection, Smart Grids, Optimization

1 Introduction

A key concern for the computer dependent systems is the threat from the malicious attacks which execute almost perfectly legitimate operations to compromise the whole system security. For example, in case of Distributed Denial of Service attacks in the Internet, the intrusion detection system needs to monitor the entire network traffic [1–3]. Another notable example is the Smart Grid [4], where many new classes of cyber attacks have emerged [5–8].

A common type of attack in Smart Grid is to alter the network dynamics by valid yet malicious commands [4]. To guard against this type of attack, *Deep Packet Inspection* (DPI) is essential to search for malicious packets. However, DPI leads to significant delays in the throughput hence increasing the latency for packets to arrive at the central monitoring node. Control messages not satisfying time constraints are discarded, which includes the risk of dropping important

* The first two authors contribute equally to this work.

control messages leading to serious physical and/or financial damage, therefore inspection cannot be performed at all points and on all packets.

Based on the above motivation, we introduce and study a new optimization problem, namely *Optimal Inspection Points* (OIP). Given a network represented by a graph $G = (V, E)$, the goal is to find a subset $D \subset V$ which represents the optimal inspection points, such that the number of scanned packets at the center node is maximized without violating the latency constraint. Clearly this problem helps to inspect the packets as much as possible to search for malicious ones while ensuring all packets arrive on time.

The routing schemes in different networks together with the strict latency constraints make this problem challenging and interesting. The time constraint in IEC 61850 [9–12], for example, could be as low as 3ms for the critical fault isolation and protection control messages [4]. Also the number of the scanned packets, which in turn increases the probability of catching a malicious packet, has to be as high as possible. Therefore, it would be nice if we can devise a Fully Polynomial Time Approximation Scheme (FPTAS) [14] for the OIP. Indeed, we have developed such a solution for the single path routing scenario. As for the multiple path routing, we devised an another FPTAS to OIP when the network can be transformed to a series-parallel graph.

The remainder of this paper is organized as follows. Section 2 presents the network model and our problem definition. The complexity and FPTAS are discussed in Section 3 and 4, respectively. We introduce the FPTAS for multiple-path routing in series-parallel graphs in Section 5 and provide more discussion with different scanning scenarios in Section 6.

2 Model and Problem Definitions

We use the Smart Grid as an example to illustrate the network model for our problem. A smart grid is modeled as a directed graph $G = (V, E)$ where the vertices in $V = \{r\} \cup O \cup S$ represent the set of nodes in the grid and E represents the set of communication links among the nodes.

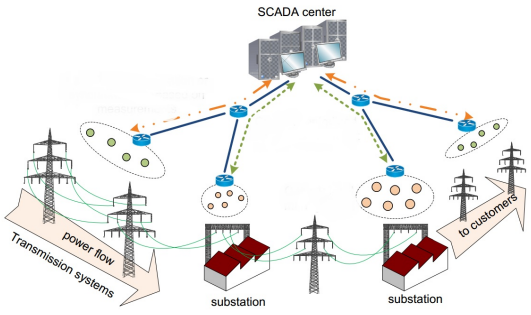

Fig. 1: A Smart Grid Structure

The set of vertices V includes the following:

- The center node r which represents the Supervisory Control And Data Acquisition (SCADA) center. All the state estimations and corresponding actions based on the message received from S are done by r. All the request packets in the smart grid communication network are routed towards r.
- S is the set of the nodes that can act as a source of malicious packets and hence can be under the control of attackers. These nodes are the Intelligent Electronic Devices (IEDs) or the Remote Terminal Units (RTUs).
- $O = V \setminus \{S \cup r\}$ is the set of intermediate nodes where DPI can be performed. If a node in O does not have DPI scanner, then equivalently the capacity of the scanner at that node is 0. We assume that there is no queueing effect and packets arrive continuously that scanners do not have to wait for packets unless the scanner capacity exceeds the amount of arrived packets.

For each $u \in V$, let $N^-(u)$ and $N^+(u)$ represent the set of incoming and outgoing neighbors of u respectively. Also let the flow $f(u, v)$ represent the network traffic from $u \to v$ (measured as the number of packets going from u to v within a time unit). Note that $f(,)$ contains the information about the routing and data forwarding in the smart grid network as follows. At a node $u \in V \setminus \{r\}$, a packet can be forwarded to any of its neighbor $v \in N^+(u)$ unless $f(u, v) = 0$. Also the probability that a packet is forwarded from u to v is proportional to $f(u, v)$, i.e., the probability is given by $\frac{f(u,v)}{f(u)}$, where

$$f(v) = \begin{cases} \sum_{u \in N^+v} f(v, u) = \sum_{w \in N^-v} f(w, v) & v \in O \\ \sum_{u \in N^+v} f(v, u) & v \in S \\ \sum_{u \in N^-v} f(u, v) & v = r. \end{cases} \tag{1}$$

For single path routing protocols, the out degree of every vertex in G is at most one. Thus G is a directed tree rooted at r. For multiple path routing protocols, a vertex in G may have multiple out going edges. In that case, we restrict our attention to the case when G is acyclic, i.e., there will be no routing loop problem in G.

We now formally define the following optimization problem:

Definition 1 (Optimal Inspection Points (OIP) problem). *Given a smart grid represented by a graph $G = (V, E)$, the center node r, the set of terminal nodes S, the set of intermediate nodes O, the capacity m_u of scanner at $u \in O$, the average traffic flow $f(u, v)$ for $(u, v) \in E$, the delay δ_u caused by DPI at $u \in O$, and the maximum delay δ_{max} for a packet. Find a subset $D \subset V$ to place scanners so that the total delay of any packet arriving at r, on any path is at most δ_{max} and the number of inspected packets is maximized.*

3 Complexity

In this section we show that OIP is NP-complete and the NP-hardness even holds for the simple path network.

Theorem 1. *The Optimal Inspection Points problem is NP-complete.*

Proof. We prove the NP-completeness of the problem even when the graph is a simple path. The decision version of OIP is defined as follows.

Decision version of OIP. Given an *acyclic* graph $G = (V, E)$, capacities m_u for $u \in V$, flow values $f(u, v), (u, v) \in E$, and maximum latency δ_{max} of a packet, is there a subset $D \subset V$ such that for any path $\mathcal{P}_v$ starting from a terminal node $v \in S$ to r,

$$\sum_{u \in D \cap \mathcal{P}_v} \delta_u \leq \delta_{max},$$

and the number of scanned packets is at least P for some $P \geq 0$?

Given a set of inspection points D, it is easy to verify in polynomial time if the total inspection time is less than the maximum delay allowed in the given system and the number of packets scanned is at least P. Hence, OIP is in NP.

To show the NP-hardness, we reduce from the $0-1$ Knapsack problem which is defined as follows. Given an instance of $0-1$ Knapsack problem with n items $a_1, a_2, \ldots, a_n$ where a_i has value v_i and weight w_i, and the bag can carry a maximum weight W. The decision version of the $0-1$ Knapsack problem asks if we can select a subset of items with total weight at most W and total value at least B, for some $B \geq 0$.

Construction. We reduce the Knapsack instance to the following instance of OIP. Construct a graph $G = (V = S \cup O \cup \{r\}, E)$ where $S = \{u_0\}, O = \{u_1, u_2, \ldots, u_{n-1}\}, r = \{u_n\}$. There is an edge (u_i, u_{i+1}) for all $i = 1 \ldots n$ (see Fig. 2). The scanner at u_i has capacity v_i and a scanning time $\delta_i = w_i$ for $i = 1 \ldots n$. The traffic flow $f(u_i, u_{i+1}) = \infty$ for $i = 0 \ldots n-1$. Set $\delta_{max} = W$ and $P = B$.

Fig. 2: Reduction from $0-1$ Knapsack.

($\rightarrow$)Suppose we have a solution $K \subset \{a_1, a_2, ..., a_n\}$ for the 0-1 Knapsack problem. Now with our construction we see that, K corresponds to a subset D of vertices O. Since, $\delta_{max} = W$, $\sum_{a_i \in K} w_i \leq W$ which implies it also satisfies the delay constraint in OIP and the number of scanned packets is at least P.

($\leftarrow$) Let say $D \subset O$ is a solution for OIP. The above solution satisfies the delay constraint δ_{max} which satisfies $\delta_{max} = W$, based on our construction, Hence $D \subset O$ is also a solution which satisfies the weight constraint and total value $B = \sum_{u_i \in D} v_i$. □

4 One Time Scan in Single Path Routing

4.1 IP Formulation

In this section, we discuss the formulation for the given problem assuming single-path-routing protocols, e.g., packets are routed following the shortest path. First,

we define the binary variable x_v for each vertex $v \in V$ as follows:

$$x_v = \begin{cases} 1 & \text{if } v \text{ is selected as an inspection point} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

For each node $u \in S$, let P_u denote the set of nodes on the unique path from u to r. The delay constraint is given by:

$$\sum_{v \in P_u} \delta_v \cdot x_v \leq \delta_{max}, u \in S \tag{3}$$

Our objective is to maximize the total number of scanned packets. The number of packets scanned at $v \in O$ is $\min\{m_v$, #unscanned packets that arrived at $v\}$. Let y_u denote the number of scanned packets going out from u, which include the packets scanned before arriving to u and also the packets scanned at u. We have

$$y_v = \min\{f(v), x_v m_v + \sum_{u \in N^-(v)} y_u\}.$$

The problem can be formulated as

$$\begin{aligned} \text{maximize} \quad & y_r & \\ \text{s.t.} \quad & \sum_{v \in P_u} \delta_v \cdot x_v \leq \delta_{max} & u \in S \\ & y_v \leq f(v) & v \notin S \\ & y_v \leq x_v m_v + \sum_{u \in N^-(v)} y_u & v \notin S \\ & y_u = 0 & u \in S \\ & x_v \in \{0, 1\} & v \in V \end{aligned}$$

4.2 FPTAS for Single-path routing one-time scanning

We give an $(1 - \epsilon)$-approximation algorithm that has an $O(\epsilon^{-2} n^5)$ time complexity for $\epsilon > 0$.

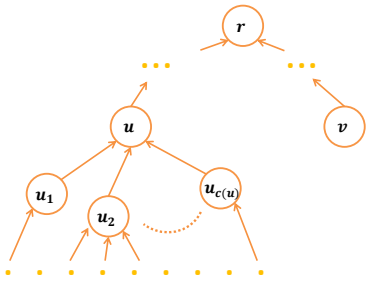

Fig. 3: In single-path-routing, the graph is a directed tree, rooted at r.

Our algorithm consists of two phases: 1) First, we standardize the capacities of the deep packet inspectors as well as the flow values so that all values are bounded by a polynomial in n and $\frac{1}{\epsilon}$; 2) Second, we use the dynamic programming to find the solution in polynomial time. The algorithm is summarized in Algorithm 1. In the first phase, the scanners' capacities m_u are standardized as shown in step 2 in Algorithm 1; then both m_u and the flow values $f(u)$ are scaled down by a factor M as defined in step 3 and rounded down. This preprocessing step ensures that all m'_u are integers between 0 and $\lceil \frac{n}{\epsilon} \rceil$.

Dynamic Programming. In the second phase, we use dynamic programming to find an optimal solution for the OIP problem instance (G, m', f'). In the case of single-path-routing, there is at most one path from each node to root node r. For simplicity, we remove nodes that have no paths to r. The remaining graph is a directed tree rooted at r as shown in Fig. 3.

Now, given the tree $T = (V, E)$ with $|V| = n$, and the SCADA center $r \in V$ serving as the root. In order to describe the dynamic algorithm, we use the following the notations:

- T^u: The subtree rooted at u in T with the set of vertices V^u and the set of edges E^u is denoted by $T^u = (V^u, E^u)$.
- $c(u)$: is the in-degree of u. Note that when referring to T^u, we disregard the edge that connect u to its parent (i.e., only consider the in-degree); thus the degree of u within T^u is one less than that in T unless $u = r$.
- $u_1, u_2, \ldots, u_{c(u)}$: represent the children of u.
- d_u: the latency is defined as $d_u = \max_{v \in N^-_{(u)}} \{d_v\} + x_u \delta_u$.
- y_u: is the number of scanned packets outgoing from u and is given $y_u = \min\{f'(u), x_u m'_u + \sum_{v \in N^-(u)} y_v\}$, where x_u indicates whether the scanner at u is switched on ($x_u = 1$) or not($x_u = 0$).

We define the following recursion functions :

- $T^u(p)$: The minimum value of latency d_u among all possible ways to deploy inspectors in the T^u rooted at u so that number of packets scanned y_u is at least p i.e. $y_u \geq p$.
- $T^u_i(p)$: The minimum value of latency d_u among the maximum value of latency among $\{d_{u_1}, d_{u_2}, \ldots, d_{u_i}\}$ among all possible ways of deploying inspectors in the subtrees $\{T^{u_1}, T^{u_2}, \ldots, T^{u_i}\}$ correspondingly, so that number of packets scanned y_u is at least p, i.e., $y_u \geq p.$, where $i = 1 \ldots c(u)$.

The core of the dynamic algorithm is to compute $T^u_{c(u)}(p)$ and $T^u(p)$ through the following recursions.

$$T^u(p) = \min\{\delta_u + \max\{T^u_{c(u)}(p - m'_u)\}, T^u_{c(u)}(p)\}, \ \forall p = 1 \ldots \lceil n^2/\epsilon \rceil \quad (4)$$

$$T^u_i(p) = \min_{q=0..p} \{\max\{T^u_{i-1}(p-q) + T^u(q)\}\}, \forall p = 1 \ldots \lceil n^2/\epsilon \rceil, i = 1 \ldots c(u) \quad (5)$$

The basis cases are as follows.

$$T^u(p) = \begin{cases} 0 & p \leq 0 \\ \infty & p > \lceil n^2/\epsilon \rceil \end{cases}, \; T_i^u(p) = \begin{cases} 0 & p \leq 0 \\ \infty & p > \lceil n^2/\epsilon \rceil \end{cases} \tag{6}$$

$$T_i^u(p) = \begin{cases} 0 & \text{if } u \in S \\ d_u & \text{if } u_1, u_2, \ldots, u_{c(u)} \in S \end{cases} \tag{7}$$

Finally, the maximum objective for the OIP instance (G, m', f') is given at the root r by $\max\{p \mid T^r(p) \leq \delta_{max}\}$.

Algorithm 1. FPTAS for Single-path-routing OIP

Phase 1: Preprosessing

1. Remove all nodes that have no paths to r.
2. For all $u \in V$, if $\delta_u > \delta_{max}$, set $m_u \leftarrow 0$; else set $m_u = \min\{m_u, f(u)\}$.
3. Given $\epsilon > 0$, let $K = \frac{\epsilon M}{n}$, where $M = \max_{u \in V}\{m_u\}$.
4. Let $f'(u) = \lfloor \frac{f(u)}{K} \rfloor$ and $m'_u = \lfloor \frac{m_u}{K} \rfloor$

Phase 2: Dynamic programming algorithm

5. Compute $T^u(p)$ and $T_i^u(p)$ using the recursions in Eqs. 4 and 5.
6. Find an optimal solution, say S', by tracing from $\max\{p|\ T^r(p) \leq \delta_{max}\}$
7. Return S'.

Lemma 1. *Algorithm 1 finds an optimal solution for the single-path-routing OIP instance (G, m', f') in an $O(\epsilon^{-2}n^5)$ time.*

Proof. The correctness of the dynamic programming algorithm comes from the sub-optimal structure of the problem.

As for the running time, the major portion of running time is to compute $T_i^u(p)$. Since we have at most $n-1$ possible pairs of u and i (the total number of children), and $q \leq p \leq \lceil n^2/\epsilon \rceil$. The running time to compute $T_i^u(p)$ is $O(n \times \lceil n^2/\epsilon \rceil \times \lceil n^2/\epsilon \rceil) = O(\epsilon^{-2}n^5)$. □

Theorem 2. *For any $\epsilon > 0$, there is an $(1-\epsilon)$-approximation algorithm for the OIP problem, single-path-routing with a time complexity $O(\frac{1}{\epsilon^2}n^5)$.*

Proof. Let $S^* \subseteq V$ be an optimal solution of the OIP instance (G, m, f) with the objective value $OPT = y_r(S^*)$ (the total number of scanned packets at any nodes).

Given $1 > \epsilon > 0$, we apply Algorithm 1 to find an optimal solution $S' \subseteq V$ for the instance (G, m', f') with an objective value $OPT' = y'_r(S')$. By Lemma 1, this takes a (polynomial) running time $O(\epsilon^{-2}n^5)$.

First, S' is also a feasible solution for the instance (G, m, f), since it satisfies the condition that the latency at r is at most δ_{max}. Let $y_r(S')$ be the objective value associated with S' w.r.t. the instance (G, m, f). We will show that

$$y_r(S') \geq (1 - \epsilon)OPT.$$

The dynamic programming must return a solution at least as good as S^* (for the OIP instance (G, m', f')). Thus $y'_r(S^*)$, the objective value associated with S^* w.r.t. the instance (G, m', f'), is at most OPT'. Due to the rounding down, Mm'_u can be smaller than m_u, but by not more than K. Hence,

$$y_r(S^*) - My'_r(S^*) \leq nK.$$

Therefore

$$y_r(S') \geq Ky'_r(S') \geq Ky'_r(S^*) \geq y_r(S^*) - nK = OPT - \epsilon M.$$

Since we filtered out nodes u with $\delta_u > \delta_{max}$, we have $OPT \geq M$. Therefore,

$$y_r(S') \geq OPT - \epsilon M \geq (1 - \epsilon)OPT.$$

Thus the objective of S' is within a factor $1 - \epsilon$ of OPT, i.e., Algorithm 1 is a $(1 - \epsilon)$ approximation algorithm for the single-path-routing OIP problem. □

5 One Time Scan in Multiple Paths Routing

In this section, we study the OIP problem in which packets can be routed using different paths to the SCADA center. We present an IP formulation for the problem in Section 5.1. We study the special case when the network has form of a series-parallel graph, which is often used to model electric networks. Accordingly, we present the definition of series-parallel graphs (SP-graphs) in Section 5.2 and describe an FPTAS for Multi-path routing OIP in SP-graphs in Section 5.3.

5.1 IP Formulation

We present the formulation for the given problem when multi-path routing protocols are in use. Let $x_u = 1$ if node u is selected as an inspection point and 0, otherwise. Also, let l_u denote the latency, the maximum possible delay of a packet, at node u. Then l_u is given by

$$l_u = \max_{v \in N^-(u)} \{l_v\} + x_u \delta_u \quad \forall u \notin S \tag{8}$$

Thus the total delay constraint is $l_r \leq \delta_{max}$.

Since the probability that a packet is forwarded from u to v is proportional to $f(u, v)$, the number of scanned packets going out from v is given by

$$y_v = \min\{f(v), x_v m_v + \sum_{u \in N^- v} y_u \frac{f(u, v)}{f(u)}\}.$$

The OIP problem with multi-path routing can be formulated as follows.

$$
\begin{aligned}
\text{maximize} \quad & y_r \\
\text{s.t.} \quad & l_r \le \delta_{max} \\
& l_v \ge l_u + x_v \delta_v && v \notin S, u \in N^-(v) \\
& y_v \le f(v) && v \notin S \\
& y_v \le x_v m_v + \sum_{u \in N^- v} y_u \frac{f(u,v)}{f(u)} && v \notin S \\
& y_u = 0, l_u = 0 && u \in S \\
& x_v \in \{0,1\} && v \in V
\end{aligned}
$$

It is common that the network traffic is much higher than the capacities of scanners, thus if we choose to activate the scanner at u, we can scan exactly m_u additional packets. Thus the above formulation can be simplified to

$$
\begin{aligned}
\text{maximize} \quad & x_u m_u \\
\text{s.t.} \quad & l_r \le \delta_{max} \\
& l_u = 0 && \forall u \in S \\
& l_v \ge l_u + x_v m_v && \forall v \notin S, u \in N^-(v) \\
& x_v \in \{0,1\} && \forall v \notin S.
\end{aligned}
$$

5.2 Series-parallel Graphs (SP-graphs)

Given two graphs G_1 and G_2 together with pairs of source-sink nodes (s_1, t_1) in G_1 and (s_2, t_2) in G_2, a *series composition* creates a new graph by merging the sink t_1 and the source s_2 and (s_1, t_2) becomes the source-sink of the composed graph. A *parallel composition* creates a new graph by merging two sources s_1 and s_2 into the new source, and two sinks t_1 and t_2 into the new sink node. A series-parallel graph (SP-graph) is a graph that is constructed by a sequence of series and parallel compositions starting from a set of single-edge graphs, i.e. cliques of size two.

An SP-graph G with a source-sink pair (s, t) can be decomposed into several single-edge base graphs. The decomposition is specified by a binary decomposition tree $T(G)$ whose nodes represent subgraphs of G. Each non-leaf node of the tree has two child subgraphs and an associated operation (either series or parallel). The parent subgraph can be constructed by applying the operation on two children subgraphs. Construction of $T(G')$ can be done in a linear time complexity [13].

5.3 FPTAS for Multi-path routing one-time scanning in GSP

We present an FPTAS for the OIP problem when the graph has form of an SP-graph. Specifically, let G' be an augment of G by adding a source node s and connecting s to all nodes in S. We set $f(s, u) = f(u)$ for each $u \in S$ and $m_s = 0$

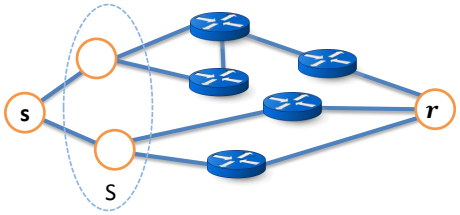

Fig. 4: A series-parallel network. Bold nodes are intermediate nodes in O.

(Fig. 4). The OIP problems G' with source-sink pair (s, r) is then equivalent to the OIP problem on G.

We present an FPTAS for the OIP problem under the following assumptions.

- Multi-path routing is possible and the augmented graph G' is an SP-graph.
- Single scanning mechanism is employed.
- The scanner capacities are relatively small to the number of arriving packets so that even when all scanners are in use, there are still unscanned packets at every node.

The last assumptions simplify the calculation of the number of scanned packets to the total capacities of deployed scanners.

We summarize the FPTAS in Algorithm 2. We begin by constructing the decomposition tree $T(G')$ [13]. Also, the scanners' capacities are scaled and rounded to the nearest integers in $[0...\lceil \frac{n}{\epsilon} \rceil]$. Then we solve the problem in each subgraph of G' starting from the leaf nodes of $T(G')$ up to the root via a set of series and parallel merging operations.

For any subgraph $\bar{G}$ with a source-sink pair $(\bar{s}, \bar{t})$, we define a function $h_{\bar{G}}(p)$ as the *minimum latency* among all possible way to deploy scanners on $\bar{G}$ so that the *total capacities of deployed scanners is at least* p. We enforce that no scanner is deployed at $\bar{s}$ to avoid duplicate scanner deployment for the series operation.

The basis. When $\bar{G}$ is a base graph with a single edge $(\bar{s}, \bar{t})$, we have

$$h_{\bar{G}}(p) = \begin{cases} 0 & p = 0 \\ \delta_{\bar{t}} & p = 1...m_{\bar{t}} \\ \infty & p > m_{\bar{t}} \end{cases}$$

The Series Operation. Assume that the subgraph $\bar{G}$ with source-sink pair $(\bar{s}, \bar{t})$ is obtained by applying *series* operation on the subgraph G_1 with the source-sink pair (s_1, t_1) and the subgraph G_2 with the source-sink pair (s_2, t_2). For a series operation, both the latency and the number of scanned packets in $\bar{G}$ are equal to those of G_1 and G_2. Thus

$$h_{\bar{G}}(p) = \min_{p'=0...p} \{h_{G_1}(p') + h_{G_2}(p - p')\}, 0 \leq p \leq \sum_{u \in \bar{G}} m'_u$$

Parallel operation. For a *parallel* operation, the latency in $\bar{G}$ is equal to the maximum of the latency in G_1 and G_2 and the number of scanned packets

in $\bar{G}$ is the sum of those in G_1 and G_2. Hence

$$h_{\bar{G}}(p) = \min_{p'=0\ldots p} \max\{h_{G_1}(p'), h_{G_2}(p-p')\}, 0 \le p \le \sum_{u\in\bar{G}} m'_u$$

Algorithm 2. FPTAS for Multi-path-routing OIP in SP-graphs

1. Construct $G' = (V', E')$ by adding a node s to G and setting $f(s,u) = f(u), u \in S, m_s = 0$.
2. Scale m_u: Let $K = \frac{\epsilon M}{n}$, where $M = \max_{u\in V}\{m_u\}$. Set $m'_u = \lfloor \frac{m_u}{K} \rfloor$.
3. Construct the decomposition tree $T(G')$.
4. Starting from leaf-node in $T(G')$ up to the root, compute the $h_{\bar{S}}(p)$ for each subgraph $\bar{G}$ in $T(G')$ using the formulations for series and parallel operations.
5. At the root node of $T(G')$, choose a solution with the maximum p value that $h_{G'}(p) \le \delta_{max}$.

Time Complexity Analysis. There are no more than n operations (either series or parallel). Since $\sum_{m'_u} \le n \times \lceil n/\epsilon \rceil$, we need to compute $h_{\bar{G}}(p)$ for at most $O(n^2/\epsilon)$ different values of p, which, in turn, requires an $O(n^2/\epsilon)$ time. Therefore, the total time complexity is $O(n^5/\epsilon^2)$.

Theorem 3. *For $\epsilon > 0$, Algorithm 2 is a $(1-\epsilon)$-approximation algorithm for the multi-path routing OIP problem when the augmented graph G' is an SP-graph.*

Proof. Let $S^* \subseteq V$ be an optimal solution of the OIP instance (G, m, f) with the objective value $OPT = \sum_{u\in S^*} m_u$ and S' be the optimal solution of the OIP instance (G, m', f') found in Algorithm 2.

Since S^* satisfies the latency constraint, it is also a feasible solution for the instance (G, m', f'). From the optimality of S', we have

$$\sum_{u\in S^*} m'_u \le \sum_{v\in S'} m'_v.$$

Let $OPT' = \sum_{v\in S'} m_v$, we will show that $OPT' \ge (1-\epsilon)OPT$. Thus the objective value given by S' is at least a $(1-\epsilon)$ times the optimal objective value.

We have

$$\begin{aligned} OPT' &= \sum_{v\in S'} m_v \ge \sum_{v\in S'} K m'_v \ge K \sum_{u\in S^*} m'_u \\ &\ge \sum_{v\in S'} (m_v - K) \ge OPT - nM \ge (1-\epsilon)OPT \end{aligned}$$

Thus Algorithm 2 gives an $(1-\epsilon)$ approximation algorithm for the OIP problem when G' is an SP-graph. □

6 Discussion

In this paper we assume that each packet will not be scanned multiple times. This can be implemented by altering the packet header to add one flag to check whether the packet has been scanned. This approach requires updating either the hardware/firmware components at the network core.

We can also relax this scanning requirement and do not check for multiple scanning of packets. This approach provides greater compatibility for legacy devices with a cost in efficacy (due to redundant scanning). Using this approach we can also formulate two new optimization problems depending on whether single path or multiple path routing is in use. These alternative formulations are more difficult than their one time scanning counterparts and are subjects of our further studies.

7 Acknowledgment

This work is partially supported by DTRA YIP #HDTRA1-09-1-0061.

References

1. J. Mirkovic and P. Reiher *A Taxonomy of DDoS Attack and DDoS Defense Mechanisms*, in ACM SIGCOMM, 2004.
2. T. Peng, C. Leckie, and K. Ramamohanarao, *Survey of Network-Based Defense Mechanisms Countering the DoS And DDoS Problems*, ACM Comput. Surv. 39, 1, Article 3, 2007.
3. Y. Kim, W.C. Lau, M.C. Chuah, and H.J. Chao, *PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks*, IEEE Trans. Dependable Secur. Comput., 3(2):141-155, 2006.
4. Wenye Wanga, Zhuo Lu, *Cyber Security in the Smart Grid: Survey and Challenges*, Computer Networks (57) 2013 1344-1371.
5. L. H. Jeffrey, H. G. James, C. P. Sandip,*Cyber security enhancements for SCADA and DCS systems*, Technical Report TR-ISRL-07-02, University of Louisville (2007) 127.
6. A. Hamieh, J. Ben-Othman, *Detection of jamming attacks in wireless ad hoc networks using error distribution*, in: Proc. of IEEE ICC 09, 2009.
7. D. Choi, H. Kim, D. Won, S. Kim, Advanced key-management architecture for secure SCADA communications, IEEE Trans. Power Delivery 24 (2009) 11541163.
8. D. Choi, S. Lee, D. Won, S. Kim, Efficient secure group communications for SCADA, IEEE Trans. Power Delivery 25 (2010) 714722.
9. R. E. Mackiewicz, *Overview of IEC 61850 and Benefits*, in Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES, pp 623 - 630.
10. S. Mohagheghi, J. Stoupis, Z. Wang,*Communication protocols and networks for power systems - current status and future trends*, in: Proc. of Power Systems Conference and Exposition (PES 09), 2009.
11. IEC Standard, IEC 62351: Data and communication security.
12. Christoph Brunner, *IEC 61850 for power system communication*, in Transmission and Distribution Conference and Exposition IEEE/PES, 2008, pp 1 - 6.
13. Takamizawa, K. and Nishizeki, T. and Saito, N., *Linear-time Computability of Combinatorial Problems on Series-parallel Graphs*, in Journal of ACM, 1982, pp 623-641.
14. Vazirani, Vijay V., *Approximation Algorithms*, Springer-Verlag New York, Inc., New York, NY, USA, 2001.