Network Topology Effects on the Detectability of Crossfire Attacks

Christos Liaskos, Sotiris Ioannidis
Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH),
N. Plastira 100, ZIP 70013, Heraklion, Crete, Greece.
Emails: {cliaskos, sotiris}@ics.forth.gr.

***Abstract*—New strains of Distributed Denial-of-Service (DDoS) attacks have exhibited potential to disconnect communication networks, even cutting off entire countries from the Internet. The "Crossfire" is a new, indirect DDoS link-flooding attack, which masks itself as natural congestion, making it very hard to counter. Several studies have proposed online attack detection schemes, whose efficiency has been shown to vary in different network topologies. However, the topology/detection relation has been studied qualitatively, without formal proof or quantification metric. The present study is motivated by the fact that network topology changes are generally expensive and slow. Therefore, network designers should be provided with means of evaluating the effects of topology modifications to the attack detection efficiency. The study fills this gap by contributing a formal proof for the topology-detection efficiency relation, as well as a novel offline metric that quantifies it. Full attack prototypes are implemented and evaluated in real Internet topologies, validating the analytical findings. It is shown that the novel metric expresses the topology-detection relation efficiently, while existing and widely-used metrics do not constitute good choices for this task.**

***Index Terms*—Network Security; DDoS; Link-flood; Crossfire; topology.**

I. Introduction

Distributed link flooding attacks are a constant threat against the connectivity of networks. Such attacks have been observed in the wild, taking down web services and service providers [1], [2]. The scale of the exerted attack traffic in these cases has reached up to $Tbps$ levels, which shows the critical importance of the threat [3]. Moreover, DDoS link-flooding attacks are evolving, gaining in stealth and execution ease [4], making their mitigation extremely hard [5]–[7]. New defense heuristics are being researched, aiming to detect the attack with minimal interference to the network's operation.

Novel, sophisticated link-flooding attacks seek to affect multiple network parts, in order to obfuscate the existence and identity of a specific target. Moreover, they employ seemingly legitimate flows in the process. The *Coremelt* and *Crossfire* attacks in particular [5], [6]: *(i)* use non-spoofed source IP addresses to send traffic, which are much harder to filter, *(ii)* send packets to publicly accessible decoy servers to flood indirectly a seemingly non-targeted node, and *(iii)* transmit legitimate low-bandwidth flows [8] (e.g., normal HTTP messages). These "under-the-radar" flows then cumulatively flood certain target links. Thus, the Crossfire class constitutes a generalization over past link-flooding attacks, given that it can attack a target indirectly, whereas past attacks where direct and, hence, easier to detect [9].

If an attack cannot be detected, the network operator will at least treat it as a congestion event, i.e., perform traffic engineering (TE) to alleviate the congested part of the network [10]. However, if an operator can detect the attack, he can take immediate reactions to mitigate it (reactive measures), such as botnet exposing and blacklisting [11]. Additionally, he can take precautionary measures against future attacks (proactive measures), such as collaborating with neighboring ISPs for extra connectivity bandwidth, making link-flooding harder. Thus, DDoS detection heuristics run in tandem with TE to detect the attack target and the involved bots [12]–[16]. The network and the attacker engage in a loop where: *(i)* links get flooded by attack traffic, *(ii)* heuristics attempt to perform bot detection and blacklisting, and *(iii)* TE is performed to relieve the flooded links and restore connectivity [17].

Related studies have noted that topological factors can facilitate the *execution* of a Crossfire attack [18]. A topology is commonly defined as a graph of nodes and links, annotated with: *(i)* the routing rules at the router-nodes, *(ii)* the network traffic matrix (i.e., the traffic flows between all node pairs), and *(iii)* the capacity of each link. Assuming these inputs, related studies search for vulnerable links on the basis of serving too many paths and being naturally and persistently overloaded [18]. Subsequently, proactive measures for fortifying a topology can be taken. Nonetheless, the relation between a topology and the *detection* of Crossfire attacks has not been formally studied, despite qualitative evidence denoting high significance [16], [18]. Therefore, network operators have no means to quantify how a generally expensive and slow topological change (e.g., deploying a new fiber)

will affect the detection efficiency of future attacks.

The present study addresses these concerns by contributing:

- An analytical proof of the relation between the network topology and the Crossfire attack detection efficiency. Practical insights on the form of the relation are derived via case-studies.
- A novel, analysis-derived metric, which receives a graph as its input, and quantifies the attack detection efficiency within it as scalar value bounded in $[0, 1]$. A practical algorithm is proposed for calculating the novel metric.
- A comparative evaluation of several commonly-used topological metrics, using realistic simulations. The novel metric is shown to accurately quantify the detection efficiency within a graph. Existing metrics are shown not to be good choices for this task in general, while only one (the heterogeneity metric [19]) can serve simply as a non-quantifiable indicator.

The proposed metric can find use in the field of network topology evolution [20]. For instance, general-purpose optimization heuristics, such as genetic algorithms, can employ the novel metric as their optimization driver, and propose topological changes in favor of fast attack detection.

The remainder of this paper is organized as follows. Section II provides the necessary prerequisites on the Crossfire attack and the considered detection process. The novel metric and the analysis of its properties follows in Section III. The corresponding algorithmic formulation of the measurement process is presented in Section IV. The evaluation follows in Section V, and the study of related works in Section VI. Section VII concludes the paper.

II. Prerequisites

The study assumes a destination-based routed network and the Crossfire attack model [5], [6], [21]. Network nodes can freely represent single physical machines or sub-networks. A concise description of the attack is given below.

A set of bots, which are entities with unique identifiers (e.g., machines with different IPs) reside within the studied network or beyond it (i.e., beyond any gateways). An attacker coordinates these bots to cut-off a targeted node from the network, essentially isolating it. The attack model defines a continuous cycle of interactions between the *attacker* (bot swarm) and the *defender* (network administrator):

Attacker Model: The attacker seeks to cut-off the paths connecting the gateways to the *target*. On these paths there exist public servers, the decoys, and the respective target links, which lead to both the target and the decoy servers. In practice, the attacker first constructs a map of the persistent network links around the target, e.g., via distributed traceroutes [6]. Then, he floods one or more target links by sending traffic only to decoy servers. This way, the flood cannot be directly observed by the target, but it still isolates it from the rest of the network.

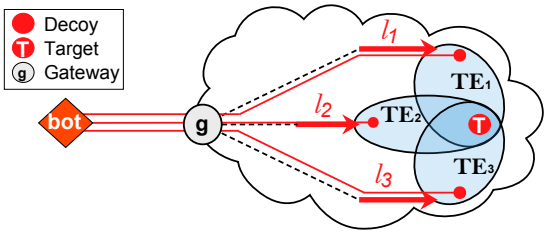


Figure 1: Attacker-Defender interaction. The attacker floods a link l_1. The defender then re-routes traffic (TE_2). The attacker updates the selected decoy servers, flooding link l_2. The defender replies with TE_3 and the attacker floods link l_3, and so on.

Defender Model: The goal of the defender is to keep the network running without any severe performance degradation (e.g., flooded links). Therefore, he first monitors his network and detects overloaded links [12]. Subsequently, attack detection actions are carried out, seeking to detect the existence of a target, the involved bots or both [16]. Finally, he performs TE to balance the incoming traffic, restoring connectivity [22].

Additional measures may include the disabling of traceroute and the deployment of moving target defenses [23]. However, in the cases of Internet providers, traceroute remains vital for monitoring congestion events, hardware failures, peering policy compliance and attack/configuration error detection (e.g., BGP prefix hijacks [24]). Moreover, there also exist public multiple monitoring alternatives that an attacker can use instead (e.g., Periscope and Looking glass [25], RIPE atlas [26]). Finally, moving target defense can be too taxing to apply at Internet provider networks, as it disrupts considerably the legitimate operations as well [23].

Attacker - Defender Interaction: The attacker monitors the network routes and reacts to routing changes performed by the defender, as exemplary shown in Fig. 1. Bots will then change their decoy server selection in case the re-routing has diverted their load from the targeted link(s), repeating the cycle. Thus, the attacker updates the link-map and flow densities, recalculates the target links and floods them again with several bot-originated flows. Notice that each attack step can affect multiple non-targeted nodes, impairing their connectivity as well (Fig. 1). A successful attack should always affect the intended target nonetheless.

Table I: Summary of Notation

Symbol	Explanation
$W=\{\mathcal{N},\mathcal{L}\}$	A graph comprising nodes $\mathcal{N}$ and links $\mathcal{L}$.
$n \in \mathcal{N}$	A single network node.
$l \in \mathcal{L}$	A single, directed network link.
$\overrightarrow{p_{(n,n')}}$	The link-path connecting node n to n'.
$\mathcal{B}(l)$	The available bandwidth of a link l.
$t = 1, 2..$	The time moments when link-floods occur.
$\mathcal{N}^t$	The node-subset affected by the attack at time t.
$\mathcal{N}\{l\}$	The node-subset served by a common link l.
k	The attacker's budget, $k \in [1, \|\mathcal{N}\|]$.
$p_k(W)$	Collateral damage probability in W under k.
k_{max}	The k value for which $p_k = const.\ \forall k \geq k_{max}$.
$\{\pi_m\}$	The profile of a graph W, $m=1...\|\mathcal{N}\|$.
$\|*\|$	The cardinality of a set $*$.

Crossfire-class attacks are inherently undetectable via traffic pattern examination means, since they consider bots that are completely indistinguishable from legitimate users in this aspect [5]–[7]. Therefore, a suitable approach for efficient attack detection is to rely on TE and contrapositive examination of susceptible bots and targets [27]. For instance, flows are rerouted away from flooded links and the network operator monitors the ones that return to cause link floods again [28]. Naturally, flows cannot be considered malevolent for returning to a link-flood event just once. Instead, an iterative approach should be followed, gradually reinforcing the evidence of a flow's malevolence (or innocence). Therefore, reinforcement learning algorithms can constitute promising additions to the Crossfire detection workflow [29]. In related contexts, heuristic-based and distributed multi-agent learning techniques have successfully taken up the tasks of practical evidence reinforcement and bot/target classification decision [15], [30]–[34].

Studies have implemented contrapositive examination using reinforcement learning principles [15], [16]. The workflow is illustrated of Fig. 1. Each attack iteration manages to affect the connectivity of a set of network nodes, which–from the defender's aspect–constitute possible attack targets. When a node is found within a set affected by a link flooding event, its probability of being an attack target is reinforced. The real attack target will lie in the intersection of the affected node sets, yielding the highest reinforcement rate, gradually giving away its existence and identity. The detection framework allows for applying the same logic to the bot detection [16]. In this approach, the probability of an IP address being a bot is reinforced when found to exert traffic over a flooded link. However, as proven in [16], bot detection can be completed evaded when the attacker commands a high amount of bots. In this case, bots can be used very rarely via cycling through the bot set. Thus, bot detection in Crossfire attacks cannot be robust and, subsequently, target detection is prioritized [16].

The employed notation is summarized in Table I. The network graph will be denoted as $W = \{\mathcal{N}, \mathcal{L}\}$, comprising a set of nodes $\mathcal{N}$ and links $\mathcal{L}$. A single node is denoted as $n \in \mathcal{N}$ and a directed network link as $l \in \mathcal{L}$. The path connecting a node n to a destination n' according to these routing rules is $\overrightarrow{p_{(n,n')}} = \{l_1, l_2, l_3, \ldots, l_k\}$, which denotes the sequential path links. The *free* bandwidth of a link is $\mathcal{B}(l)$.

III. Defining the Attack Detectability in a Network

The objective of a Crossfire attack is to isolate a targeted node from the network, while hiding the existence and identity of the target. An attacker consistently tries to confuse the defender by affecting multiple non-targeted nodes, together with the intended one, thereby masking the attack as a natural congestion issue [15], [18]. In other words, noticing the visualization of Fig. 1, the attacker seeks to maintain a large intersection of affected node sets. The posed question is whether the network graph affects this objective.

In order to quantify the Crossfire attack detection efficiency, we first define the helping term of *collateral damage probability* as follows:

Definition 1. The collateral damage probability of a Crossfire attack is the probability of a non-targeted node being affected at a given attack iteration.

Notably, the collateral damage probability depends on the position of the flooded link(s) within the network graph. Attacks near the network leafs affect very few nodes, while attacks near a gateway can affect great parts of the network. In order to generalize Definition 1 to account for this observation, we introduce a tunable factor to the collateral damage and introduce its formal notation as follows:

Let the attacker be able to flood links that carry traffic towards k or less network nodes in a routing tree configuration. With the introduction of the variable k the collateral damage probability within a graph W will be denoted as $p_k(W)$. (For ease of exposition, the variable k will be denoted as *attack budget*, due to its qualitative relation to the attacker's potential. In general, the capacity of a link is related to the number of nodes being serviced through it [35]. Thus, k can be viewed as loosely related to the maximal link capacity that can be flooded by an attack).

The collateral damage probability, $p_k(W)$ is strongly connected to the *detection probability* of Crossfire attacks within a network. In order to understand this claim, consider a network under attack for t iterations. According to Definition 1, at iteration t, any non-targeted node will have suffered $p_k(W) \cdot t$ link-flooding attacks due to wrongful couplings to the real target. Assuming a reinforcement learning approach, this also translates to

$p_k(W) \cdot t$ reinforcements of its probability for being a target. Notably, the probability reinforcement *rate* is equal to $p_k(W)$. Thus, we define the Crossfire detection probability as follows:

Definition 2. The Crossfire attack detection probability within a network is defined as the probability reinforcement *rate* for a targeted node being classified as the target.

Remark 3. The Crossfire detection probability within a graph W is equal to $(1 - p_k(W))$.

The detection probability is related to the practical Crossfire attack detection time yielded by a reinforcement learning-based detection process. In general, such a process receives as input a series of observations and reaches a decision outcome. At each single observation, decision outcomes are reinforced by gradually increasing their likelihood. When the likelihood of a decision outcome reaches a threshold value, the process concludes.

In Crossfire attack detectors [15], [16], the observations are the affected nodes $\mathcal{N}^t$ on every attack incident $t = 0, 1, \ldots$, and the possible decision outcomes are the identity of the node under attack. The attack target likelihood of each node is represented by an integer which is reinforced at each t, counting the occurrences of a node in the affected sets $\mathcal{N}^t$, $\forall t$. Notice that the affected node set will always contain the real target, which will have a count equal to t. The final decision is based on the difference between the counter of the real target and any other node in the affected sets, Δ^t. Therefore, we write:

$$\Delta^t = Count^t_{target} - Count^t_{other} \tag{1}$$

where $Count^t_{target} = t$, while the *average* value of $Count^t_{other}$ is $p_k(W) \cdot t$. Let Δ_c be the confidence needed by the practical detection process, in order to yield a final decision. The detection will then occur *on average* at time moment $t = t_d$ when:

$$\Delta_c = t_d - p_k(W) \cdot t_d \Rightarrow t_d = \frac{\Delta_c}{1 - p_k(W)} \tag{2}$$

Therefore, the higher the detection probability $(1 - p_k(W))$, the speedier the detection of the targeted node.

Having motivated the use of $(1 - p_k(W))$ as the detection efficiency metric, we proceed to formulate its value within a given graph, and derive its analytical properties. For ease of presentation, we will focus on $p_k(W)$ (collateral damage probability) and study full-mesh graphs first. Subsequently, the analysis will be generalized to hold for *any* graph.

A. Full-mesh Network Graphs

In a full-mesh graph, any two nodes are directly connected with a link. Under this circumstance, consider an ordered subset of the graph nodes. Due to the full-mesh connectivity, there always exists a routing configuration that offers a path traversing each node in the ordered set. Moreover, as mentioned in Section II, we consider destination-based routing and, subsequently, there exists one routing path connecting any two nodes. Subsequently, there also exists a link l, denoted as *common service link*, which connects this subset of nodes, $\mathcal{N}\{l\}$, to the rest of the network in a routing configuration. This trait of full-mesh graphs is formalized as follows.

Lemma 4. *In a full-mesh graph, for any node subset $\mathcal{S}$ of the given network $\mathcal{N}$, there exists a common service link l so as $\mathcal{N}\{l\} = \mathcal{S}$.*

(As a side note, in a multi-path routing scenario–which is allowed in source-based routing cases– the current definition of the common service link refers to attacks affecting the connectivity of a target in the general sense. I.e., it examines attacks against at least one of the paths that connect the target, degrading its overall quality of service but not necessarily disconnecting it fully).

Using Lemma 4, we can prove the following Theorem, which gives the collateral damage probability in a full-mesh graph.

Theorem 5. *The p_k metric (collateral damage probability) within a full-mesh graph is:*

$$p_k = \frac{\sum_{m=1}^{k} \binom{\|\mathcal{N}\|-2}{m-2}}{\sum_{m=1}^{k} \binom{\|\mathcal{N}\|-1}{m-1}} \tag{3}$$

Proof: Let $\mathcal{N}^t$ denote the set of all nodes affected the attack at time t. An attacker with budget k is able to flood links with $\|\mathcal{N}^t\| \leq k$, $k = 1 \ldots \|\mathcal{N}\|$. From Lemma 4, this means that there exists a node subset $\mathcal{N}^t \subseteq \mathcal{N}$ with $\|\mathcal{N}^t\| \leq k$.

Let us assume the targeted node n_t at time t. The attack floods the corresponding set of links, affecting the node set $\mathcal{N}^t$, where $\|\mathcal{N}^t\| = m \leq k$. Since the number of non-targeted nodes that are affected by the attack (i.e., nodes in $\mathcal{N}^t$) is $m-1$, the probability that a non-targeted node in the network (i.e., any node out of the $\|\mathcal{N}\| - 1$ non-targeted nodes) is affected by the attack is equal to:

$$q_m = \frac{m-1}{\|\mathcal{N}\| - 1} \tag{4}$$

Additionally, two comments can be made:

1) The number *of* m-sized sets that comprise: i) the targeted node, and ii) $m-1$ other nodes is $\binom{1}{1}\binom{\|\mathcal{N}\|-1}{m-1}$.
2) With budget k, the attacker can select and flood any link serving $m = 1 \ldots k$ nodes (i.e., the targeted one and $m-1$ other nodes) with equal probability. The total number of these sets is $\sum_{m=1}^{k} \binom{1}{1}\binom{\|\mathcal{N}\|-1}{m-1}$.

The probability of the size of the target set $\mathcal{N}^t$ being equal to m is derived from these two remarks. We simply divide the number of m-sized sets (from comment 1) by the number of all sets *up to* size m (from comment 2). Since $\binom{1}{1} = 1$, we obtain:

$$P\left\{\left\|\mathcal{N}^t\right\| = m\right\} = \frac{\binom{\|\mathcal{N}\|-1}{m-1}}{\sum_{m=1}^{k}\binom{\|\mathcal{N}\|-1}{m-1}} \tag{5}$$

Finally, from equations (4) and (5), it follows that the probability that a non-targeted node in the network is affected by the attack is:

$$p_k = \sum_{m=1}^{k} q_m \cdot P\{\left\|\mathcal{N}^t\right\| = m\} = \frac{\sum_{m=1}^{k}\frac{m-1}{\|\mathcal{N}\|-1}\cdot\binom{\|\mathcal{N}\|-1}{m-1}}{\sum_{m=1}^{k}\binom{\|\mathcal{N}\|-1}{m-1}} = \frac{\sum_{m=1}^{k}\binom{\|\mathcal{N}\|-2}{m-2}}{\sum_{m=1}^{k}\binom{\|\mathcal{N}\|-1}{m-1}} \tag{6}$$

■

We proceed to study the properties of p_k versus the attack budget k. In the following Result, we derive a simple and insightful approximate expression for p_k.

Result 6. *The collateral damage probability within a full-mesh graph is approximated as:*

$$p_k = \begin{cases} \frac{k-1}{\|\mathcal{N}\|}, & 1 < k \leq \frac{\|\mathcal{N}\|}{2} \\ \frac{1}{2}, & \frac{\|\mathcal{N}\|}{2} < k \leq \|\mathcal{N}\| \end{cases} \tag{7}$$

Proof: See Appendix. ■

Result 6 becomes exact for large networks ($\|\mathcal{N}\| \gg 1$), however, its accuracy is significant even for small/medium size networks, as shown in the evaluation (Section V).

A number of interesting implications for the mesh graph, for both the attacker and defender, follow from Result 6:

- From the aspect of the attacker, it is shown that there exists a "sweet-spot" in the attack budget selection, which maximizes p_k. Particularly, at $k = \|\mathcal{N}\|/2$, p_k has already reached its maximal value, i.e., $p_{\|\mathcal{N}\|/2} = 1/2$. Therefore, the attacker does not need to increase its attack budget further than this point.
- p_k increases linearly with the attack budget $k \in [1, \|\mathcal{N}\|/2]$, meaning that there are no "sharp" gains for the attacker. On the contrary, the attacker will need to increase his attack budget proportionally to his gains in detection avoidance.
- Moreover, the upper bound of $p_k \leq 1/2$ is also advantageous from the aspect of the defender. It dictates that no matter the attack budget, the target will be observed at least 2 times more than any other node in the $\mathcal{N}_t$ sets. In other words, the target naturally stands-out by a factor of 2 at a minimum.

We proceed to generalize the results of the previous section, where we considered full-mesh graphs, and derive expressions for the metric p_k under any network W.

B. Generic Network Graph

In a graph that is not a full-mesh, Lemma 4 does not hold, since some of the nodes are not connected with direct links. Thus, the TE module has fewer alternative routing trees than in a full-mesh network.

To quantify the number of missing links in a graph, and be able to calculate the probability within it, we define the *detectability profile* of the graph.

Definition 7. The detectability profile of a network graph $W = \{\mathcal{N}, \mathcal{L}\}$, is a vector of scalars such as:

$$\{\pi_m\},\ m = 1 \ldots \|\mathcal{N}\| \tag{8}$$

where $\pi_m \in [0,1]$, $\forall m$ denotes the fraction of subsets $S \subseteq \mathcal{N}$ of size m for which there exists a common service link l so as $N\{l\} = \mathcal{S}$.

Note: The definition of the detectability profile is based on the simple fact that any graph of $\|\mathcal{N}\|$ nodes is a sub-graph of the corresponding full-mesh (whose profile is $\{\pi_m\} = 1,\ \forall m$). The detectability profile terms, π_m, simply measure the percentage of the $\binom{\|\mathcal{N}\|}{m}$ full-mesh subsets S of size m that are also present in the given, specific graph.

Now, given the detectability profile of a graph, we can calculate the $p_k(W)$ within it.

Theorem 8. *The collateral damage probability within a graph W with profile $\{\pi_m\}$, $m = 1 \ldots \|\mathcal{N}\|$, is equal to:*

$$p_k(W) = \frac{\sum_{m=1}^{k}\pi_m \cdot \frac{m-1}{\|\mathcal{N}\|-1}\cdot\binom{\|\mathcal{N}\|-1}{m-1}}{\sum_{m=1}^{k}\pi_m\cdot\binom{\|\mathcal{N}\|-1}{m-1}} = \frac{\sum_{m=1}^{k}\pi_m\binom{\|\mathcal{N}\|-2}{m-2}}{\sum_{m=1}^{k}\pi_m\binom{\|\mathcal{N}\|-1}{m-1}} \tag{9}$$

Proof: Since in a full-mesh graph the number of sets with size equal to m is $\binom{\|\mathcal{N}\|}{m}$ (cf. proof of Result 6), from Def. 7 it follows easily that the number of sets with size equal to m in a graph with profile $\{\pi_m\}$ is:

$$\pi_m \cdot \binom{\|\mathcal{N}\|-1}{m-1} \tag{10}$$

Hence, using the above value and following the same steps as in the proof of Result 6, we derive the expression of equation (9). ■

Using Theorem 8, we can calculate the exact value for the collateral damage probability within any network graph. Moreover, and similar to the full-mesh case, the following Results provide some further insights about the properties of the collateral damage probability within an arbitrary graph.

Result 9. *The collateral damage probability within of any graph:*

1) *Is sub-linearly, non-strictly increasing for* $1 \leq k < \frac{\|\mathcal{N}\|}{2}$,
2) *Remains constant from a* $k_o \in \left[\frac{\|\mathcal{N}\|}{2}, \|\mathcal{N}\|\right]$ *and on,* ($\|\mathcal{N}\| \gg 1$).

Proof: see Appendix ■

Result 10. *The collateral damage probability within a network graph* W *of size* $\|\mathcal{N}\|$ *is upper bounded by the respective full-mesh graph of size* $\|\mathcal{N}\|$*, i.e.,*

$$p_k(W) \leq p_k(mesh) \tag{11}$$

Proof: See Appendix. ■

The above analysis has three interesting implications:

(i) Result 9 shows that the collateral damage probability for any graph follows a similar pattern as in the full-mesh case: it increases sub-linearly with k, it takes its maximum value, and then remains constant. Thus, the existence of a "sweet-spot" for the attacker is proven for any graph.

(ii) Result 10 indicates that among all the physical graphs of equal size $\|\mathcal{N}\|$, the full-mesh is the one offering the highest collateral damage probability. Result 10 refers to the average over all routing trees that can be deployed over a physical graph. In a mesh graph, any routing tree can be deployed, which leads to increased collateral damage probability averaged over all possible routing configurations.

Motivated by the second implication presented above, and in order to give a practical connection between the traits of a graph and its detectability profile, we study an example case. We consider a graph with detectability profile:

$$\{\pi_m\}:\ \pi_m = \begin{cases} 1, & 1 < m \leq k_{max} \\ 0, & k_{max} < m \leq \|\mathcal{N}\| \end{cases}, \tag{12}$$

for some $k_{max} \in \left[1, \frac{\|\mathcal{N}\|}{2}\right]$. As noted, each π_m corresponds to the presence of links that jointly serve m-sized sets of nodes. Therefore, the profile of equation (12) corresponds to a graph without "fat" links, i.e., links serving many nodes. In particular, in the border case of $k_{max} = 2$, the graph becomes a star-layout of nodes. As k_{max} increases beyond 2, the pairwise paths among nodes increase in size accordingly, drifting away from the star layout.

In a graph corresponding to the profile of equation (12), since the first k_{max} terms of $\{\pi_m\}$ are equal to 1, the collateral damage probability coincides with the corresponding full-mesh value, up to $m = k_{max}$. After this point, the graph offers no node subset larger than k_{max} (since the $\{\pi_m\}$ terms for $m > k_{max}$ are equal to zero). Therefore, the collateral damage probability is maximized for $k = k_{max}$ and remains constant for any greater attack budget as follows:

$$p_k = \begin{cases} \frac{k-1}{\|\mathcal{N}\|}, & 1 < k \leq k_{max} \\ \frac{k_{max}-1}{\|\mathcal{N}\|}, & k_{max} < k \leq \|\mathcal{N}\| \end{cases} \tag{13}$$

Notice that the maximal p_k can be much lower than the boundary value, $1/2$, depending on the value of k_{max}. Indeed, for $k_{max} = 2$ and a network of $\|\mathcal{N}\| = 100$ nodes, p_k becomes negligible ($p_k = 1/100$) compared to the upper bound ($1/2$).

(iii) Finally, from equation (9) we notice that the detectability profile, $\{\pi_m\}$, is the only input required to calculate p_k. In other words, a measurement of the detectability profile can explicitly project the behavior of a given graph, for any attack budget k. Thus, in the following Section we proceed to study techniques for measuring and approximating the $\{\pi_m\}$ values of a given graph.

Algorithm 1 Detectability Profile Measurement (DePROM).

INPUTS: The network graph $W(\mathcal{N},\mathcal{L})$; an integer K.
OUTPUT: The detectability profile $\{\pi_m\}$ of W.

checked_pairs$\leftarrow \{\emptyset \rightarrow \emptyset\}$; //Empty Hash
unique_sets$\leftarrow \{\emptyset \rightarrow \emptyset\}$; //Empty Hash
$\pi_m \leftarrow 0,\ m = 1 \ldots \|\mathcal{N}\|$;
while $\exists$ pair $(n,n')\colon n \neq n'$, checked_pairs$[(n,n')]=\emptyset$
  checked_pairs$[(n,n')] = 1$;
  $paths \leftarrow$ KShortestPaths$(n \rightarrow n',\ K)$;
  foreach path $\overrightarrow{p_{(n,n')}}$ **in** $paths$ **do**
    foreach link l **in** $\overrightarrow{p_{(n,n')}}$ **do**
      if unique_sets$[\mathcal{N}\{l\}] = \emptyset$ **then**
        unique_sets$[\mathcal{N}\{l\}] \leftarrow 1$;
        $\pi_{\|\mathcal{N}\{l\}\|} \leftarrow \pi_{\|\mathcal{N}\{l\}\|} + 1$;
      end if
    end foreach
  end foreach
end while
$\pi_m \leftarrow \pi_m / \binom{\|\mathcal{N}\|-1}{m-1}$; //Normalization.

IV. Measuring the Detectability Profile

The detectability profile, $\{\pi_m\}$, is an expression of the paths and links supported by a given graph. A non-mesh graph may not contain links to jointly serve any given node set. This fact may be due to actual lack of physical connections, or due to disallowed connections, such as the ones imposed by security or routing policies [36]. To this end, we propose the Detectability Profile Measurement (DePROM), whose workflow is overviewed in Fig. 2 and formulated as Algorithm 1.

DePROM essentially counts sizes of node subsets in a graph, which can be parts of a connected routing tree.

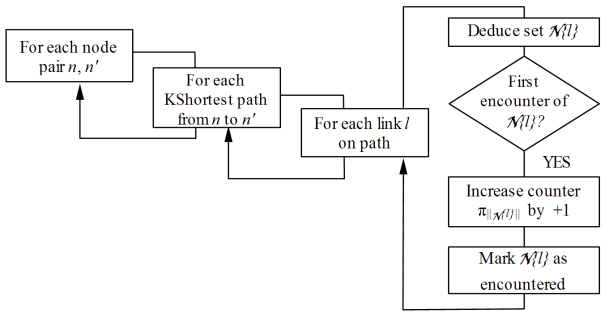


Figure 2: Overview of the DePROM workflow.

Since an exhaustive search of all such subsets can be a combinatorial task, DePROM focuses on practical routing configurations. Thus, as shown in Fig. 2, it calculates the K-Shortest paths (non-disjoint) between all node pairs in the graph. Then, for each path, DePROM then counts all connected node subsets, $\mathcal{N}\{l\}$, over it, for every link l on the path. For example, given a path:

$$\overrightarrow{p_{(1,5)}} : \{l_{1,2},\, l_{2,3},\, l_{3,4},\, l_{4,5}\} \tag{14}$$

we have the node subsets $\mathcal{N}\{l_{1,2}\} = \{2,\, 3,\, 4,\, 5\}$, $\mathcal{N}\{l_{2,3}\} = \{3,\, 4,\, 5\}$, $\mathcal{N}\{l_{3,4}\} = \{4,\, 5\}$ and $\mathcal{N}\{l_{4,5}\} = \{5\}$. The detectability profile, $\{\pi_m\}$, is then derived as an enumeration the encountered, *unique* node subsets over the studied paths. The input parameter K regulates the graph exploration degree by defining the number of paths examined per node pair.

The specific workflow of DePROM is given in the listing of Algorithm 1. At first, two utility hashmaps, `checked_pairs` and `unique_sets`, are defined at lines $1-2$. `checked_pairs` maps node pairs to a Boolean value. Similarly, `unique_sets` maps a node-set to a Boolean value as well. Both variables are initialized as empty maps. In addition, the required $\{\pi_m\}$ values are initialized as an array of zero values at line 3.

Having completed the initialization process, DePROM proceeds to iterate over node pairs (n, n'), $n \neq n'$, performing the following actions. At first, the `checked_pairs` hashmap is updated to denote that the presently examined pair has been iterated over, and needs not be processed further (line 5). Then, the K-shortest paths connecting node n to n' are found and stored in the paths list (line 6). We proceed to examine each link l on the K paths at lines $7-14$, mapping l to its serving node subset, $\mathcal{N}\{l\}$, as described above. If a set $\mathcal{N}\{l\}$ is encountered for the first time, it is marked to avoid future re-processing (lines $9-10$) and the profile π_m receives a unary increase at position $m = \|\mathcal{N}\|$ (line 11). Finally, DePROM normalizes the profile π_m at line 16, following the form of the denominator of relation (9).

Computational Complexity. DePROM is intended to be an offline algorithm. Its complexity is defined by the employed K-ShortestPath implementation. The approach of Eppstein et al yields a complexity of: $O\left(\|\mathcal{L}\| + \|\mathcal{N}\|\, log\, \|\mathcal{N}\| + K \cdot \|\mathcal{N}\|\right)$ per run [37]. Each run produces all K-shortest paths from a given node to all other network nodes. Thus, DePROM needs to execute this procedure $\|\mathcal{N}\|$ times, in parallel over the number of independent computing cores C. Apart from the K-shortest paths sub-process, the DePROM loops of lines $7-14$ require a maximum of $O\left(K \cdot \|\mathcal{L}\|\, \|\mathcal{N}\|\right)$ computations (i.e., K paths per node, each path occupying a maximum of $\|\mathcal{L}\|$ links). Notice that hashmaps have $O(1)$ lookup time. Thus, the total computational complexity of DePROM is $O\left(\left(K + \frac{1}{C}\right) \cdot \|\mathcal{L}\|\, \|\mathcal{N}\| + \frac{\|\mathcal{N}\|^2 (log \|\mathcal{N}\| + K)}{C}\right)$.

DePROM can trade precision for smaller complexity by using low K values as input parameters. Additionally, we note that the formulation of DePROM assumes that an attack can seek to disconnect any two nodes within the network. However, a network designer may be more interested in evaluating the scenario of disconnecting any node from a given node, such as a gateway. In this case, the pair selection of line 4 can focus on the gateway-to-node pairs only, which yields a total complexity of $O\left(\|\mathcal{L}\|\, (K \cdot \|\mathcal{N}\| + 1) + \|\mathcal{N}\|\, (log\, \|\mathcal{N}\| + K)\right)$.

Finally, it is noted that common offline graph metrics, like betweenness centrality[1], have quadratic complexity [38]. Minimum cuts—another widely used concept in computer networks—are NP-Hard to calculate [39]. In that sense, DePROM does not yield a complexity beyond the norm. Moreover, a desirable feature for offline metrics, such as betweenness centrality and DePROM, is the degree of parallelism. Both metrics derive their complexity from the calculation of paths between all node pairs on the network. In principle, each node-pair can be assigned for calculation to a different processor. Therefore, both metrics may be parallelized to reduce their actual runtime by a factor of $\|\mathcal{N}\|/2$.

V. Experimental Evaluation

We proceed to validate the analytical findings and compare DePROM to related studies via numerical and simulations-derived results. The experiments detailed below were executed in 100 real Internet topologies from the TopologyZoo database [40]. This database contains graphs for multiple Internet Service Providers across the globe. In the selected dataset, the graphs comprise 5 to 160 nodes (i.e., not necessarily $\|\mathcal{N}\| \gg 1$). Preliminary executions of DePROM in all examined graphs showed the value of p_k remained unaltered for $K \geq 30$, implying that a full graph exploration is attained at this value. Thus, K is set to 30 for all runs. The related source-code is freely available.

[1]Betweenness centrality calculates the shortest path for each node pair and counts how many times a link is used over all paths.

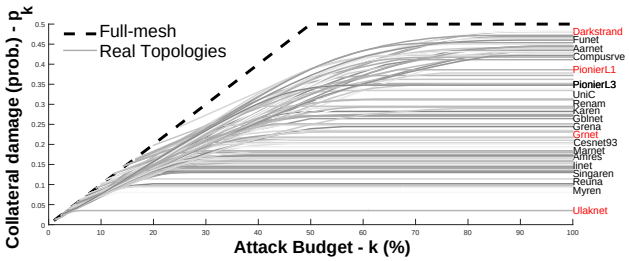
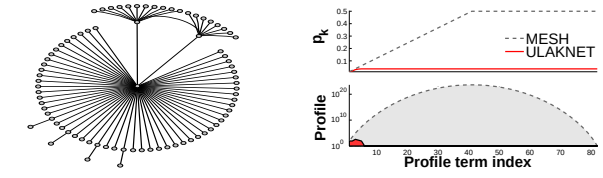
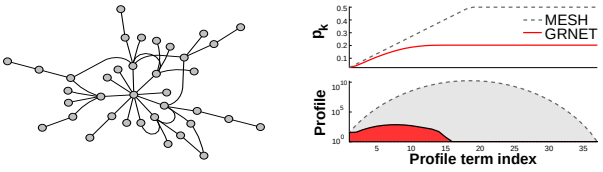


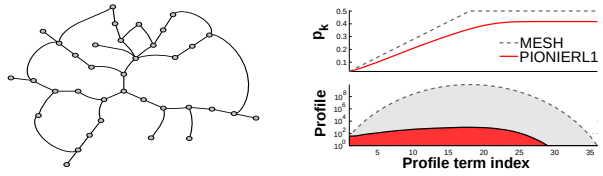
Figure 3: The collateral damage probability (Definition 1) within multiple real Internet topologies (graphs), derived by measuring their detectability profiles with DePROM.



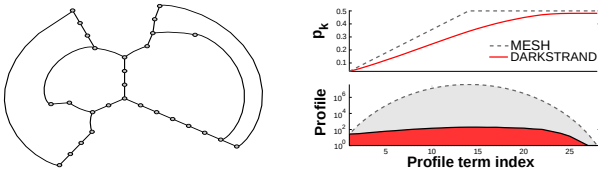
(a) Ulaknet - Turkey ($p_{\|\mathcal{N}\|} \approx 0$).



(b) Grnet - Greece ($p_{\|\mathcal{N}\|} \approx 0.2$).



(c) Pionier L1 - Poland ($p_{\|\mathcal{N}\|} \approx 0.4$).



(d) Darkstrand - USA ($p_{\|\mathcal{N}\|} \approx 0.5$).

Figure 4: Connection between real Internet topologies (graphs) and detectability profiles.

The related metrics include the link-rank exponent [18], the average shortest path length [16], as well as several commonly use topological metrics (heterogeneity, centrality, average neighbor count, network density, clustering coefficient, betweenness centrality exponent, and average connected node pair count [41]). The link-rank exponent and average shortest path length metrics have been theorized to exhibit a qualitative relation to the efficiency of Crossfire attacks (higher values favor the attacker), without formal proof [16], [18]. The remaining metrics are studied for exploratory reasons. The link-rank exponent is the only non-standard metric and is calculated as follows [18]. First we calculate the shortest paths between all node pairs in the topology. Then, we count how many times a link is found in this set of paths. The ensuing distribution (link-rank) is sorted by descending value and fitted to the Zipf-Mandelbrot distribution, i.e., to the formula $f(x) \sim (x+b)^{-a}$. The a value deduced by the curve fitting process is the link-rank exponent. The curve fitting process is identical in the case of the betweenness centrality metric as well.

A. *Numerical Results*

First, we validate the analytical findings regarding the form and properties of p_k in Fig. 3. DePROM is executed for each graph and for $1 \leq k \leq \|\mathcal{N}\|$. Subsequently, given the DePROM-derived graph profiles π_m, the p_k value of each graph is derived via Theorem 8 and plotted it in Figure 3. The x-axis is normalized in $[1, 100]\,\%$ of $\|\mathcal{N}\|$, and is therefore common for all graphs. (Overlapping graph names are omitted).

The dashed-line plot in Fig. 3 corresponds to the theoretical result of Result 6, which is shown to constitute an upper bound for all studied graphs. This outcome is in accordance with Result 10.

The collateral damage probability within each specific graph is shown to follow the general form projected by Result 9. Specifically, each curve follows a sub-linear increase for up to an attack budget value, while, beyond this point, each plot remains constant. Notably, this outcome also demonstrates the existence of a "sweet-spot" in the attacker's budget in the studied graphs. It is also worth noting that Fig. 3 contains plots for graphs with even $\|\mathcal{N}\| = 5$ nodes (e.g., "Renam" case). Regardless of this rather low number of nodes, the form of these plots is accurately predicted by Result 9 as well.

Second, we study the connection between a graph and its detectability profile π_m. Given that the collateral damage probability is bounded within the range $[0, 1/2]$, we select four graphs whose maximal p_k values (i.e., for $k = \|\mathcal{N}\|$) span the complete range. Thus, Fig. 4 presents the profile and form of the "Ulaknet" (Turkey), "Grnet" (Greece), "PionierL1" (Poland) and "Darkstrand" (USA) cases, whose $p_{\|\mathcal{N}\|}$ values are approximately 0.02, 0.2, 0.4 and 0.5. The left part of Fig. 4 shows the real graph, while the right part presents the detectability profile and p_k metrics.

The "Ulaknet" case exhibits a minimal detectability profile. Almost all of the profile terms are absent (i.e., $\pi_m = 0$ for most m), meaning that the graph does not contain links that jointly serve more than a trivial percentage of the nodes. The absence of such links can be validated by observing Fig. 4a, which resembles a star layout. Therefore, pairwise node paths have a hop

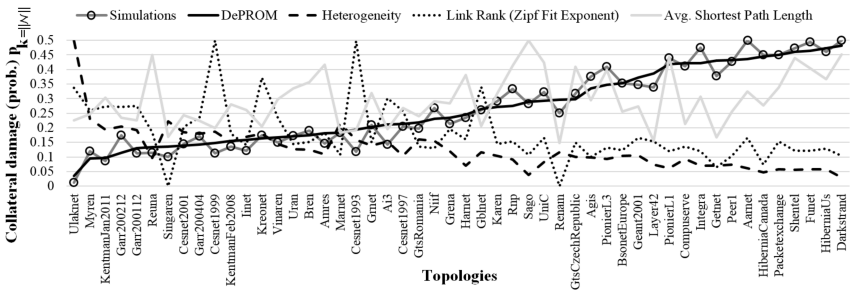


Figure 5: Comparison of collateral damage probabilities derived via simulations, DePROM and related metrics. Maximal attack budgets are assumed ($k = \|\mathcal{N}\|$). The values of heterogeneity, link rank and avg. shortest path length are scaled in $[0, 1/2]$ for ease of exposition.

count of 2, for most node pairs. Thus, most links jointly serve just two nodes. As a consequence of equation (12), the p_k plot versus the attack budget k, exhibits a minor linear increase and ends to a flat line near zero.

The "Grnet" case (Fig. 4b) contains more non-zero profile terms than the "Ulaknet" network. Nonetheless, the higher order terms are still absent, resulting in low p_k values (but higher than "Ulaknet"). Moreover, the layout is no longer a clear star graph, but rather resembles a graph of smaller stars interconnected via random links. The "PionierL1" case (Fig. 4c) exhibits a more complete detectability profile "Grnet" and "Ulaknet". The graph of "PionierL1" does not resemble a star, but retains the characteristic of small, uniform diameter. Finally, the "Darkstrand" case has an essentially complete detectability profile, containing almost all possible terms (Fig. 4d). As a result, its p_k value approximates the boundary value of $1/2$. Its form also exhibits a considerably larger diameter. Particularly, the left and right parts of the graph are connected via long paths. Subsequently, each of the links comprising these paths serve a high number of many nodes, which are affected by susceptible attacks.

Summarizing, the detectability profile describes the following aspects of a graph. First, the absence of higher order terms lowers the maximal p_k value of the network ($p_{\|\mathcal{N}\|}$), as discussed in the context of relation (13). Secondly, when more high order terms are absent, the graph essentially tends to become a star layout, as discussed in the context of relation (12).

B. Simulation Results

We proceed to compare DePROM and the related metrics via realistic network simulations. We use an open-source simulator of Crossfire attacks [16] which implements the attack ([6]) and defense process overviewed in Section II. The simulator is implemented in JAVA and runs on the AnyLogic platform [42]. The attacker is represented by a set of bots that continuously execute a Crossfire attack, attempting to cut-off the target from the gateway. The defender is represented by a TE process that is triggered on link-overload events and deploys new routing rules, seeking to minimize the maximum link utilization. In each run, the simulator receives as inputs: *(i)* a graph, *(ii)* the identity of a target node, and *(iii)* the identity of a gateway node.

At the run initialization stage, we select a random node to serve as gateway and deploy an OSPF set of routing rules in each network node. Link capacities are picked at random from the uniform range $[0.5, 1.5]$ Gbps. Furthermore, we instantiate 1000 benign traffic sources (IPs) and 1000 malevolent traffic sources (default simulator values). Each traffic source can exert a flow of 1.5 Mbps, without discrimination. These numbers indicate that the attacker has full attack budget potential. The benign flows pick a random (uniform) network node as their destination and exert their traffic.

The run operational stage comprises 20 cycles of bot attacks and subsequent TE rerouting for load balancing. The output is the *simulated* collateral damage probability. It is calculated as the number of times (out of 20) that each non-targeted node was present in the affected node sets, averaged over all non-targeted nodes. Runs are executed for each of the 100 TopologyZoo networks, and for 20 randomly picked target-gateway pairs within each network. Figure 5 compares the simulation output, the DePROM measurements as well as the related link rank exponent, average short path length and heterogeneity metrics (showing 50 topologies for brevity). The values of all metrics apart from DePROM are scaled to fit in $[0, 1/2]$ for ease of exposition.

$p_{k=\|\mathcal{N}\|}$(simulations) Vs: ↓	Corr. Coeff.	P-Value
$p_{k=\|\mathcal{N}\|}$ **(DePROM)**	$\mathbf{0.956}$	$\mathbf{10^{-7}}$
Heterogeneity	$\mathbf{-0.877}$	$\mathbf{2 \cdot 10^{-7}}$
Network Centrality	-0.612	10^{-7}
Avg. Neighbors	0.585	$2 \cdot 10^{-7}$
Link Rank (Zipf Fit Exp.) [18]	-0.583	$2 \cdot 10^{-7}$
Avg. Shortest Path Length [16]	0.385	$8 \cdot 10^{-5}$
Clustering Coefficient	0.283	$4 \cdot 10^{-3}$
Betweenness Centrality (Zipf Fit Exp.)	-0.233	10^{-2}
Network Density	0.188	$6 \cdot 10^{-2}$
Avg. Connected Node Pairs	-0.087	$3 \cdot 10^{-1}$

Table II: Correlation (Spearman rank [43]) of simulation results versus several topology metrics, including the proposed DePROM.

Notably, DePROM provides the best match to the simulation results. Moreover, DePROM is the only metric that is naturally bounded in $[0, 1/2]$. Therefore, a single DePROM measurement in a given topology can be used directly for deducing the exact collateral damage (or, equivalently, the detection) probability. The related metrics are not bounded, meaning that their values have comparative ordering value only. In other words, one can only perceive them as indications that one topology is more/less vulnerable than another, without quantification or significant precision in this statement.

The link rank exponent and heterogeneity metrics exhibit an inverse relation to the simulation outputs. However, the heterogeneity metric exhibits fewer fluctuations than the link rank. The later exhibits several fluctuations even to minimal and maximal values, for intermediate plot points (e.g. at topologies "Singaren", "Cesnet1999", "Kreonet", etc.). Finally, the average shortest path length does not exhibit an obvious statistical relation to the simulation results.

We proceed to quantify the relation between the simulation results, DePROM and all compared metrics using statistical correlation. To this end, we return to Fig. 5 and correlate the measured collateral damage probability (y-axis), to several topological metrics of the x-axis networks. Specifically, we use the Cytoscape tool to calculate the metrics of interest [19] and present them in table form (Table II).

To obtain Table II we create a 100×11 array, where the rows correspond to the 100 real topologies tested in the simulations (x-axis, Fig. 5). The first 10 columns refer to the graph metrics listed in the labels of Table II. The last (11^{th}) column contains the simulation values derived via the preceding simulations in each topology (x-axis, Fig. 5). Finally, we calculate the Spearman rank (i.e., correlation) between the last column and each of the first 10 columns [43], filling in the corresponding entries of Table II. The Table rows are listed by descending absolute correlation coefficient value.

DePROM yields the best result with an almost perfect

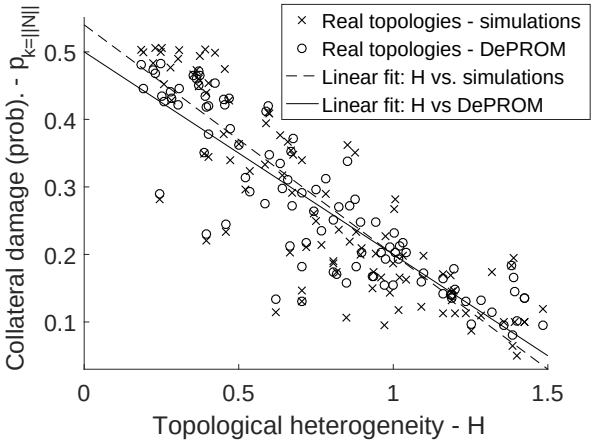


Figure 6: Scatter plot showing the statistical relation between the collateral damage probability (simulated and DePROM-derived) and the heterogeneity metric.

coefficient (0.956), and a P-value of 10^{-7}, i.e., much below the acceptable upper threshold (0.05) [43]. Among the remaining metrics, heterogeneity is the only one that stands out with a significant correlation coefficient. To understand this outcome, we revisit the definition of heterogeneity, $H(W)$, of a network W [41]:

$$H(W) = \frac{\sqrt{Variance(NB_n)}}{Average(NB_n)} \tag{15}$$

where NB_n is an array containing the number of direct neighbors (node degree) of each node n in the network W. At first, note that $H(W)$ captures both averages and fluctuations over node connectivity, contrary to other metrics listed in Table II. Moreover, equation (15) states that heterogeneity is zero in mesh-topologies, since NB_n will be equal over all nodes n, yielding zero variance. Therefore, the more mesh-like the topology, the lowest its heterogeneity. Additionally, consider a star-topology of $\|\mathcal{N}\| \gg 1$ nodes. In such a layout, $NB_n = 1$ for $\|\mathcal{N}\| - 1$ nodes, while $NB_n = \|\mathcal{N}\| - 1$ for the center node. Thus, for $\|\mathcal{N}\| \gg 1$ it holds that:

$$Average(NB_n) = \frac{2 \cdot (\|\mathcal{N}\| - 1)}{\|\mathcal{N}\|} \approx 2 \tag{16}$$

$$\sqrt{Var(NB_n)} = \sqrt{\frac{\sum_n (NB_n - 2)^2}{\|\mathcal{N}\|}} = \sqrt{\frac{(\|\mathcal{N}\|-1)+(\|\mathcal{N}\|-3)^2}{\|\mathcal{N}\|}} \approx \sqrt{\|\mathcal{N}\|} \tag{17}$$

Therefore, heterogeneity increases for larger star topologies. These observations are aligned with the topological interpretation of Fig. 4, explaining qualitatively its high correlation to the heterogeneity metric.

Finally, given the statistical significant of heterogeneity, we proceed to study its relation to DePROM in more detail in Fig. 6. Each point in this scatter plot corresponds

to one of the 100 topologies studied in the preceding evaluation. Different markers are assigned to the simulations-derived and DePROM-derived values of $p_{k=\|\mathcal{N}\|}$. (Notice that the heterogeneity values are *not* scaled to $[0, 1/2]$ in this Figure). We proceed to derive statistically a linear relation connecting $p_{k=\|\mathcal{N}\|}$ and heterogeneity. By applying linear fitting to the heterogeneity versus DePROM scatter plot, we obtain:

$$p_{k=\|\mathcal{N}\|}(H) = 0.5 - 0.3 \cdot H \tag{18}$$

with a root mean square error (RMSE) of 0.05753. Repeating again the linear fit process for the simulation-derived outcomes versus heterogeneity, we obtain the very similar relation:

$$p^*_{k=\|\mathcal{N}\|}(H) = 0.54 - 0.34 \cdot H \tag{19}$$

with a similar RMSE of 0.072. These relations show promise towards deriving a simple topology metric similar to heterogeneity, with absolute relation to p_k and, subsequently, to the Crossfire attack detection probability. As discussed earlier in this Section, heterogeneity itself has comparative value. It can only be used as indication that one topology is more/less vulnerable than another, without quantification or significant precision in this statement. Deriving such a metric using the relations of (18), (19) as a basis constitutes a promising extension of the present work.

VI. Related Work

Studer et al. introduced the Coremelt attack [5], where swarms of attack bots send traffic among each other in order to indirectly cause significant congestion within core network links. Older link-flooding attacks used the bot swarm to directly attack a target, thereby causing direct DDoS [9]. Thus, the Coremelt attack constituted a generalization of the existing link-flooding DDoS attacks. The Crossfire attack provided an additional generalization [6], assuming that bots can also use benign, existing servers as decoy destinations. Both Coremelt and Crossfire require bot orchestration, which can accomplished stealthily as well [7]. In this work, we study analytically such link-flooding DDoS attacks, decoupled from traffic patterns, and propose metrics to quantify the Crossfire detection potential in a given physical graph.

We note that related research around Crossfire attacks has focused on practical detection processes, but not on the effects of the topology. For example, Xue et al. propose the LinkScope system for detecting malicious link floods and for locating the target link or area whenever possible [12]. Their system uses end-to-end and hop-by-hop network measurement techniques to detect abrupt degradation of performance. It is also worth noting that more generic DDoS defenses, such as Ingress [44], PacketScope [45] and Pushback [46], can be useful in the Crossfire case as well. The survey of Bhuyan et al. [13] gives an overview of the methods and tools used for detecting DDoS attacks. In contrast to these works, we do not propose explicit detection schemes. Instead, we study the detection potential of an attack within a given graph, which could provide an indication of the expected end-performance of explicit detector algorithms in general.

The Crossfire and similar attack types have been studied in light of the SDN paradigm shift as well. Braga et al. capitalize on controller features for traffic analysis using Self Organizing Maps (SOMs) to classify flows and enable DDoS attack detection caused by heavy hitters [47]. Lim et al. propose a SDN-based scheme to block botnet-based DDoS attacks that do not exhibit detectable statistical anomalies [48]. They focus on HTTP communications between clients and servers and they employ CAPTCHA challenges for HTTP URL redirection. Kang et al present a re-routing scheme that forces malevolent flows to increase their traffic volume, leading to their exposure [14]. The studies of [16] and [15] study source-based and destination-based routing respectively. They apply TE in a manner that forces malevolent flows to constantly re-home to new destinations, accelerating their detection. The recent work of Lee et al. (CoDef) studies Crossfire attacks at inter-AS level, and highlights the importance of AS collaboration for security purposes [49]. They propose a practical, cooperative TE-based detection method for identifying low-rate attack traffic. These methods can also be classified as explicit detection schemes.

The study of [18] examined the role of topology traits to routing bottlenecks in general. The study deduces the qualitative best-practices for defending against common DDoS exploits, including Crossfire attacks. Extensive measurements show that path diversity limits link sharing, facilitating load balancing. On the contrary, the present study does not refer to factors that facilitate the attack *execution* within a network, but rather to attack *detection potential* within a given graph.

Finally, in their previous work [16], the authors proposed a novel framework for *detecting* Crossfire and Coremelt attacks within a network, employing a novel combination of TE and reinforcement learning. This study contributed: i) an analytical model for the Crossfire and Coremelt attacks, with general applicability to multi-graphs, multipath routing and generic bot behavior, ii) A lightweight detection process based on reinforcement learning, described in Section II, iii) A novel way of incorporating the detection process within existing TE modules, without affecting their operation. Statistical analysis hinted that the network topology affects the

detection efficiency, which was designated as a future research direction.

The present work was motivated by: i) this preceding *evidence*, i.e., that the topology might significantly affect the attack detection, and ii) the fact that physical changes in a topology are expensive and slow. Therefore, network designers should have a metric to quantify how a perspective topological change will affect the detection efficiency of an attack. The present study fills this gap, contributing: i) a *formal proof* of the relation between a network graph and the attack detection efficiency, ii) a novel metric which quantifies this relation and an algorithm for calculating it, iii) practical insights on the relation, e.g., showing that detection is more efficient in star-like topologies, and that the higher the heterogeneity of a graph, the higher the detection efficiency.

VII. Conclusion and Future Work

The present work studied recent, sophisticated DDoS link-flooding attacks which have exhibited potential to segment the Internet while remaining stealthy. Using analysis, it was shown that network graph attributes can influence the attack target detection potential within a given network layout. Novel, security-oriented topological metrics were proposed and extensively compared to existing ones via simulations, in multiple real Internet topologies. The novel metrics were found to effectively capture the security attributes of a topology better than any other related approach.

Future work will focus on using the novel metrics for security-aware topology design and evolution, facilitating their proactive fortification. The authors aim at developing evolutionary algorithms that seek to increase the attack detection efficiency within a topology with a constrained number of topological changes. In this sense, the installation of new physical links or nodes can become security-aware. Application scenarios in consideration also include inter-autonomous system peering agreements, where the aim is to provide an additional security dimension when evaluating perspective peering decisions.

Acknowledgment

This work was funded by the European Union project CIPSEC: “Enhancing critical Infrastructure protection with an innovative security framework”, grant number EU700378. (www.cipsec.eu).

References

[1] M. Prince, “The DDoS That Almost Broke The Internet,” 2013. [Online]. Available: http://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet

[2] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, “Botnet in DDoS Attacks: Trends and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015.

[3] B. Bright, “Can a DDoS break the Internet? Sure... just not all of it.” *Ars Technica [Online:] https://arstechnica.com/information-technology/2013/04/can-a-ddos-break-the-internet-sure-just-not-all-of-it/*, 2013.

[4] J. J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Z. Granville, and A. Pras, “Booters — An analysis of DDoS-as-a-service attacks,” in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Ottawa, Canada, May, 2015, pp. 243–251.

[5] A. Studer and A. Perrig, “The Coremelt Attack,” in *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, Saint-Malo, France, September, 2009, pp. 37–52.

[6] M. S. Kang and S. B. Lee, “The Crossfire Attack,” in *Proceedings of the IEEE Security and Privacy symposium (SP)*, San Francisco, California, USA, May, 2013, pp. 127–141.

[7] Y.-M. Ke, C.-W. Chen, H.-C. Hsiao, A. Perrig, and V. Sekar, “CICADAS: Congesting the Internet with Coordinated and Decentralized Pulsating Attacks,” in *Proceedings of the ACM Asia Conference on Computer and Communications Security (ACCCS)*, Xi’an, China, May, 2016, pp. 699–710.

[8] H. Li, J. Zhu, Q. Wang, T. Zhou, H. Qiu, and H. Li, “LAAEM: A Method to Enhance LDoS Attack,” *IEEE Communications Letters*, vol. 20, no. 4, pp. 708–711, 2016.

[9] Y. Zhang, Z. M. Mao, and J. Wang, “Low-Rate TCP-Targeted DoS Attack Disrupts Internet Routing,” in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, February, 2007, pp. 1–15.

[10] C. Assi, S. Ayoubi, S. Sebbah, and K. Shaban, “Towards Scalable Traffic Management in Cloud Data Centers,” *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 1033–1045, 2014.

[11] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, and K. Han, “Botnet research survey,” in *Computer Software and Applications, 2008. COMPSAC’08. 32nd Annual IEEE International*. IEEE, 2008, pp. 967–972.

[12] L. Xue, Z. Luo, W. Chan, and C. Zhan, “Towards Detecting Target Link Flooding Attack,” in *Proceedings of the USENIX conference*, Seattle, WA, USA, November, 2014, pp. 81–96.

[13] M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya, and J. K. Kalita, “Detecting distributed denial of service attacks: methods, tools and future directions,” *The Computer Journal*, p. bxt031, 2013.

[14] M. S. Kang, V. D. Gligor, and V. Sekar, “SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks,” in *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, February, 2016, pp. 1–15.

[15] D. Gkounis, V. Kotronis, C. Liaskos, and X. Dimitropoulos, “On the Interplay of Link-Flooding Attacks and Traffic Engineering,” *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 1, pp. 5–11, 2016.

[16] C. Liaskos, V. Kotronis, and X. Dimitropoulos, “A Novel Framework for Modeling and Mitigating Distributed Link Flooding Attacks,” in *Proceedings of IEEE INFOCOM*, San Francisco, CA, USA, April, 2016, pp. 1–9.

[17] A. P. Athreya, X. Wang, Kim Y. S., and Y. Tian, “Resistance Is Not Futile: Detecting DDoS Attacks without Packet Inspection,” in *Proceedings of of the International Workshop on Information Security Applications (WISA)*, Jeju Island, Korea, August, 2014, pp. 1 – 15.

[18] M. S. Kang and V. D. Gligor, “Routing bottlenecks in the internet: Causes, exploits, and countermeasures,” in *Proceedings of the ACM SIGSAC*, Scottsdale, Arizona, USA, November, 2014, pp. 321–333.

[19] S. Killcoyne *et al.*, “Cytoscape: A Community-based Framework for Network Modeling,” in *Protein Networks and Pathway Analysis*. Springer, 2009, pp. 219–239.

[20] R. Albert and A. Barabasi, “Topology of evolving networks: local events and universality,” *Physical review letters*, vol. 85, no. 24, p. 5234, 2000.

[21] M. Caesar and J. Rexford, “BGP Routing Policies in ISP Networks,” *IEEE Network*, vol. 19, no. 6, pp. 5–11, 2005.

[22] C. E. Hopps, "Analysis of an Equal-Cost Multi-path Algorithm," *IETF RFC 2992*, 2000.

[23] A. Aydeger, N. Saputro, K. Akkaya, and M. Rahman, "Mitigating crossfire attacks using sdn-based moving target defense," in *Local Computer Networks (LCN), 2016 IEEE 41st Conference on*. IEEE, 2016, pp. 627–630.

[24] B. Al-Musawi, P. Branch, and G. Armitage, "Bgp anomaly detection techniques: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 377–396, 2017.

[25] V. Giotsas, A. Dhamdhere, and K. C. Claffy, "Periscope: Unifying looking glass querying," in *International Conference on Passive and Active Network Measurement*. Springer, 2016, pp. 177–189.

[26] R. Staff, "Ripe atlas: A global internet measurement network," *Internet Protocol Journal*, vol. 18, no. 3, 2015.

[27] M. Sharif, K. Singh, J. Giffin, and W. Lee, "Understanding precision in host based intrusion detection," in *RAID*. Springer, 2007, pp. 21–41.

[28] D. Gkounis, "Cross-domain DoS Link-flooding Attack Detection and Mitigation Using SDN Principles," *MSc Thesis, ETH Zurich*, 2014.

[29] D. Michie *et al.*, Eds., *Machine Learning, Neural and Statistical Classification*. Upper Saddle River, NJ, USA: Ellis Horwood, 1994.

[30] K. Malialis and D. Kudenko, "Distributed response to network intrusions using multiagent reinforcement learning," *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 270–284, 2015.

[31] ——, "Large-scale ddos response using cooperative reinforcement learning," in *11th European Workshop on Multi-Agent Systems (EUMAS)*, 2013.

[32] J. Kaur, R. Singh, and P. Kaur, "Prevention of ddos and brute force attacks on web log files using combination of genetic algorithm and feed forward back propagation neural network," *International Journal of Computer Applications*, vol. 120, no. 23, 2015.

[33] Q. Li, "Mobile internet anomaly traffic detection technology research based on improved wavelet neural network," *Journal of Residuals Science & Technology*, vol. 13, no. 5, 2016.

[34] J. Ashraf and S. Latif, "Handling Intrusion and DDoS Attacks in SDNs using Machine Learning," in *IEEE NSEC*, 2014.

[35] T. Bonald and J. W. Roberts, "Internet and the Erlang formula," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 1, pp. 23–30, 2012.

[36] R. Klöti, V. Kotronis, A. Bernhard, and X. Dimitropoulos, "Policy-Compliant Path Diversity and Bisection Bandwidth," in *Proceedings of IEEE INFOCOM*, Hong Kong, China, April, 2015, pp. 675–683.

[37] D. Eppstein, "Finding the K Shortest Paths," *SIAM Journal of Computing*, vol. 28, no. 2, pp. 652–673, 1999.

[38] U. Brandes, "A faster algorithm for betweenness centrality*," *The Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[39] T. Leighton and S. Rao, "An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms," in *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pp. 422–431.

[40] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[41] J. Dong and S. Horvath, "Understanding network concepts in modules," *BMC systems biology*, vol. 1, no. 1, p. 24, 2007.

[42] X. Technologies, "The AnyLogic Simulator," *[Online:] http://www.xjtek.com/anylogic*, 2015.

[43] J. H. Zar, "Significance testing of the Spearman rank correlation coefficient," *Journal of the American Statistical Association*, vol. 67, no. 339, pp. 578–580, 1972.

[44] C. E. Hopps, "Ingress Filtering for Multihomed Networks," *RFC Editor, RFC 3704*, 2004.

[45] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "PacketScore: Statistics-based overload control against distributed denial-of-service attacks," in *Proceedings of IEEE INFOCOM*, Torondo, CA, April, 2004, pp. 2594–2604.

[46] J. Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, California, February, 2002, pp. 1 – 12.

[47] R. Braga, E. Mote, and Passito A., "Lightweight DDoS Flooding Attack Detection using NOX/OpenFlow," in *Proceedings of the IEEE Local Computer Networks (LCN)*, Denver, Colorado, U.S.A., October, 2010, pp. 408–415.

[48] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "An SDN-oriented DDoS Blocking Scheme for Botnet-based Attacks," in *Proceedings of IEEE International Conference on Ubiquitous and Future Networks (ICUFN)*, Shanghai, China, July, 2014, pp. 1–6.

[49] S. B. Lee, M. S. Kang, and V. D. Gligor, "CoDef: Collaborative Defense Against Large-scale Link-flooding Attacks," in *Proceedings of ACM CoNEXT*, Santa Barbara, California, USA, December, 2013, pp. 417–428.

[50] P. G. Ciarlet and J. L. Lions, *Handbook of Numerical Analysis*. North-Holland, ISBN-9780444703668, 1990.

APPENDIX

Proof of Result 6. We will show that: 1) p_k is a strictly rising function. 2) $p_1 = 0$, $p_{\frac{\|\mathcal{N}\|}{2}} \approx \frac{1}{2}$, $p_{\|\mathcal{N}\|} \approx \frac{1}{2}$, 3) $p_k \leq \frac{k-1}{\|\mathcal{N}\|}$, and 4) p_k is concave for $k \in \left[1, \frac{\|\mathcal{N}\|}{2}\right]$.

Claim 1. Let $\Delta p = p_{k+1} - p_k$. We will show that $\Delta p > 0$. First, for ease of presentation, we define the f_k and g_k helping quantities:

$$f_k = \sum_{m=1}^{k} \binom{\|\mathcal{N}\| - 2}{m-2},\ g_k = \sum_{m=1}^{k} \binom{\|\mathcal{N}\| - 1}{m-1} \quad (20)$$

Then, via standard finite differences, we have [50]:

$$\Delta p = \Delta\left(\frac{f_k}{g_k}\right) = \frac{\Delta f \cdot g_k - f_k \cdot \Delta g}{g_k \cdot (g_k + \Delta g)} \quad (21)$$

Furthermore, from the definition of f_k, g_k in equation (3) we obtain:

$$\Delta g = \begin{pmatrix} \|\mathcal{N}\| - 1 \\ k \end{pmatrix} \text{ and } \Delta f = \begin{pmatrix} \|\mathcal{N}\| - 2 \\ k - 1 \end{pmatrix} \quad (22)$$

Notice that $\Delta g,\ \Delta f,\ g_k,\ f_k > 0$. Therefore, the denominator of equation (21) is strictly positive and we focus on the numerator, rewriting equation (21) as:

$$\Delta p \propto \Delta f \cdot \left(g_k - f_k \frac{\Delta g}{\Delta f}\right) \Rightarrow \Delta p \propto \left(g_k - f_k \frac{\Delta g}{\Delta f}\right) \quad (23)$$

Moreover, from equations (22) we obtain:

$$\frac{\Delta g}{\Delta f} = \frac{\|\mathcal{N}\| - 1}{k} \quad (24)$$

Thus, equations (23) and (22) yield:

$$\Delta p \propto \cdot \left(g_k - f_k \frac{\|\mathcal{N}\| - 1}{k}\right) \quad (25)$$

From the f_k definition in equation (3) can be rewritten as:

$$f_k = \sum_{m=1}^{k} \frac{m-1}{\|\mathcal{N}\| - 1} \binom{\|\mathcal{N}\| - 1}{m-1} \quad (26)$$

Finally, we substitute g_k and f_k from equation (3) and (26) into (25) producing:

$$\Delta p \propto \sum_{m=1}^{k} \left(1 - \frac{m-1}{k}\right) \binom{\|\mathcal{N}\| - 1}{m-1} \tag{27}$$

Since $m \in [1, k]$, $\left(1 - \frac{m-1}{k}\right)$ is strictly positive. Hence:

$$\Delta p > 0 \Longleftrightarrow p_{k+1} > p_k, \text{ QED.} \tag{28}$$

Claim 2. Since $f_1 = 0$ and $g_1 = \binom{\|\mathcal{N}\|}{1} = \|\mathcal{N}\|$, we have:

$$p_1 = \frac{f_1}{g_1} = 0 \tag{29}$$

Using the identity $\sum_{m=0}^{x} \binom{x}{m} = 2^x$, we have that $f_k = 2^{\|\mathcal{N}\|-2} - 1$ and $g_k = 2^{\|\mathcal{N}\|-1} - 1$. Therefore:

$$p_{\|\mathcal{N}\|} \approx \frac{1}{2}, \ \|\mathcal{N}\| \gg 1 \tag{30}$$

At this point, notice that p_k is bounded in $\left[0, \frac{1}{2}\right]$, due to Claim 1 and equations (29) and (30).

Proceeding to the value $p_{\frac{\|\mathcal{N}\|}{2}}$, we remark that the function $F(S, m) = \binom{S}{m}$ is always positive and symmetric around $m = \frac{S}{2}$, for $S \gg 1$. Therefore, it holds that: $\sum_{m=1}^{\frac{S}{2}} F(S, m) \approx \sum_{m=\frac{S}{2}}^{S} F(S, m) \approx \frac{1}{2} \sum_{m=1}^{S} F(S, m)$.
Returning to $p_{\frac{\|\mathcal{N}\|}{2}}$ we obtain that it is equal to:

$$\frac{\sum_{m=1}^{\frac{\|\mathcal{N}\|}{2}} F(\|\mathcal{N}\|-2, m-2)}{\sum_{m=1}^{\frac{\|\mathcal{N}\|}{2}} F(\|\mathcal{N}\|-1, m-1)} \approx \frac{\frac{1}{2}\sum_{m=1}^{\|\mathcal{N}\|} F(\|\mathcal{N}\|-2, m-2)}{\frac{1}{2}\sum_{m=1}^{\|\mathcal{N}\|} F(\|\mathcal{N}\|-1, m-1)}$$
$$\Rightarrow p_{\frac{\|\mathcal{N}\|}{2}} \approx \frac{f_{\|\mathcal{N}\|}}{g_{\|\mathcal{N}\|}} \overset{\text{eq. (30)}}{=} \frac{1}{2}, \ \text{QED} \tag{31}$$

Claim 3. We will first prove the utility:

$$\Delta p \leq \frac{1}{\|\mathcal{N}\|} \tag{32}$$

It must hold that:

$$\frac{\Delta f \cdot g_k - f_k \cdot \Delta g}{g_k \cdot (g_k + \Delta g)} = \frac{\Delta g \left(\frac{\Delta f}{\Delta g} \cdot g_k - f_k\right)}{g_k \cdot (g_k + \Delta g)} \leq \frac{1}{\|\mathcal{N}\|} \tag{33}$$

or, equivalently, and replacing $\frac{\Delta f}{\Delta g}$ from equation (24):

$$\Delta g \left[k \cdot g_k - (\|\mathcal{N}\| - 1) f_k\right] \leq g_k (g_k + \Delta g) \Leftrightarrow$$
$$\frac{\Delta g}{g_k} \left[k \cdot g_k - (\|\mathcal{N}\| - 1) f_k\right] \leq g_k + \Delta g \tag{34}$$

We focus on the term $a = \left[k \cdot g_k - (\|\mathcal{N}\| - 1) f_k\right]$ and replace g_k, f_k with their definitions, producing:

$$a = \sum_{m=1}^{k} k \binom{\|\mathcal{N}\| - 1}{m-1} - \sum_{m=1}^{k} (\|\mathcal{N}\| - 1) \binom{\|\mathcal{N}\| - 2}{m-2} =$$
$$\sum_{m=1}^{k} \frac{k(\|\mathcal{N}\|-1)}{m-1} \binom{\|\mathcal{N}\|-2}{m-2} - \sum_{m=1}^{k} (\|\mathcal{N}\|-1) \binom{\|\mathcal{N}\|-2}{m-2} =$$
$$\sum_{m=1}^{k} (\|\mathcal{N}\| - 1) \left(\frac{k}{m-1} - 1\right) \binom{\|\mathcal{N}\| - 2}{m-2} \tag{35}$$

Notice that, since $1 \leq m \leq k \leq \|\mathcal{N}\|$, it always holds:

$$(\|\mathcal{N}\| - 1) \left(\frac{k}{m-1} - 1\right) = \frac{(\|\mathcal{N}\| - 1)}{m-1} (k - m + 1) \geq 1 \tag{36}$$

Therefore, from equation (35):

$$\sum_{m=1}^{k} 1 \cdot \binom{\|\mathcal{N}\| - 2}{m-2} \leq a \Leftrightarrow g_k \leq a \tag{37}$$

Thus, inequality (34) becomes:

$$\frac{\Delta g}{g_k} g_k \leq \frac{\Delta g}{g_k} a \leq g_k + \Delta g \Rightarrow \Delta g \leq g_k + \Delta g \tag{38}$$

which is always true, proving the utility $\Delta p \leq 1/\|\mathcal{N}\|$.

We will work by induction to prove Claim 3. For $k = 1$ it holds that $p_1 \leq 0$, in accordance with Claim 2. Let a $k = K$ for which: $p_K \leq \frac{K-1}{\|\mathcal{N}\|}$. Then, we derive:

$$p_K \leq \frac{K-1}{\|\mathcal{N}\|} \Leftrightarrow p_K - p_{K+1} + p_{K+1} \leq \frac{K-1}{\|\mathcal{N}\|} \Leftrightarrow$$
$$p_{K+1} \leq \frac{K-1}{\|\mathcal{N}\|} + \Delta p \overset{\text{eq. (32)}}{\Leftrightarrow} p_{K+1} \leq \frac{K}{\|\mathcal{N}\|} \tag{39}$$

Thus, it always holds that $p_k \leq \frac{k-1}{\|\mathcal{N}\|}$, QED.
Claim 4. We will show that $\Delta^2 p < 0$. We have [50]:

$$\Delta^2 (p_k) = \Delta^2 \left(\frac{f_k}{g_k}\right) =$$
$$\frac{f_{k+2} \cdot g_k \cdot g_{k+1} - 2 \cdot f_{k+1} \cdot g_k \cdot g_{k+2} + f_k \cdot g_{k+1} \cdot g_{k+2}}{g_k \cdot g_{k+1} \cdot g_{k+2}} \tag{40}$$

We study the properties of the terms f_{k+1}, f_{k+2}, g_{k+1}, g_{k+2}.
Initially, it holds that:

$$f_{k+1}, \ f_{k+2}, \ g_{k+1}, \ g_{k+2} > 0 \tag{41}$$

and, therefore, we can focus on the numerator of equation (40). Then, we write:

$$g_{k+1} = g_k + (g_{k+1} - g_k) = g_k + \Delta g \tag{42}$$

where $\Delta g = g_{k+1} - g_k$. In addition, we write:

$$g_{k+2} = g_k + (g_{k+1} - g_k) + (g_{k+2} - g_{k+1}) \tag{43}$$

From the definition of g_k in equation (3), we obtain that:

$$g_{k+2} - g_{k+1} = \binom{\|\mathcal{N}\|-1}{k+1} = \frac{\|\mathcal{N}\| - k - 1}{k+1} \cdot \binom{\|\mathcal{N}\|-1}{k} \tag{44}$$

$$\Delta g = g_{k+1} - g_k = \binom{\|\mathcal{N}\| - 1}{k} \tag{45}$$

Thus, equations (44) and (45) transform equation (43) as:

$$g_{k+2} = g_k + \Delta g + \frac{\|\mathcal{N}\| - k - 1}{k+1} \cdot \Delta g \Leftrightarrow$$
$$g_{k+2} = g_k + \frac{\|\mathcal{N}\|}{k+1} \cdot \Delta g \quad (46)$$

Working similarly for f_{k+1} and f_{k+2} we obtain:

$$f_{k+1} = f_k + \Delta f \quad (47)$$
$$f_{k+2} = f_k + \frac{\|\mathcal{N}\| - 1}{k} \cdot \Delta f \overset{\text{eq. (24)}}{=} f_k + \Delta g \quad (48)$$

Returning to equation (40), we notice that the denominator is once again strictly positive due to relations (41). Therefore we focus on sign of the numerator. Using equations (42), (46), (47), (48) and the condition $\|\mathcal{N}\| \gg 1$ in the numerator of (40) produces:

$$\Delta^2 p_k \propto -[k \cdot g_k - \|\mathcal{N}\| f_k] \cdot [(2k - \|\mathcal{N}\|) \cdot g_k + \|\mathcal{N}\| \Delta g] \quad (49)$$

We will show that both terms in brackets are positive for $\|\mathcal{N}\| \gg 1$ and $k \in [1, {}^{\|\mathcal{N}\|}/_2]$. For the first term, we replace the definitions of f_k, g_k, producing:

$$\sum_{m=1}^{k} k \binom{\|\mathcal{N}\| - 1}{m-1} - \sum_{m=1}^{k} \|\mathcal{N}\| \binom{\|\mathcal{N}\| - 2}{m-2} =$$
$$\sum_{m=1}^{k} \left(k - \frac{\|\mathcal{N}\| (m-1)}{\|\mathcal{N}\| - 1} \right) \binom{\|\mathcal{N}\| - 1}{m-1} \quad (50)$$

which is positive for every $m \leq k$ when $\|\mathcal{N}\| \gg 1$. The second term of (49) in brackets holds for $k = 1$:

$$2 - \|\mathcal{N}\| + \|\mathcal{N}\| (\|\mathcal{N}\| - 1) > 0, \ \forall \|\mathcal{N}\| \quad (51)$$

Let a $k = K$ for which it holds that:

$$(2K - \|\mathcal{N}\|) \cdot g_K + \|\mathcal{N}\| \Delta g > 0 \overset{(45)}{\Rightarrow}$$
$$g_K < \frac{\|\mathcal{N}\|}{(\|\mathcal{N}\| - 2K)} \binom{\|\mathcal{N}\| - 1}{K} \quad (52)$$

We will show that it also holds for $k = K + 1$. We add $\binom{\|\mathcal{N}\|-1}{K}$ in both parts of (52), producing g_{K+1} to the left, and will show that:

$$g_{K+1} < \frac{2\|\mathcal{N}\| - 2K}{(\|\mathcal{N}\| - 2K)} \binom{\|\mathcal{N}\| - 1}{K} < \frac{\|\mathcal{N}\|}{(\|\mathcal{N}\| - 2(K+1))} \binom{\|\mathcal{N}\| - 1}{K+1} \quad (53)$$

Simplifying the binomials yields:

$$\frac{2(\|\mathcal{N}\| - K)}{(\|\mathcal{N}\| - 2K)} \frac{1}{(\|\mathcal{N}\| - K - 1)} < \frac{\|\mathcal{N}\|}{(\|\mathcal{N}\| - 2K - 2)} \frac{1}{(K+1)} \quad (54)$$

Finally, using $\|\mathcal{N}\| - 2 \approx \|\mathcal{N}\| - 1 \approx \|\mathcal{N}\|$, it is not difficult to see that (54) holds for $k \in [1, {}^{\|\mathcal{N}\|}/_2)$. Therefore,from (49), we obtain $\Delta^2 p_k < 0$ for the studied range of k, QED.

From Claims 1, 2 and 3, p_k must start from 0 and reach ${}^1/_2$ for $k = {}^{\|\mathcal{N}\|}/_2$, in a strictly increasing fashion, while remaining on or below the line $\frac{k-1}{\|\mathcal{N}\|}$, $\|\mathcal{N}\| \gg 1$. However, Claim 4 adds that p_k cannot be below this line, since this would upset its concavity. Thus, p_k is approximated by the line $\frac{k-1}{\|\mathcal{N}\|}$ for $k \in \left[1, \frac{\|\mathcal{N}\|}{2}\right]$. For $k \in \left[\frac{\|\mathcal{N}\|}{2}, \|\mathcal{N}\|\right]$, p_k must be almost constant and equal to $\frac{1}{2}$, due to Claims 1 (monotonicity) and 2 ($p_{\frac{\|\mathcal{N}\|}{2}} = p_{\|\mathcal{N}\|}$), which concludes the proof.

Proof of Result 9. Assume a network w and note that:

$$\Delta g(w) = \pi_k \cdot \Delta g \ \text{ and } \Delta f(w) = \pi_k \cdot \Delta f. \quad (55)$$

$\Delta f(w)$ and $\Delta g(w)$ are *jointly* 0 for $\pi_k = 0$, or $\neq 0$ otherwise.

Property 1. We repeat the above procedures of Result 6, Claims 1, 3 and 4. For Claim 1, note that, by equation (21), $\Delta p(w) = 0$ when $\Delta f(w), \Delta g(w) = 0$. If $\Delta f(w), \Delta g(w) \neq 0$, the proof of Claim 1 for $g_k(w)$ and $f_k(w)$ instead of g_k and f_k leads to the addition of a positive π_m factor within the sum of relation (27), which does not upset its sign. Thus, $\Delta p(w) \geq 0$, and $p_k(w)$ is non-strictly increasing. Working similarly, Claims 3 and 4 yield a linear upper bound in $\overset{w}{p}$ and concavity for $k \in \left[1, \frac{\|\mathcal{N}\|}{2}\right)$. Since $\Delta p(w)$ can be 0, the increase is sub-linear in the general case, QED.

Property 2. From Result 6, p_k is constant to $\frac{1}{2}$ for $k \in \left[\frac{\|\mathcal{N}\|}{2}, \|\mathcal{N}\|\right]$. Thus, it must hold that $\Delta p = 0$. Dividing the numerator and denominator of (21) by $g_k \cdot \Delta g$:

$$\Delta p = \frac{\frac{\Delta f}{\Delta g} - \frac{f_k}{g_k}}{\frac{g_k}{\Delta g} + 1} = 0, \ \forall k \in \left[\frac{\|\mathcal{N}\|}{2}, \|\mathcal{N}\|\right] \quad (56)$$

However, $\frac{f_k}{g_k} = p_k = \frac{1}{2}$, while from equation (24) we have that $\frac{\Delta f}{\Delta g}$ varies linearly in $\left(\frac{1}{2}, 1\right)$, for the studied range of k. Thus, the numerator of equation (56) is not zero. For (56) to hold for every studied k, it must hold that:

$$\frac{g_k}{\Delta g} \to \infty, \ \forall k \in \left[\frac{\|\mathcal{N}\|}{2}, \|\mathcal{N}\|\right] \quad (57)$$

Next, we study $\Delta \check{p}$, noting that by the Mean Value Theorem for sums, there exist π_F, $\pi_G \geq 0$:

$$f_k(w) = \sum_{m=1}^{k} \pi_F \binom{\|\mathcal{N}\| - 2}{m-2} = \pi_F \cdot f_k \ \text{ and } \ g_k(w) = \pi_G \cdot g_k \quad (58)$$

Using $f_k(w), g_k(w), \Delta f(w), \Delta g(w)$ in equation (56), and using the substitutions (55) and (58), we conclude that $\Delta p(w) = 0$ due to (57). If $\pi_F = 0$, then so does π_G, due to relation (55). In this case, from the form of (21), it also holds $\Delta p(w) = 0$, QED.

Proof of Result 10. We will show that $p_k(W)$ and p_k have the same initial value (for $k = 1$), p_k increases faster, and are both upper bounded at ${}^1/_2$.

From relations (6) and (9) we obtain for $k = 1$ that:

$$p_1(\text{mesh}) = p_1(W) = 0 \quad (59)$$

Additionally, Result 9 states that $p_k\,(\text{mesh})$ increases *linearly* with k, while $p\,(W)$ increases *sub-linearly* with k. Finally, we show by induction that $p_{\|\mathcal{N}\|}\,(W) \leq \frac{1}{2},\ \forall k$. Starting with relation (6), we require:

$$p_{\|\mathcal{N}\|}\,(W) = \frac{\sum_{m=1}^{\|\mathcal{N}\|} \pi_m \cdot \frac{m-1}{\|\mathcal{N}\|-1} \cdot \binom{\|\mathcal{N}\|-1}{m-1}}{\sum_{m=1}^{k} \pi_m \cdot \binom{\|\mathcal{N}\|-1}{m-1}} \leq \frac{1}{2} \Leftrightarrow$$

$$\sum_{m=1}^{\|\mathcal{N}\|} 2 \cdot \pi_m \cdot \frac{m-1}{\|\mathcal{N}\|-1} \cdot \binom{\|\mathcal{N}\|-1}{m-1} \leq \sum_{m=1}^{k} \pi_m \cdot \binom{\|\mathcal{N}\|-1}{m-1} \Leftrightarrow$$

$$0 \leq \sum_{m=1}^{\|\mathcal{N}\|} \pi_m \cdot \left(1 - 2 \cdot \frac{m-1}{\|\mathcal{N}\|-1}\right) \binom{\|\mathcal{N}\|-1}{m-1} \quad (60)$$

The binomial factor $\binom{\|\mathcal{N}\|-1}{m-1}$ is positive and symmetric in $m \in [1, \|\mathcal{N}\|]$ around $m_o = \frac{\|\mathcal{N}\|-1}{2} + 1$. The factor $\left(1 - 2 \cdot \frac{m-1}{\|\mathcal{N}\|-1}\right)$ is a linear ramp that ranges from -1 for $m = 1$ to $+1$ for $m = \|\mathcal{N}\|$, and becomes zero at $m_o = \frac{\|\mathcal{N}\|-1}{2} + 1$. Therefore, the quantity $\sum_{m=1}^{\|\mathcal{N}\|} \left(1 - 2 \cdot \frac{m-1}{\|\mathcal{N}\|-1}\right) \binom{\|\mathcal{N}\|-1}{m-1}$ by itself would be equal to zero. Thus, the right part of inequality (60) is indeed positive if the π_m distribution is positively skewed, i.e., if takes generally larger values for $m \in [1, m_o)$ than for $m \in (m_o, \|\mathcal{N}\|]$ which can be shown as follows.

First, notice that it always holds that $\pi_1 \neq 0$ (i.e., there is always a way to cut-off single nodes) while $\pi_{\|\mathcal{N}\|} = 0$ (since the opposite would imply cutting-off the complete network from another, non-existing node). Moreover, consider a path connecting any two nodes and comprising n links. Notice that there exist 2 service links with $\|\mathcal{N}\,\{l\}\| = 1$, 2 service links with $\|\mathcal{N}\,\{l\}\| = 2$,. . ., 2 service links with $\|\mathcal{N}\,\{l\}\| = n-1$ and 0 service links with $\|\mathcal{N}\,\{l\}\| = n$, which is a positively skewed distribution. Any complex routing can be decomposed to smaller linear paths (branches with $n < \|\mathcal{N}\|$), i.e., a composition of positively-skewed distributions. Thus, the overall distribution is positively skewed as well, QED.

Christos Liaskos received the Diploma in Electrical Engineering from the Aristotle University of Thessaloniki (AUTH), Greece in 2004, the MSc degree in Medical Informatics in 2008 from the Medical School, AUTH and the PhD degree in Computer Networking from the Dept. of Informatics, AUTH in 2014. He is currently a researcher at the Foundation of Research and Technology, Hellas (FORTH). His research interests include security in computer networks, traffic engineering and novel control schemes for wireless communications.

Sotiris Ioannidis received a BSc degree in Mathematics and an MSc degree in Computer Science from the University of Crete in 1994 and 1996 respectively. In 1998 he received an MSc degree in Computer Science from the University of Rochester. He received his PhD from the University of Pennsylvania (2005). His research interests are in the area of systems and security. Ioannidis has authored more than 100 publications in international conferences, journals and book chapters.