Communication Links Vulnerability Model for Cyber Security Mitigation

Eman Hammad(✉), Abdallah Farraj, and Deepa Kundur

Department of Electrical and Computer Engineering,
University of Toronto, Toronto, Canada
{ehammad,abdallah,dkundur}@ece.utoronto.ca

Abstract. We consider the problem of defining a metric to capture communication links vulnerability that is a function of threat models of concern. The model is based on the Confidentiality-Integrity-Availability (C-I-A) framework and combines communication links parametric models with dynamical historical models. The proposed model arrives at a vulnerability matrix to describe the cyber component of a cyber-physical system. The vulnerability matrix is used for flexible adaptive constrained routing implemented on Software Defined Networks (SDNs) as a mitigation approach for threats of concern.

Keywords: Smart grid · Cyber security · Metric · Vulnerability · Constrained shortest path · Quality of Service · Routing · Software defined networks

1 Introduction

Cyber security is perceived as a challenge on different levels of cyber-physical systems such as the smart grid. This is, in part, due to the fact that current standard communication protocols were not designed with a cyber-security perspective. Complex interconnected cyber-physical systems offer a multitude of opportunities and challenges within the cyber security context. Vulnerabilities resulting from the cyber enablement of the physical system may not be fully uncovered or understood, and this unveils many challenges into how to reenforce the system against the unknown. A distinguishing characteristic of the cyber-physical systems (e.g. smart grid) is that cyber-security approaches cannot be considered without studying their impact on real-time operations of the physical system.

One of the coupled interactions in the smart grid is between communication network infrastructure and cyber-enabled control; in this context developing a functional cyber-security assessment framework that can be used for flexible cyber-physical mitigation approaches is still lacking. Information Technology (IT) based security did not prove to be efficient due to its focus on the cyber plane of the system. This motivated the emergence of several impact based cyber-security frameworks; of which the C-I-A (Confidentiality-Integrity-Availability)

Y. Zhou and T. Kunz (Eds.): ADHOCNETS 2016, LNICST 184, pp. 285–296, 2017.
DOI: 10.1007/978-3-319-51204-4_23

framework has prevailed as an adequate tool for high level cyber-security impact analysis in cyber-physical systems.

Given a certain cyber infrastructure for a cyber-physical system such as the smart grid, system engineers and operator should be able to asses the existing system according to various cyber-security threat models. This establishes a baseline for system security planning and testing: highlighting weaknesses in the system, and providing guidance and input to different mitigation schemes. Previously, authors have developed a communication link vulnerability mitigation framework that satisfies Quality of Service (QoS) constraints of the underlying power system. The mitigation framework is enabled through the utilization of Software Defined Networks (SDN), as a flexible adaptive communication infrastructure and control platform. The problem was formulated as a constrained shortest path routing problem, that optimizes for the least vulnerable route with satisfactory QoS (delay). A model for the vulnerability of a communication link was roughly outlined. In this work we further develop the proposed communication link vulnerability. The proposed vulnerability model is directly assessed in a dynamic constrained QoS routing setting.

Ad hoc networks are considered a promising solution for networking on the distribution level [7], specifically for smart-metering systems and applications. Moreover, previous works have considered optimizing ad hoc network management using SDN [5,8]. This suggests that the proposed vulnerability metric could be extended to ad hoc networks, with proper parameters pertinent to the network specifics and operation.

The main contributions of this work include the following:

1. propose and develop a vulnerability metric model for communication links in cyber-physical systems,
2. employ the proposed vulnerability metric via an SDN based adaptive QoS routing.

2 System Model

Let N denote the number of nodes in the power system; for this discussion let N refer to number of buses in the power grid. Then, we can assume a communication network connecting the N buses in a topology that parallels that of the electrical grid.

Consider a graph representation of the corresponding communication network. The weighted undirected graph model $G(V, E, w)$ describes an N-node and M-link network, where the node set $V = \{v_1, \ldots, v_N\}$ and the edge set $E = \{e_{ij}, i, j = 1, 2, \ldots, M\}$ denote the buses and communication links, respectively. The weight w on the edge between two nodes is defined as the cost of the corresponding communication link. Then, the adjacency matrix A can be defined as

$$A_{i,j} = \begin{cases} w_{ij} \; i \neq j, \text{ for } (i,j) \in E \\ 0 \quad \text{otherwise.} \end{cases} \tag{1}$$

Consider next the routing problem of communicating data between a source node s and destination node t in the graph G. The shortest path route between the pair can be found using various algorithms. Due to its simplicity and optimality, Dijkstra-based routing algorithm has long been the most used algorithm to arrive at the shortest path.

The SDN framework allows us to obtain a dynamic delay cost matrix A_d sampled from the network at pre-defined intervals. Similarly, provided that a vulnerability cost metric is defined, then a corresponding vulnerability cost matrix A_v can be evaluated for the network. Hence, the problem of QoS routing while mitigating link vulnerabilities is then formulated as a constraint shortest path (MCSP) problem.

Within the smart grid, cyber-enabled control systems require information delivery between relevant nodes with certain delay requirements; as an example, the IEC 61850 GOOSE messaging specifies the message delay constrains for performance class P2/3 to be within 3 ms [2]. Accordingly, if we define the delay cost matrix A_d, then the problem of finding paths that satisfy the delay constraints can be formulated as a constraint shortest path (CSP) problem. While the CSP is NP-hard, many algorithms have been developed to find a feasible or a set of feasible solutions [16].

In the context of smart grid systems, networked sensory and control impose many constraints on data communications; nevertheless, in this paper we are more focused on the development of a tractable vulnerability metric for communication links. The proposed vulnerability metric will be utilized for various threat models (Fig. 1).

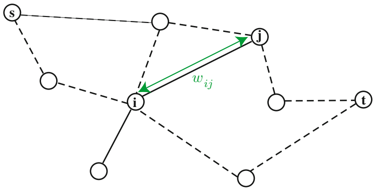


Fig. 1. Communication network graph

2.1 Vulnerability Metric

The increased cyber coupling of the smart grid through more cyber-enabled sensory and control increases the vulnerability surface of the smart grid. The Confidentiality-Integrity-Availability (C-I-A) framework provides a neat classification of vulnerabilities based to their impact with respect to information. In this work, we consider a vulnerability threat model directly related to the C-I-A framework, where threats and attacks can be classified using the aforementioned framework. Developing a metric that captures the elements affecting the vulnerability level of a certain communication link will enable us to develop a corresponding network response. It is intuitive that a link vulnerability is not a binary characteristic; i.e., a simple label of a link as vulnerable or not-vulnerable is not very informative for system operation.

The communication link vulnerability metric proposed in this work satisfies on the following "security metric" properties [14]: a security metric should be quantitative, objective, employs a formal model, not boolean (0, 1), and reflects time dependance [14]. Further, it would be advantageous if the metric self-reflects the associated risk level to better provide an insight.

We next try to formalize our definition of vulnerability from a cyber-security perspective; based on several definitions from information technology and computing, to disaster management; the vulnerability of a system or group of systems is defined as a weakness in that system that hinders its ability to withstand threats [3]. Extending this definition to communication networks in cyber-physical systems, leads us to consider the attributes and installed mechanisms to arrive at a measure relating how vulnerable communication links are to threat models of concern. In a smart grid's communication network infrastructure, few of the communication link attributes can be combined to describe and quantify a link vulnerability metric. These attributes can be grouped into categories as shown in Fig. 2, and as follows

1. **Dynamic**; attributes in this category dynamically vary over time and in response to events.
   - History L_H^{ij}; a link that was previously targeted by an attacker is more likely to be targeted again by a passive/active adversary.
2. **Parametric**, attributes in this category tend to be static and are characteristic of the link. The parameters are scored and ranked in a hierarchical fashion, in the first layer the threat model is linked to the parameters through an impact analysis based on the C-I-A impact framework, and in the second level each of the link parameter sets is internally ordered based on relative vulnerability preference to establish set weights.

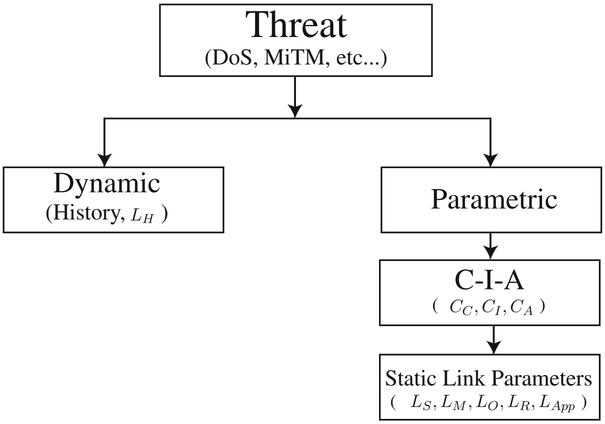


Fig. 2. Link vulnerability model

i. C-I-A
   - C_C: cost on confidentiality;
   - C_I: cost on integrity;
   - C_A: cost on availability.

ii. Ordered static link parameters, and these could include the following among others.
   - L_S^{ij}: security installed measures: a link with strong encryption is typically less vulnerable,
   - L_M^{ij}: physical-channel modality: a wireless link is usually more vulnerable than a fiber optic due to technology,
   - L_O^{ij}: ownership: a self-owned and operated channel is typically less vulnerable than a shared and/or leased channel,
   - L_R^{ij}: redundancy: a link with installed redundancy mechanisms is less vulnerable compared to a single link,
   - L_A^{ij}: application: a link dedicated for command/control traffic is more vulnerable to being targeted.

It is important to acknowledge the existence of interdependencies between the various parameters, yet we argue that virtually decoupling the interdependencies as a list is informative for our problem of interest. Link history L_H is modeled by a Markov model as is described in Sect. 2.2. Moreover, link parameters $\{L_S, L_M, L_O, L_R, L_A, \ldots\}$ are subsets with embedded monotonically increasing ordering where a higher order is related to a more vulnerable state. The subsets are constructed by system engineers and operators where an exhaustive enumeration of the related parameter possible values is established with the proper ordering. Let the ordered subset related to a generic parameter L_x be denoted $\mathbb{X}$, then a plausible mapping of the set entries to a normalized weight is defined by

$$X_w = \frac{1}{|\mathbb{X}|} \times \{X_1, X_2, \ldots\} \tag{2}$$

Lets consider link security as an example: the set $\mathbb{S}$ corresponding to L_S includes a ranking of the different installed and configured security measures. For simplicity let a subset of security measures include three different encryption mechanisms $\{Encr_1, Encr_2, Encr_3\}$, where corresponding L_S will be assigned based on the strength of the encryption. Let $\rhd$ denote a security comparison operator where left hand is a stronger encryption than the right hand operator, then if $Encr_1 \rhd Encr_2 \rhd Encr_3$, a possible assignment could be $L_S^{ij} \in \{0, 0.33, 0.66\}$.

A similar approach can be used to arrive at the communication link parameters subsets and their corresponding weights. We next consider how to combine these parameters into a single representative vulnerability metric (L_V^{ij}). The vulnerability metric should reflect the attributes that make a link more probable to be targeted by an attempted adversary action, as well as being affected by that targeting. We base our vulnerability metric definition on the C-I-A framework threat model. Consider the threat set Γ, where the set could include threats such as denial of service (DoS), false data injection (FDI), among others.

And consider a general link parameter ordered subset $\mathbb{X} = \{X_1, X_2, X_3\}$. A threat $\varGamma_m$ can be decomposed using the C-I-A model into a three part binary indicator based on the threat cost/impact, as is shown below

$$\mathbf{\Gamma_m} = \mathbb{1}_m^{C-I-A} = [\mathbb{1}_m(C_A) \quad \mathbb{1}_m(C_I) \quad \mathbb{1}_m(C_C)] \tag{3}$$

Further, as shown in Fig. 2, a link vulnerability can be modeled as a function of dynamic link history L_H and parametric vulnerability L_P of the link.

$$L_V^{ij}(\varGamma_m) = L_H^{ij}(\varGamma_m) + L_P^{ij}(\varGamma_m) \tag{4}$$

where L_P, the parametric link vulnerability is further developed below.

This indicator is then embedded in the scoring of the ordered parameters subsets $\mathbb{X}$ to arrive at the communication link parametric vulnerability $L_P^{ij}(\varGamma_m)$.

To facilitate the threat model embedded scoring of link parameters, ordered parameter subsets are further clustered into sub-subsets according to their correlation with the C-I-A decomposition of any threat. I.e consider the following clustering of the ordered subset $\tilde{\mathbb{X}} = \mathbb{X}_C \cup \mathbb{X}_I \cup \mathbb{X}_A$. This is extended to link parameter subsets $\tilde{\mathbb{P}} = \{\tilde{\mathbb{S}}, \tilde{\mathbb{M}}, \tilde{\mathbb{O}}, \tilde{\mathbb{R}}, \tilde{\mathbb{A}}, \ldots\}$ This leads us to the following model for the parametric link vulnerability (the superscript ij have been removed for clarity, wherever L is presented it is related to communication link ij

$$\begin{aligned} L_P(\varGamma_m) &= L_S + L_M + L_O + L_R + L_A + \ldots \\ L_P(\varGamma_m) &= \mathbf{\Gamma_m} \cdot \tilde{\mathbb{P}} \end{aligned} \tag{5}$$

$$L_P(\varGamma_m) = \frac{1}{|\mathbb{P}|} \sum_{n=1}^{|\mathbb{P}|} \left([\mathbb{1}_m(C_A) \quad \mathbb{1}_m(C_I) \quad \mathbb{1}_m(C_C)] \cdot \begin{bmatrix} \mathbb{S}_C & \mathbb{M}_C & \mathbb{O}_C & \ldots \\ \mathbb{S}_I & \mathbb{M}_I & \mathbb{O}_I & \ldots \\ \mathbb{S}_A & \mathbb{M}_A & \mathbb{O}_A & \ldots \end{bmatrix} \right) \tag{6}$$

To further illustrate this proposal, if the threat model is focused on DoS attacks, then the attack/threat decomposition vector will explicitly model the DoS as $\varGamma_m = [1 \quad 0 \quad 0]$, in terms of direct cost on availability. Similarly, the corresponding link parametric sub-subsets will identify the relevant components from each parameter subset that will be part of the vulnerability metric.

Finally, necessary normalization is performed when combining the different metrics, and the vulnerability cost matrix for the whole communication network is constructed such that $A_v^{ij} \in [0–100]$. We next discuss the dynamic link history vulnerability.

2.2 Link History

Most of the communication link attributes considered above are static and do not change with time, unless advanced functionalities are installed such as service-adaptive cryptography levels. However, vulnerability history of a link is affected by events and status of the network; thus a link history L_H is best modeled by a dynamical model. It is important to note that the goal of the proposed dynamical model is to provide a tool to quantify the probability of a communication link in the system being targeted based on detected events history. However, we do not intend to provide an intrusion detection/prediction capability. For our purposes it suffices to consider a Markov model, which is simply a system with one step history. On the other hand, a longer history of the link would be beneficial to tune the Markov model parameters (i.e., transition probabilities) [1,17].

Let the event of "targeting" the communication link between nodes i and j be a stochastic event that happens with a probability P_A^{ij}. Further, let the link status be termed as $S_{ij} \in \{G, T\}$, where G denotes a good link and T denotes a targeted link, and is modeled by a Markov Model. A finite-state Markov chain process is described by its transition matrix P where the $P(l,m)$ element is defined as the probability of state $X^{k+1} = m$ given that the previous state is $X^k = l$. This is commonly known as the Markov property where the next state of the system depends only on the current state, and is described as $P(l,m) = \mathbb{P}(X^{k+1} = m | X^k = l)$.

The transition matrix P for the 2-state model is mathematically described by the following probabilities

$$\begin{aligned} \mathbb{P}(L_{ij}^{k+1} = G \mid L_{ij}^{k} = T) &= P_D \\ \mathbb{P}(L_{ij}^{k+1} = T \mid L_{ij}^{k} = G) &= P_A \\ \mathbb{P}(L_{ij}^{k+1} = G \mid L_{ij}^{k} = G) &= 1 - P_A \\ \mathbb{P}(L_{ij}^{k+1} = T \mid L_{ij}^{k} = T) &= 1 - P_D \end{aligned} \tag{7}$$

and $P = \begin{bmatrix} 1 - P_A & P_A \\ P_D & 1 - P_D \end{bmatrix}$.

Given that the Markov chain described above is time-homogeneous, then we consider the stationary equilibrium/distribution of the chain for important insights such as the probability of being in a certain state. Specifically, we can obtain the probability that the communication link will be in state T (i.e., targeted by an attacker) L_H^{ij}. This is then combined with the parametric attributes of the channel according to Eq. (4) to arrive at the vulnerability metric of the link, L_V^{ij}.

2.3 Sustainable Security

Above referenced communication link parameters are mostly characteristic of the communication network, its usage and any ancillary additions to it. From a cyber-security perspective, system engineers and administrators are responsible for tracking the various network components and their corresponding configurations. Based on this perspective, the proposed vulnerability model assumes that system administrators should be able to log communication links respective parameters and flag each parameters sub-subcomponent relationship with the C-I-A framework. Further, it assumes a fundamental understanding of the scope and limits of each installed and configured component, their relative ordering with respective to vulnerability, and the planned/unplanned interactions between the cyber-physical components and the cyber-cyber components.

The usefulness of the proposed metric relies on a sustainable cyber-security environment [13,15]; which can be described by two characteristics: (1) the establishment of a cyber-security eco-system where validation and frequent updates are regulated to ensure up-do-date match between envisioned and actual system state. This is necessary for any algorithms or network defined functionality such as the one proposed by the authors via adaptive routing. (2) existence of defined policies and system procedures for active recovery and mitigation feedback, where system engineers continuously adapt by applying necessary measures to ensure a minimum future risk.

Further, the proposed model can be used as a tool for cyber-security assessment of the communication network in use. As it will pinpoint the most vulnerable links based on system configuration and dynamic history of cyber events. This assessment is helpful to (1) sketch a system update-upgrade plan (2) develop a cyber-security monitoring procedure/application check points, (3) revise response/recovery procedures. An Autonomous cyber-security system is a future vision that will require tremendous intelligence and adaptability, and is probably a threat to itself.

3 Software Defined Networks and Adaptive Constrained Routing

Provided that we can obtain an updated communication network vulnerability matrix A_v that is regularly updated, then we can adaptively route information based on a set criteria. The previous is valid if we have a communication network paradigm/architecture that is able to: (1) have a dynamically updated global network state, (2) be programmed with additional intelligence to control network traffic, (3) be managed and configured with low complexity. Software Defined Networks (SDN) is a very promising network architecture that is capable of supporting and enabling our adaptive routing. Moreover, it allows a more complicated processing to optimize vulnerable link avoidance to minimize both delay (of extreme importance in smart grid) and information leakage through vulnerable links.

3.1 Software Defined Networks

Software defined networking offers the potential to change the traditional way networks operate. Current communication networks are typically built from a large number of network devices, with many complex protocols implemented on them. Operators in traditional communication networks are responsible for configuring policies to respond to a wide range of network events and applications. Consequently, network management and performance tuning is quite challenging and error-prone [1,10].

The main characteristic of SDN, is the separation of control and data planes, where the network is decomposed to an SDN controller and various SDN data forwarding switches. This architecture enables revolutionary approaches to network

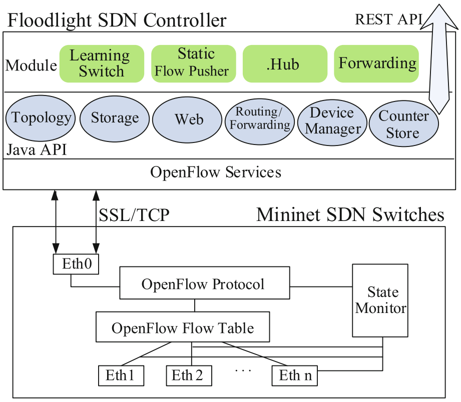


Fig. 3. SDN architecture

management, including adaptive networks that can dynamically be configured and programmed to respond to changes in the network. Network layer applications can acquire detailed traffic statistics from network devices to construct an up-to-date network view. One common standard for the implementation of software defined networks is OpenFlow [11]. The OpenFlow standard defines a communication protocol between network switches forming the data plane and one or multiple controllers forming the control plane.

In this work the SDN system setup is built using free open source tools. We use Floodlight v1.0 [12] as the SDN controller and Mininet 2.2.0 [9] for the SDN switches. Floodlight is an Apache-licensed, Java-based OpenFlow SDN Controller. Mininet can create a realistic virtual network. The SDN controller can communicate with the switches via the OpenFlow protocol through the abstraction layer present at the forwarding hardware.

The architecture of an SDN network is illustrated in Fig. 3 and is comprised of Floodlight controller and Mininet switches. An OpenFlow controller typically manages a number of switches, and every switch maintains one or more flow tables that determine how packets belonging to a flow will be processed and forwarded. Communication between a controller and a switch happens via the OpenFlow protocol, which defines a set of messages that can be exchanged between these entities over a secure channel. The state monitor module can be used to collect switch state and transmit it to the controller.

3.2 Constrained Shortest Path Problem and LARAC Algorithm

Given a network $G(V, E)$, assume every link $L_{u,v} \in E$ has two weights $c_{uv} > 0$ and $d_{uv} > 0$ (denoting, cost and delay). For source and destination nodes (s, t) and maximum delay $T_{max} > 0$, let $\mathbf{P}_{st}$ denote the set of paths from s to t. Further, for any path p define

$$c(p) = \sum_{(u,v)\in p} c_{uv},\ d(p) = \sum_{(u,v)\in p} d_{uv}. \tag{8}$$

CSP problem seeks to arrive at the shortest path between s and t nodes with a certain link cost c. However, when the path is constrained by more than one constraint, the problem is termed an MCP problem. Given that there are multiple paths between s and t, a modified MCP problem, often called the multiconstrained optimal path (MCOP) problem, is defined where the goal is to retrieve the shortest path among a set of feasible paths.

A feasible path $s \rightarrow t$ is defined as path p_{st} that satisfies $d(p_{st}) \leq T_{max}$; let $P_{st}(T_{max})$ be the set of all feasible paths from s to t. Then, the CSP problem can be formulated as an integer linear program (ILP) with a set of zero-one decision variables [4,6,16]. The CSP NP-hard problem have many algorithmic approaches that successfully arrived at feasible solutions. The Lagrangian Relaxation Based Aggregated Cost (LARAC) algorithm developed in [4] solves the integer relaxation of the CSP problem efficiently.

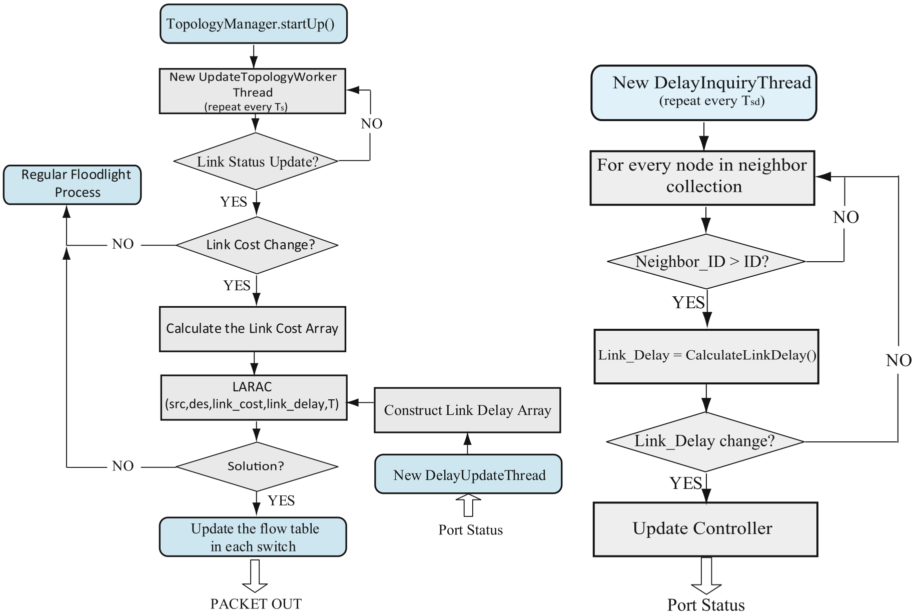
3.3 Vulnerable-Link Adaptive Avoidance (VLAA) via SDN

We adopt the CSP formulation to capture the problem of best-effort avoiding vulnerable links while maintaining a QoS constraint (specifically, a delay constraint). We propose a Vulnerable-Link Adaptive Avoidance (VLAA) algorithm that uses previously-defined vulnerability metric in addition to communication delay in order to arrive at a set of feasible paths between source node s and destination node t. Link delays are observed through SDN switches at each update interval, and if changes are observed, the OpenFlow Floodlight controller is updated. Similarly, link vulnerability costs are observed and the controller cost matrix is updated when changes are sensed.

The VLAA algorithm is implemented in two parts; a controller function which is implemented in Floodlight using Java, and a switch function implemented in Mininet using Python. The flowchart of the VLAA algorithm that is implemented in the controller side is shown in Fig. 4(a). Here, the algorithms perform the following main tasks [1]:

- Listening to messages from switches and calculating link delay value of each link, and then constructing the link-delay cost matrix.
- Calculating the link-vulnerability cost matrix according to the proposed metric; this matrix can be modified and calibrated by network engineers.
- Running a topology-update thread, and checking the link cost matrix updates regularly; if a change is detected, the controller recalculates the routing paths.
- Calculating the routing paths based on the link cost metrics of interest, and updating the flow table of each switch by advertise a PACKET OUT message to switches.

The main function of the VLAA algorithm in the switches' side is to collect the values of link delays for the directly-connected switches as shown in Fig. 4(b). This is achieved by periodically testing the link between that switch and all connected switches with higher ID. Link delay testing is done periodically and the average value is then compared with the last known value. If the new delay is


TopologyManager.startUp()
New UpdateTopologyWorker Thread (repeat every T_s)
NO
Link Status Update?
YES
Regular Floodlight Process
NO
Link Cost Change?
YES
Calculate the Link Cost Array
LARAC (src,des,link_cost,link_delay,T)
Construct Link Delay Array
NO
Solution?
YES
New DelayUpdateThread
Port Status
Update the flow table in each switch
PACKET OUT
New DelayInquiryThread (repeat every T_{sd})
For every node in neighbor collection
NO
Neighbor_ID > ID?
YES
Link_Delay = CalculateLinkDelay()
NO
Link_Delay change?
YES
Update Controller
Port Status

(a) in the SDN controller

(b) in the switches

Fig. 4. Flowchart of the VLAA algorithm

significantly different from the previous value, the switch updates the controller accordingly.

The mechanism in which an SDN switch exchanges link-delay updates with the Floodlight controller is implemented using Port Status messages. OpenFlow standards (v1.0–v1.4) expect the switch to send Port Status messages to the controller as port configuration state changes. These events include change in port status (for example, if it was brought down directly by a user) or a change in port status as specified by 802.1D standard.

4 Conclusions

In this work, we proposed a communication link vulnerability metric that is suitable for cyber security study and mitigation. The proposed metric relies on different parametric and dynamic characteristics of the communication network, and is most useful in adaptive communication networks such as SDNs. Future work would evaluate the proposed metric performance for different attack and threat models within the mitigation framework via QoS routing implementation in SDN.

References

1. Hammad, E., Zhao, J., Farraj, A., Kundur, D.: Mitigating link insecurities in smart grids via QoS multi-constraint routing. In: IEEE ICC Workshops: Workshop on Integrating Communications, Control, and Computing Technologies for Smart Grid (ICT4SG) (2016)
2. Hohlbaum, F., Braendle, M., Alvarez, F.: Cyber security practical considerations for implementing iec 62351, ABB Technical Report (2010)
3. International Federation of Red Cross and Red Crescent Societies: What is vulnerability (2016). http://www.ifrc.org/en/what-we-do/disaster-management/about-disasters/what-is-a-disaster/what-is-vulnerability/. Accessed 14 June 2016
4. Jüttner, A., Szviatovski, B., Mécs, I., Rajkó, Z.: Lagrange relaxation based method for the QoS routing problem. In: Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, pp. 859–868 (2001)
5. Ku, I., Lu, Y., Gerla, M., Gomes, R.L., Ongaro, F., Cerqueira, E.: Towards software-defined vanet: architecture and services. In: 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), pp. 103–110. IEEE (2014)
6. Kuipers, F., Van Mieghem, P., Korkmaz, T., Krunz, M.: An overview of constraint-based path selection algorithms for QoS routing. IEEE Commun. Mag. **40**(12), 50–55 (2002)
7. Liotta, A., Geelen, D., van Kempen, G., van Hoogstraten, F.: A survey on networks for smart-metering systems. Int. J. Pervasive Comput. Commun. **8**(1), 23–52 (2012)
8. Mendonca, M., Obraczka, K., Turletti, T.: The case for software-defined networking in heterogeneous networked environments. In: Proceedings of the 2012 ACM Conference on CoNEXT Student Workshop, pp. 59–60. ACM (2012)
9. Mininet: Mininet (2015). http://mininet.org/. Accessed 9 June 2015
10. Nunes, B., Mendonca, M., Nguyen, X.-N., Obraczka, K., Turletti, T.: A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun. Surv. Tutorials **16**(3), 1617–1634 (2014)
11. Open Networking Foundation: Software-defined networking: the new norm for networks. ONF White Paper (2012)
12. Project Floodlight: Project Floodlight (2015). http://www.projectfloodlight.org/floodlight/. Accessed 9 June 2015
13. Stamp, J., Campbell, P., DePoy, J., Dillinger, J., Young, W.: Sustainable Security for Infrastructure Scada. Sandia National Laboratories, Albuquerque (2003). www.sandia.gov/scada/documents/SustainableSecurity.pdf
14. Wang, A.J.A.: Information security models and metrics. In: Proceedings of the 43rd Annual Southeast Regional Conference, vol. 2, pp. 178–184. ACM (2005)
15. White, G.B.: The community cyber security maturity model. In: 2011 IEEE International Conference on Technologies for Homeland Security (HST), pp. 173–178. IEEE (2011)
16. Xiao, Y., Thulasiraman, K., Xue, G., Jüttner, A.: The constrained shortest path problem: algorithmic approaches and an algebraic study with generalization. AKCE Int. J. Graphs Comb. **2**(2), 63–86 (2005)
17. Zhao, J., Hammad, E., Farraj, A., Kundur, D.: Network-aware QoS routing for smart grids using software defined networks. In: Leon-Garcia, A., et al. (eds.) Smart City 360, pp. 384–394. Springer, Heidelberg (2016)