Vulnerability Analysis in Multicommodity Stochastic Networks by Game Theory

Piyanan Satayapiwat, Kalika Suksomboon and Chaodit Aswakul

Department of Electrical Engineering, Faculty of Engineering,
Chulalongkorn University, Payathai Rd., Pathumwan, Bangkok 10330, Thailand
piyanan.satayapiwat@gmail.com, kmitmink@yahoo.com and chaodit.a@chula.ac.th

***Abstract*— In this paper, by applying the game theoretical framework, we propose a new vulnerability identification method in the multicommodity stochastic network. A new performance indicator–expected achievable capacity (EAC)–is proposed to quantify the vulnerability level of network links when the network is attacked by an intelligent adversary. To compute for EAC, a maximin problem is formulated and solved by the method of successive average and linear programming. Reported numerical results on a grid network topology with multiple OD demands show that the effect of network vulnerability can be well represented by the proposed EAC and hence this suggests the usefulness of the proposed network vulnerability analysis framework.**

I. Introduction

Occurrence of failures within a network can be traced towards many possible causes, i.e., natural disasters, malfunction of network equipment and improper routine maintenance operations. In the conventional analysis of network reliability [1], [2], the most common assumption is that link/node failure events occur either one at a time or in a simplified random manner. This assumption is well justified for failures occurring naturally. However, in recent years with the emergence of terrorist attempts, apart from natural failures, it is equally important that engineers must also be concerned with the new form of network reliability threat from intentional network attacks by hackers or terrorists.

The reliability analysis based on an *intelligent* attacking entity has been investigated by using the framework of game theory. With game players being a dispatcher and a demon, the risk in transporting hazardous materials across a road network can be quantified [3]. The game objective is for the transport company to minimize the risk of exposing hazardous materials upon the road accidents which occur on purpose to maximize that risk. Likewise, when the system is a road network, the game players can be defined as the intelligent drivers that can optimally steer their vehicles to avoid the road congestion that is worsen by an imaginary network tester [4]-[6]. In a mobile ad hoc network (MANET), its reliability of communication has been modelled by a game competed between a router and an imaginary network tester [7]. This work has defined a new cost function to accommodate random link failure costs due to MANET wireless transmission nature. By solving this game, the relationship between mean link failure cost and optimal path selection scenarios can then be investigated.

From these literatures, a game is formulated for two players, namely, a network router and a network attacker. The router has the objective of trying to find the optimal path for transmission of commodities across the network. Conversely, the tester tries to obstruct such transmission in the most disruptive way. The existing literature relies on various definitions of cost function, depending on the system measures. These functions include the network delay or travel time [4]-[7] and the number of eavesdropped or intercepted packets [8].

In this work, the focus is steered towards a new cost function in terms of the total achievable flow capacity between multiple pairs of terminals or nodes. The aim is in finding how much flow at most can be sent across a network. In deed, this is inspired by the fundamental question in the well-known theory of maximum-flow, minimum-cut problem [9]. However, by adopting the game theoretical framework, the solution can be further refined. That is, this work is aimed at finding how much *reliable* flow at most can be sent across a *multicommodity stochastic network* whose components may fail randomly but in the *most disruptive* ways. The solution relies on a newly defined measure, herein called *expected achievable capacity (EAC)*, as to be further defined in Section II. In addition, an algorithm is here proposed to identify the links whose failure would affect the network achievable capacity the most. This algorithm can thus be applied in helping engineers sort out the vulnerability of links so that an efficient link backup plan can be well prepared.

II. Expected Achievable Capacity

A network comprises of a set of nodes and links. Links in the network are indexed by i. Let us assume that a failure scenario j belongs to the failure scenario set of J possible cases. The set of completely disrupted links, given the occurrence of failure scenario j, is represented by Q_j. The functional capacity of link i is C_i, which is reduced to 0 if it fails. The achievable capacity of path k is defined by

$$R_k = \min_{i \in L(k)} C_i \tag{1}$$

Let $C_{i,j}$ denote the achieved capacity from link i under failure scenario j. We then have

$$C_{i,j} = \begin{cases} 0, & i \in Q_j \\ C_i, & \text{otherwise} \end{cases}$$

978-1-4244-2101-5/08/$25.00 ©2008 IEEE

Note here that it is straightforward to extend from this formulation to partial link failure events where not the whole capacity of link is disrupted by its failure by adjusting the conditional values of $C_{i,j}$. A source node tries to send its data traffic towards its destination with the total of K possible paths. The set $L(k)$ of all links along path k is chosen by the source node. Since the achievable capacity equals the capacity of the bottleneck link on that path. The achievable capacity of path k under failure scenario j, $R_{k,j}$, can be obtained from

$$R_{k,j} = \min_{i \in L(k)} C_{i,j} \tag{2}$$

In the multiple Origin-Destination (OD) network, path index k represents a path which must be a member of *only one* specific OD pair. For notational convenience, the payoff table of achievable capacity defined as *achievable capacity matrix* **R** can then be written as

$$\mathbf{R} = \begin{bmatrix} R_{1,1} & \dots & R_{1,J} \\ \vdots & \ddots & \vdots \\ R_{K,1} & \dots & R_{K,J} \end{bmatrix}$$

In the subsequent formulated network game with mixed strategy, the sender selects path k with probability h_k and the attacker forces failure scenario j to occur with probability q_j. The vector form of these strategy selection probabilities are denoted as $\mathbf{H}^T = [h_1, \dots, h_K]$, $\mathbf{Q}^T = [q_1, \dots, q_J]$. It should be noted here that the number of rows in the matrix **R** and the path selection strategies **H** now include paths from all OD pairs.

Finally, we define a new game cost function, *Expected Achievable Capacity* (EAC) as the maximum capacity achieved on average at the interval of data transmission when the worst-case single-link failure occurs. Given **H** and **Q**, EAC can then be calculated from **R** directly:

$$EAC = \sum_{k=1}^{K} \sum_{j=1}^{J} h_k q_j R_{k,j} = \mathbf{H}^T \mathbf{R} \mathbf{Q} \tag{3}$$

III. Game Formulation and Solution Methods

A. Player Strategy and Aim

Network reliability analysis is visualized as a network game between two players, a router and an intelligent network attacker. Both players are assumed to be rational. The router objective is to maximize the achievable capacity by utilizing the stochastic routing technique in which the router selects paths for data transmission optimally for every pair of origin and destination nodes. Conversely, the network attacker objective is to minimize the achievable capacity by choosing to invoke failure scenarios in an optimal and random manner. In this game, the objective cost function is directly computable from the proposed EAC, which is defined in (3). Let the total number of demand pairs M be indexed by m and $Z(k,m)$ be the path-OD index defined by

$$Z(k,m) = \begin{cases} 1, \, k \in P(m) \\ 0, \, \text{otherwise} \end{cases}$$

where $P(m)$ is the set of candidate paths of OD pair m. Let $\delta(k,i)$ be the path-link index defined by

$$\delta(k,i) = \begin{cases} 1, \, i \in L(k) \\ 0, \, \text{otherwise} \end{cases}$$

In the network game formulation, the router seeks the best path selection strategy **H** by solving

$$\max_{\mathbf{H}} \min_{\mathbf{Q}} \mathbf{H}^T \mathbf{R} \mathbf{Q} \tag{4}$$

and the attacker seeks the worst-case failure scenario **Q** by solving

$$\min_{\mathbf{Q}} \max_{\mathbf{H}} \mathbf{H}^T \mathbf{R} \mathbf{Q} \tag{5}$$

Both (4) and (5) are optimized subject to the following constraints

$$\sum_{k=1}^{K} \delta(k,i) h_k R_k \le C_i; \, \forall i \tag{6}$$

$$\sum_{k=1}^{K} Z(k,m) h_k \le 1; \, \forall m \tag{7}$$

$$\mathbf{H} \ge \mathbf{0} \tag{8}$$

$$\sum_{j=1}^{J} q_j = 1, \mathbf{Q} \ge \mathbf{0} \tag{9}$$

Because multiple demand pairs can select the same or overlapped data transmission path, this can result in an over utilized link. The constraint (6) prevents this issue by ensuring that the average capacity usage for all demand pairs on a link cannot exceed the link capacity. For every link, the constraint (6) must not be violated since an over utilized link may cause an unacceptable delay on that link. Constraint (7) implies that the total summation of path selection probabilities for each demand pair must not be larger than 1. The inequality is used to represent the possibility of the senders that have decided not to transmit any data at all, e.g., to save bandwidths for the other senders with higher efficiency in utilizing network resources. Constraint (8) guarantees non-negative values of the selection probability of router player, whereas (9) guarantees non-negative selection probability values and sets the total summation of selection probability for attacker player to 1.

B. Solving Game by MSA

In practice, the optimization problem formulated in (4) and (5) cannot be solved easily by linear programming. Therefore, this paper proposes an algorithm, modified from the well-known method of successive average or MSA [5], which iteratively updates the selection probability for every possible strategy in each turn. An advantage of MSA is that it can find a solution for the maximin problem even when link costs are traffic dependent [5], [10]. The solution methods by MSA for the formulated game are summarized as follows.

1) Let $\phi(k)$ denote the number of all candidate paths that serve the same OD as path k. For each OD, equalize the router's probability of selecting all candidate paths

by $h_k = 1/\phi(k)$. Also, equalize the attacker's probability of selecting to invoke all failure scenarios by $q_j = 1/J$

2) Set the iteration index $n = 1$.
3) Given the failure scenario $\mathbf{Q}$, the router calculates the optimal path selection probability $\mathbf{H}^* = [h_1^*, \ldots, h_K^*]^T$ by linear programming

$$\mathbf{H}^* = \arg\max_{\mathbf{H}} EAC = \arg\max_{\mathbf{H}} \mathbf{H^T RQ}$$

subject to constraints (6)-(8).

4) Router updates the new path selection probability (h_k) by using MSA:

$$h_k \leftarrow (\frac{1}{n})h_k^* + (\frac{n-1}{n})h_k.$$

5) With the newly obtained path selection scenario $\mathbf{H}$, the attacker calculates the optimal failure selection probability $\mathbf{Q}^* = [q_1^*, \ldots, q_J^*]^T$, by linear programming

$$\mathbf{Q}^* = \arg\min_{\mathbf{Q}} EAC = \arg\min_{\mathbf{Q}} \mathbf{H^T RQ}$$

subject to constraint (9).

6) Attacker updates failure selection probability (q_j) by using MSA:

$$q_j \leftarrow (\frac{1}{n})q_j^* + (\frac{n-1}{n})q_j.$$

7) Evaluate EAC from the game at iteration n

$$EAC = \sum_{k=1}^{K}\sum_{j=1}^{J} h_k q_j R_{k,j}.$$

8) If the difference between EAC computed from router in (4) and attacker in (5) is larger than a tolerable threshold, then update $n \leftarrow n+1$ and go back to step 3. Otherwise, stop this recursion.

IV. Vulnerability Identification Method

This paper proposes to use the EAC value to help identifying the network component vulnerability. The main concept of the proposed method is to quantify the effect of link capacity reduction on EAC. Hence, the proposed vulnerability identification begins with the removal of a certain amount of capacity from each link. Then, by using MSA, the remaining network with reduced capacity is analyzed for the EAC value of game. This value represents the obtainable maximum flow that can still pass through the remaining network on average when the worst-case link-failure scenario occurs. Let α denote the amount of capacity reduced when a capacity degradation event occurs on link i. Also, let $EAC_i(\alpha)$ be the obtained EAC when link i is degraded by α capacity units. Obviously, the more the network has EAC, the more resilience/reliability the network can be. In this regard, a link is said to be vulnerable if it causes the reduction of EAC once it is degraded. Therefore, the most vulnerable link $\hat{i}$ is defined by the link which gives the lowest EAC from

$$\hat{i} = \arg\min_{i} EAC_i(min(\alpha, C_i)) \qquad (10)$$

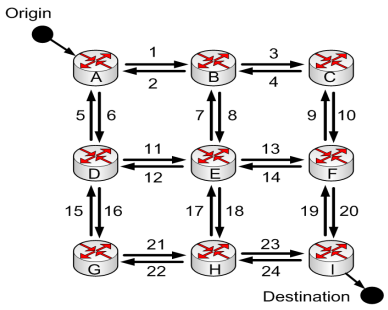

Fig. 1. Grid network with five demand pairs for router player. Each link in this network has 200 units.

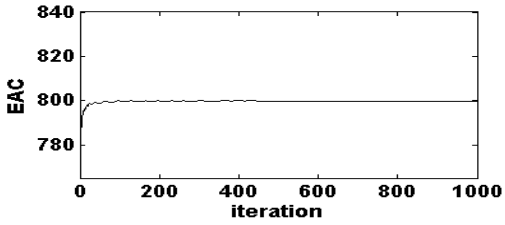

Fig. 2. Convergence of EAC of grid network.

V. Numerical Results

A. Network Vulnerability Identification

The identification of link vulnerability to the overall EAC is investigated in this part. Fig. 1 represents a grid network with each link capacity of 200 units. For each network state, the EAC value can be obtained from the converged value of EAC by the proposed multi-OD game model. The EAC value converges to its stable state when the difference between router objective function (4) and attacker objective function (5) is smaller than the threshold value (see Fig. 2). This represents the Nash equilibrium point where both players cannot gain any advantage over the other by changing his/her own strategy.

By using the proposed vulnerability identification method, the effect of link capacity degradation on EAC can then be shown (see Fig. 3). The most vulnerable link can be

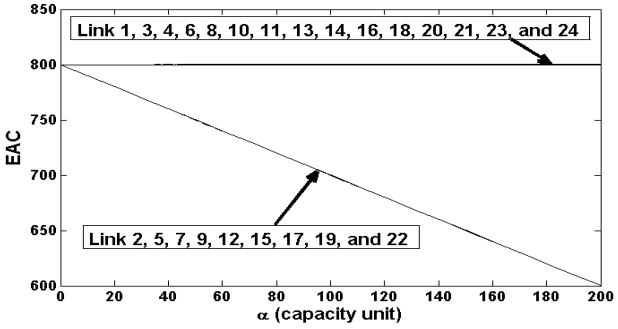

Fig. 3. Effect of link capacity reduction on EAC for a grid network. The test uses 5 different OD pairs with 6 possible shortest paths for each OD using hop count as a metric. These OD pairs consist of demands from A to I, C to G, H to A, I to A, and G to C.

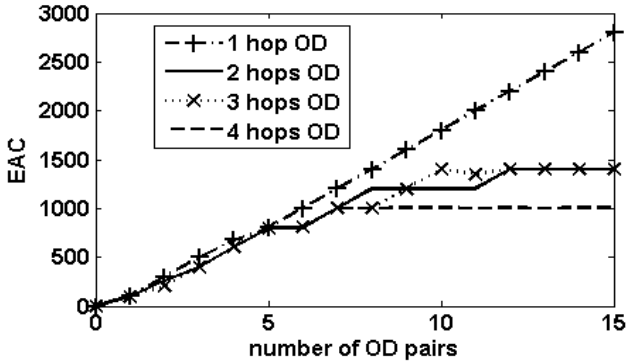

Fig. 4. EAC obtained when adding different types of OD demand pairs.

identified using (10). If the target is on the overall system performance when a link is *completely* failed, then the link whose complete failure gives the lowest EAC must be protected first, i.e. links 2, 5, 7, 9, 12, 15, 17, 19, and 22. When a *partial* link failure is a major concern, the set of links that must be protected first can be identified by (10) depending on the level of failure that engineers want to prevent. If we want to have a network with high fault tolerance while this network must be able to withstand the partial link failure event of 160 capacity units ($\alpha = 160$). The links that must be upgraded first are link 2, 5, 7, 9, 12, 15, 17, 19, and 22.

B. Complexity of the Proposed Method

The complexity of the proposed method relies on the number of decision variables in linear programming function calculated by the router and attacker. To show the complexity of the proposed method, the worst case of the simplex algorithm [11] is used. It is known that the worst case time complexity of the simplex algorithm grows exponentially with the number of decision variables [12]. Thus, the computational complexity of linear programming of router is $O(2^K)$. Likewise, the computational complexity of linear programming of attacker is $O(2^J)$. In each game iteration, the computational complexity of both players is $O(2^n)$ where $n = max(K, J)$. However, the computational complexity here is of the worst case of simplex method. In practice, other algorithms with lower complexity may be used to calculate the EAC in linear programming function. To compute the EAC in Fig. 1, a computer with Intel(R) Core(TM)2 CPU 1.83 Ghz and 1024 MB of RAM is used. The calculation time of EAC for 1,000 iterations as shown in Fig. 2 is 24.4 seconds, which may be considered acceptable for the calculation of general network topologies.

C. Effect of Demand Distance on EAC

Apart from upgrading vulnerable links to increase EAC, the distance between demand pair also affects the amount of the total EAC obtained from the game. To study this effect using the network in Fig. 1, four different types of OD pairs are used, i.e., demand pairs with shortest distance between OD of 1, 2, 3, and 4 hops. Fig. 4 shows the effect of increasing the demand pairs of the same type. By randomly places demand pairs into the network, the EAC increases along with the number of demand pairs until the network is saturated. The result indicates that the increment of EAC in the network with short-distance OD pairs is higher than the network with long-distance demand pairs. This is because of the high capacity usage of the long-distance OD pairs along multiple hops which causes the network to saturate rapidly.

The amount of EAC obtained from one network depends on many factor, e.g., distance between demand pair, number of candidates paths, number of demand pairs, link capacity, and network topology. Finding an efficient method to upgrade a network with high fault tolerance based on the worst-case of failure requires a careful consideration, and is one of the interesting area for future research work.

VI. Conclusion

The contribution of this work is twofold. Firstly, based on multi-commodity network model, we propose the EAC as a new network reliability indicator. Secondly, we propose a new method to identify link vulnerability of multicommodity stochastic network from the EAC. The work scope is extended to cover a general aspect of multicommodity network where multiple user demand pairs co-exist and share limited network resources. Furthermore, network vulnerability identification problem is now solved by the proposed vulnerability identification method which can sort out the vulnerable links in both aspects of failure (i.e. complete or partial link failure).

References

[1] A. Chen, H. Yang, H.K. Lo, and W. Tang, "A capacity related reliability for transportation network" *Journal of Advanced Transportation*, vol. 33, no. 2, pp. 183-200, 1999.

[2] H.K. Lo, and Y.K. Tung, "Network with degradable links: capacity analysis and design" *Transportation Research Part B: Methodological*, vol. 37, no. 4, pp. 345-363, 2003.

[3] M.G.H. Bell, "Mixed Route Strategies for the Risk-Averse Shipment of Hazardous Materials," *Netw. and Spat. Econ.*, vol. 6, no. 3, pp. 253-265, 2006.

[4] M.G.H. Bell "The measurement of reliability in stochastic transport networks," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oakland, 2001, pp. 1183-1188.

[5] M.G.H. Bell, "The use of game theory to measure the vulnerability of stochastic networks," *IEEE Trans. Reliab.*, Vol. 52, no. 1, pp. 63-68, 2003.

[6] M.G.H. Bell, "A game theory approach to measuring the performance reliability of transport networks," *Transportation Research B*, vol. 34, no. 6, pp. 533-545, 2000.

[7] H. Karaa, and J.Y. Lau, "Game Theory Applications in Network Reliability," in *Proc. Communications, 23rd Biennial Symposium*, 2006, pp. 236-239.

[8] S. Bohacek, J.P. Hespanha, J. Lee, C. Lim, and K. Obraczka, "Game Theoretic Stochastic Routing for Fault Tolerance and Security in Computer Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 9, pp. 1227-1240, 2007.

[9] L.R. Ford, and D.R. Fulkerson, *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[10] Z.C. Li, and H.J. Huang, "Fixed-Point Model and Schedule Reliability of Morning Commuting in Stochastic and Time-Dependent Transport Networks," in *LNCS*, vol. 3828, pp. 777-787, 2005.

[11] G.B. Dantzig, *Linear Programming and Extensions*. Princeton University Press, Princeton, NG, 1963.

[12] V. Klee, and G. J. Minty, "How good is the simplex algorithm?," in *(O. Shisha, ed.) Inequalities III, New York, Academic Press*, 1972, pp. 159175.