The Routing Impact of Link Failure Location: A Formal Analysis *

Xiaoliang Zhao[1], Beichuan Zhang[1], Dan Massey[1], Andreas Terzis[2], Lixia Zhang[3] †

Abstract

Routing instability inside the global routing system may adversely affect packet delivery performance. To improve the overall routing stability, we need a deep understanding on the deciding factors and their routing impacts. In this work, we focus on one factor, the link failure location. We are interested to understand whether link failures occurring at different places will have different routing impacts, such as the failure-induced routing changes and triggered routing traf£c; and what the differences would be if any. We build a formal analysis framework to gain solid understanding of this problem. At meantime, we also validate our analytical results with simulations to avoid oversimpli£cation problem during formal analysis. Our results show that link failure occurs at different locations, such as at the edge or at the core of a network, will have different routing impacts. Our analysis formally models and captures such differences.

1 Introduction and Motivation

Routing instability inside the global routing system, BGP in particular, may adversely affect packet delivery performance. To improve the overall routing stability, we need a deep understanding on the deciding factors and their routing impacts. As an initial step toward this direction, we examine one factor in this paper: the link failure location. Link failure constitutes the input signal to the routing system, which is designed to handle such signal by recomputing a new routing path. Previous data measurements [19, 11] seem suggest that the link failure location has impacts on routing convergence delay and overall routing traf£c, but the previous work did not provide a conclusive and clear result. In this paper, we £rst focus on link failures occurring at the core of the network versus those occurring at the edge. The core of the network usually provides transit service to ship data packets from one place to another, while the edge usually forms the source and the destination of the data traf£c. We are interested to know if link failure location is *a* factor or not, *i.e., would the routing impacts, such as the failure-induced routing changes and triggered routing traf£c, be different when a link failure occurs at the core or at the edge?* Second, if the impacts are different, *what would be the differences?*

To discover insights in BGP, current research methodology mainly focused on data measurement and simulations, and very few exploited the power of formal analysis framework. On one hand, formal analysis framework can provide a solid ground of understanding but it may (and often) oversimpli£es the problem, which renders the results impractical. On the other hand, simulation integrates more ingredients so it makes a reasonable approximation to a realistic system. But due to resource limitations, most of simulations only run on small network topologies [1], which only produces limited and sometimes biased results. In addition, simulation results are generally tied with the topologies simulated. Comparatively, formal analysis can be used to study arbitrary size of network and provide general results. In this paper, we explore the possibility to combine two methods together to produce both solid and meaningful results.

We build a formal analysis framework starting from a basic model which models network topology, routing protocol, policies and network growth pattern. This basic model is built with an effort to simplify the problem as well as to include essential routing ingredients. Based on this model, we formally analyze the routing impacts caused by link failures at different locations, examine dominant factors and provide the insightful analysis. Then we move one step further toward a more realistic model by taking special considerations on core nodes into account. We re-examine some results based on the evolved model.

*This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No DABT63-00-C-1027 and by National Science Fundation(NSF) under Contract No ANI-0221453. Any opinions, £ndings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily re¤ect the views of the DARPA or NSF.

† [1]Xiaoliang Zhao, Beichuan Zhang and Daniel Massey are with USC/ISI. E-mail: {xzhao,bzhang,masseyd}@isi.edu [2]Andreas Terzis is with JHU CSD. Email: terzis@cs.jhu.edu [3]Lixia Zhang is with UCLA CSD. E-mail: lixia@cs.ucla.edu

[1]We are aware of possibilities to run simulations on a much large network topology offered by MicroGrid [12] and other simulation tools. However, it is still a very time consuming and resource demanding method.

The following is a brief summary of our major results for large networks, which have more than 10^4 nodes.

- More nodes tends to be affected due to an edge link failure than a core link failure. By "affected node", we refer to a node which changes its route due to a link failure.
- More routes, *i.e., routing entries*, tends to be affected due to a core link failure than an edge link failure, especially when the network has a small core.
- Both edge link failures and core link failures equally contribute the same percentage of total updates.
- When considering core nodes originates multiple pre£xes (M_c factor) and core nodes are connected via multiple links (K factor), M_c factor tends to increase the number of affected routes and trigger more updates due to core link failures, and K factor ampli£es the routing overhead due to edge link failures, while reduce the routing overhead due to core link failures.
- The number of edge nodes and edge links becomes the dominant factors on total updates in very large networks, for both edge link failures and core link failures.

This paper is organized as three parts: the £rst part will study a basic model which simpli£es the problem by making several assumptions. In this part, Section 3 de£nes the basic network model, the routing protocol, the metrics used to quantify the impacts and four assumptions used to simplify the problem. Section 4 formally analyzes the impacts of core link failures and edge link failures, respectively. In Section 5, We verify our analytical results and examine the trend in larger network. In the second part, we relax two assumptions made on core nodes and examine the effects of those assumptions. At last, Section 9 concludes the paper.

2 Related Work

There have been several studies of BGP routing dynamics. Labovitz et al [9] found several pathological behaviors by examining BGP update logs and analyzed the possible origins of such behavior in [10]. Other studies examined the BGP dynamics during stressful events such as worm attacks. [3] reported a correlation between the surge of BGP traf£c and worm activities and [17], Wang et al showed the worm impacted some edge networks, which could be possible causes for BGP dynamics surge. Another study [18] also showed worm attack affected some edge networks like Department of Defense (DoD) networks. Our study uses a formal approach to compare the different dynamics caused by edge and core link failures and represents a £rst step toward a better understanding the impact of failure location on route dynamics.

PART I: A BASIC NETWORK MODEL

3 Network Model

3.1 Topology

To study the impact that the location of failed links has on routing dynamics, we need a network topology model. Our motivation is better understand the different routing dynamics triggered by *core* and *edge* links in the Internet. While a number of interesting results on the properties of the Internet AS graph have been recently published (e.g. [4]), creating a "representative" Internet topology is still an open research question. A number of studies [16, 5] have shown that the Internet topology has a hierarchical structure, where a small number of networks, mainly large ISPs, form a core at the root of the hierarchy providing transit service, while the majority of networks form the network edge at different lower levels. Core networks have rich connectivity among themselves, while edge networks have a limited number of connections and rely on higher level networks for connectivity. Our topology model captures these fundamental characteristics by using a clique as the core and having clearly de£ned edges. We do not take into account other characteristics of the Internet topology to keep analysis tractable.

Each node in our network topology belongs either to the core or the edge. All the core nodes, V_c, are connected by a full-mesh. Edge nodes, V_e, connect only to core nodes. Formally, our model can be de£ned as an undirected graph $G(V, E)$ satisfying the following conditions:

- $V = V_c \bigcup V_e, V_c \cap V_e = \emptyset$

- $(u, v) \in E_c, \forall u, v \in V_c$

Figure 1(a) shows a complying topology, in which core nodes are represented as squares while edge nodes are represented as circles. On the other hand, Figure 1(b), shows a non-compliant topology since there are direct links between edge nodes (e.g. (u, v) and (u, w)).

3.1.1 Routing

Our goal is to understand the in¤uential factors of BGP [14] since it is the de-facto inter-domain routing protocol used in the Internet today. For our simulations we use the BGP package from the SSFNet simulator [15]. For our analysis, we use the Shortest Path Vector Protocol (SPVP) [6], which is a simpli£ed version of BGP. SPVP can be summarized as follows:

- When a node u has multiple paths to node v, u chooses the shortest length path as its best route to v. If there is a tie, the path whose next hop node ID is smaller is preferred.
- When the best route changes, u will send a routing update to all its neighbors.

In our model, every AS originates its own pre£x and advertises this pre£x to all of its neighbors. With SPVP, every node constructs a routing table containing the best routes to each pre£x. Since core nodes form a clique, the best route between any two core nodes is one hop while the best route between any two edge nodes is at most three hops (i.e., edge-core-core-edge). Moreover, we have the following property inherited from the de£nition of our routing protocol.

Theorem 3.1. *Paths are symmetric: if a path p was used by u to reach v, then v use p to reach u as well.*

Proof. Since we don't have any path length longer than 3, we will examine each case with path length equals to 1, 2 and 3. When path length equals to 1, it means u and v are directly connected, so u and v use the same path only containing the link (u, v) to reach each other. When the path length equals to 2, if u uses $p = (u, a, v)$ to reach v, while v uses $p' = (v, b, u)$ to reach u, it is easy to have a contradiction of $ID(a) > ID(b)$ and $ID(a) < ID(b)$ where ID is a function to return the ID number of a node. If the path length equals to 3, we will show a contradiction if u and v use different paths. Assume u uses $p = (u, a, b, v)$ while v uses $p' = (v, c, d, u)$ and without loss of generality, assume $b \neq c$. Based on our topology de£nition, a, b, c, d are core nodes and connected with each other. So a connects to both b and c, and since a chooses b to reach v, we have $ID(b) < ID(c)$. Similarly, since v choose c over b to reach u, we have $ID(c) < ID(b)$. Then a contradiction. □

We also apply a simple but widely adopted routing policy:

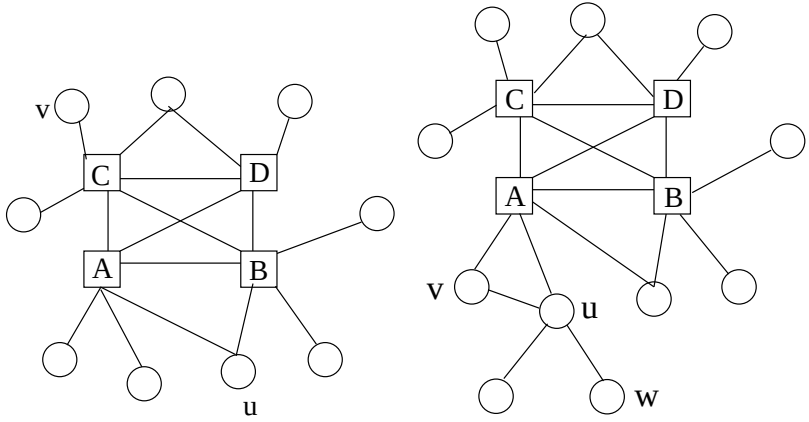
- Core nodes provide transit service only to its customers, i.e., a core node only announces its own and its customers pre£xes to its peers.
- Edge nodes do not provide transit service, i.e., edge nodes do not send updates regarding other nodes' pre£xes.

This policy implies that an edge node will only appear as path source or destination and that edge nodes do not send routing updates in case of link failures.

3.2 Metrics

When a link fails, nodes that were using the disabled path will recalculate their best paths and will subsequently send their our routing updates with the new path information. To understand the different impacts of the location of a link failure, we take a three-step approach. First, we are interested to know *who need to make routing changes due to the link failure*, which is measured by the metric of *number of affected nodes*. This metric also re¤ects the scope of the impact. Second, we are interested to know *what need to be changed*, which is measured by the metric of *number of affected routes*. It counts how many best-route entries have been affected, and re¤ects the potential impact on data forwarding. Finally, we are interested to know *how many information needed to convey changes*, which is measured by the metric of *number of triggered updates*. It counts how many routing updates sent, which captures the total routing overhead caused by the failure.

If each edge link failure triggers t_e updates on the average and there L_e edge links in total, each failing with probability p_e, then the total number of updates due to edge links failures is $p_e t_e L e$. Similarly, the total number of updates due to core link failures is $p_c t_c L_c$. Let $U^e = t_e L_e$, $U^c = t_c L_c$, $\eta_u = \frac{U^c}{U^e}$ and $\eta_p = \frac{p_c}{p_e}$. The percentage of updates due to edge link failures, γ_e, is then:



(a) Complying topology (b) Non-complying topology

Figure 1. Complying and non-complying Topologies

$$\gamma_e = \frac{p_e U^e}{p_e U^e + p_c U^c} = \frac{1}{1 + \eta_p \eta_u}$$

3.3 Simplifying Assumptions

In addition to the topology constraint, we make three additional assumptions in this part of the work. We will address these assumptions in next part of the paper.

First, our analysis assumes that edge and core links have an equal probability of failure. In practice, it is generally believed that Internet core links tend to be more reliable than edge links. Internet core links correspond to the connection between large ISPs who have resources dedicated to monitoring and management.

Second, we assume each node announces a single destination. In practice, both core and edge nodes (i.e. an Autonomous System) can originate routes to several destinations (i.e. pre£xes). Intuitively, an company (edge node) is allocated a relatively small address block and may originate routes to a few pre£xes, but a large ISP (core node) is allocated large address blocks and may originate routes for a large numbers of pre£xes.

Third, we assume core nodes are connected via one single link. In reality, two core nodes (corresponding to large ISPs) are often connected at more than one locations, which results in multiple physical links between them.

Fourth, we do not consider the impact of slow convergence when counting the number of updates. After a link failure, BGP may explore several transient paths before converging on the new stable path [8, 13, 1]. In general topologies, studies [19] have shown that edge link failures tend to create more transient path exploration and thus more updates than core link failures. However, the network topology model and policy we used in this work has the effect of naturally limiting the slow convergence.

Slow convergence occurs mainly because a node is exploring a path which is already "invalid", *i.e.,* a path including the failed link [13]. In our model, if a core link (u, v) fails, because the policy does not permit any other nodes to provide transit service for u to reach v, u will have no any alternative paths to explore. For edge link failures, for example, when the edge link (u, v) fails where $u \in V_e, v \in V_c$, there are two cases. If u is single-homed, clearly v will not have any alternative paths to reach u. If u is multi-homed, for example, u connects to both v and w, v and w are core nodes thus connected with each other. According to the policy, w provides transit service for u to any nodes, including u. Therefore v will have a valid alternative path, which is also the second shortest one for v, to use after (u, v) fails.

4.1 Overview

BGP is a large-scale distributed system. Formal analysis on realistic BGP is dif£cult, if not impossible. Current research is focusing on simulation techniques to approximate to real BGP as much as possible. Our approach is building formal analysis on the top of the simulation, and leave the problem of how we can be closer to real BGP to simulation community. By this two level approximation, we can provide both a solid ground of understanding BGP and reasonable approximation of real BGP. (but we should note that some analytical results such as affected nodes/paths are beyond simulation, is it?)

The failure of link (u, v) affects nodes that use this link. As a result, these nodes will change their routing tables and send their own routing updates. Let $r(u, v)$ be the set of nodes who use u to reach v, including node u. When link (u, v) fails, nodes at both sides of the link are affected, thus the total number of affected nodes is $a(u, v) = r(u, v) + r(v, u)$. Due to symmetry, each affected node at u's side has $r(v, u)$ routing table entries affected and each node at v's side has $r(u, v)$ routing table entries affected, therefore the number of affected paths is $p = 2 * r(u, v) * r(v, u)$. Since edge nodes do not provide transit service for other nodes, they do not send routing updates after a link failure. Affected core nodes will send one update for each affected routing table entry to all their neighbors except the failed link, e.g., u will send $r(v, u) * (d(u) - 1)$ updates, where $d(u)$ is u's degree.

However, even located in the same region, such as in the core, different link failure still has different impact. Therefore, it makes no sense to compare individual link failure instances when we try to factor out the location's impact. Thus, we are trying to answer the following questions which make more senses. *For core link failures or edge link failures, what will be the expected value of the measured metrics?* If core link failures is expected to affect more nodes and paths then edge link failures in a statistical sense, even a particular instance of core link failure may affect few nodes, we will claim that core link failure has more impacts than edge link failure.

In the subsections that follow, we obtain the metrics for each individual link failure as described above, sum over all target links (e.g., all core links) and compute the average metric per category by dividing this sum by the number of links in that category.

4.2 Edge Link Failure

For an edge link $(u, v|u \in V_e, v \in V_c)$, the number of affected nodes $r(u, v) = 1$, which is node u itself.

4.2.1 Number of Affected Nodes

Theorem 4.1. *The average number of nodes affected by an edge link failure is*

$$E(|V^e|) = 1 + \frac{N_e}{L_e}(N - 1)$$

Where N is the total number of network nodes. The proof of this Theorem along with all the other proofs are in the Appendix.

When all edge nodes are single-homed, in which case $N_e = L_e$, any edge link failure will partition an edge node from the network and affect all nodes and $E(|V^e|) = N$, as expected. As edge nodes increase their connectivity L_e increases, reducing the average number of nodes affected by an edge link failure. From this discussion it is easy to see that $\frac{N_e}{L_e}$ re¤ects the improvement in stability due to multi-homing in terms of the scope of an edge link failure.

4.2.2 Number of Affected Paths

Theorem 4.2. *The average number of paths affected by an edge link failure is*

$$E(|P^e|) = 2\frac{N_e}{L_e}(N - 1)$$

Again, multi-homing by increasing L_e can reduce the average number of affected paths due to an edge failure.

4.2.3 Number of Triggered Updates

Theorem 4.3. *The number of updates triggered by all edge links is*

$$U^e = L_e(N - 2)$$

From this result, we can see that as L_e increases, the total number of updates U^e will increase even the number of nodes is unchanged. It implies that multi-homing will increase the total number of updates triggered by edge link failures.

4.3.1 Number of Affected Nodes

Theorem 4.4. *The average number of nodes affected by a core link failure is*

$$E(|V^c|) = \frac{1}{N_c(N_c-1)}(2N(N_c-1) - \sum_{e \in V_e}(d(e) * (d(e)-1)))$$

When every edge node w is single-homed, $d(w) = 1$, and $\sum_{w \in V_e}(d(w) * (d(w)-1))) = 0$, which gives $E(|V^c|) = 2N/N_c$. When multi-homing is used, edge nodes depend less on individual core links, therefore the number of affected nodes decreases. Unlike the case of edge link failure, the quantitative effect of multi-homing cannot be decided by L_e alone, since it also relies on how the edge links are distributed among edge nodes, as measured by $\sum_{w \in V_e}(d(w) * (d(w)-1)))$.

4.3.2 Number of Affected Paths

Theorem 4.5. *The average number of paths affected by a core link failure is*

$$E(|P^c|) = \frac{2}{N_c(N_c-1)}(N(N-1) - 2L_e - \beta N_e(N_e-1))$$

β is a topology-dependent factor, which measures how much the end-to-end connectivity between edge nodes is independent from core links. For example, if all edge nodes connect to the same core node, none of the paths between two edge nodes need to traverse a core link, thus $\beta = 1$, and we have least number of affected paths per core link failure.

4.3.3 Number of Triggered Updates

Theorem 4.6. *The number of updates triggered by all core link failures is*

$$U^c = (N-1)L_e - \sum_{u \in V_c} d_e(u)^2$$

5 Simulation

Our analysis in previous section is based on a much simpli£ed version of BGP. We use SSFNET [15] package to simulate full version of BGP to con£rm our analytical results. Due to resource limitation, simulation is not practical on very large network topologies. However, after con£rming that our analytical results match simulations in small topologies, we can use the theorems to calculate metrics for very large topologies, and study the impact of link failure location in large networks.

5.1 Simulation Settings

In all our simulations, BGP parameters are set to default values, e.g., 30 seconds with random jitter for the $MRAI$ timer. Link delay is set to be 2 milliseconds and processing delay of each routing message is ignored.

After the simulation initialization phase, a link is brought down, which means the BGP session on this link is terminated. This link failure will triggers some routing events in the network. After the network converges again, the metrics are collected. This simulation run is repeated for every link in the network, and £nally the average number of affected nodes, the average number of affected routes and the total number of triggered updates are calculated.

5.2 Topology Generation

We use network topologies with different sizes to study how the network growth affects the results. We pick a small number s as a seed and generate a sequence of topologies with $N_c = s * i$ and $N_e = s * 2^{i-1}$, $(1 \leq i \leq n)$, which means the number of core nodes grows linearly, while the number of edge nodes grows exponentially. For results presented in this paper, we set $s = 6$ and $i \in [1, 5]$. It is observed that since core nodes in the Internet are usually large ISPs providing national or international transit services, their number increase very slowly because of extremely high cost. On the other hand, edge networks can be added with a much smaller cost. Studies [2, 7] show that the growth of edge networks is the major contributor to the Internet growth, and is much faster than that of core networks. Our topology generation tries to capture the difference between the growth rate of core nodes and edge nodes. In each topology, core nodes form a clique,

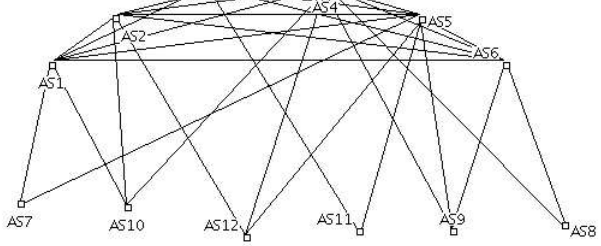
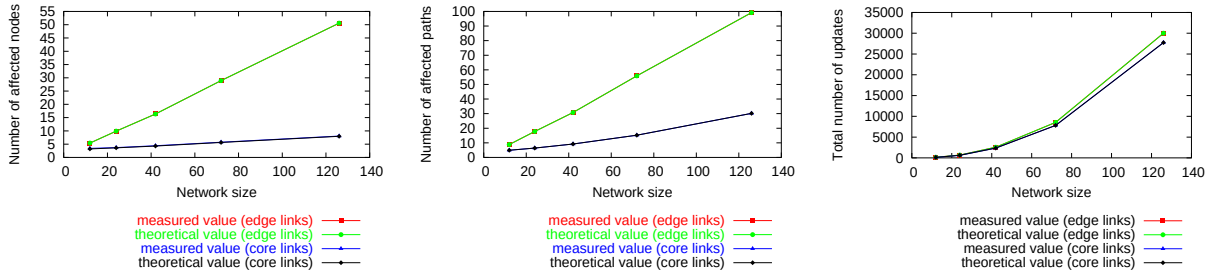


Figure 2. An example network topology used in simulations



(a) Average Affected Nodes (b) Average Affected Paths (c) Total Updates

Figure 3. Simulation results to verify the analysis.

and edge nodes are randomly attached to core nodes. To simulate the limited connectivity of edge nodes, and the increasing trend of multi-homing in the Internet, we limit each edge node to degree of 2 or 3. An example topology used in simulations is shown in Figure 5.2. Note, this topology generation is only used for simulation, and our analytical results do not depend on this.

5.3 Veri£cation of Analysis

To verify analytical results in previous section, we compare the metrics values obtained from simulation and calculation. Figure 3 shows that for all three metrics and both edge link failure and core link failure, simulation results and theoretical results match each other almost exactly, which means our routing model and analysis did capture BGP behavior in the topologies being examined.

In calculating the average affected routes for core link failure, β is obtained in the following ways. For an edge node u, it connects to multiple core nodes $C = \{c_1, c_2, \ldots, c_m\}$, and each c_i connects a set of edge nodes, so u can reach those nodes via c_i and form a path with a pattern edge-core-edge. Therefore we have

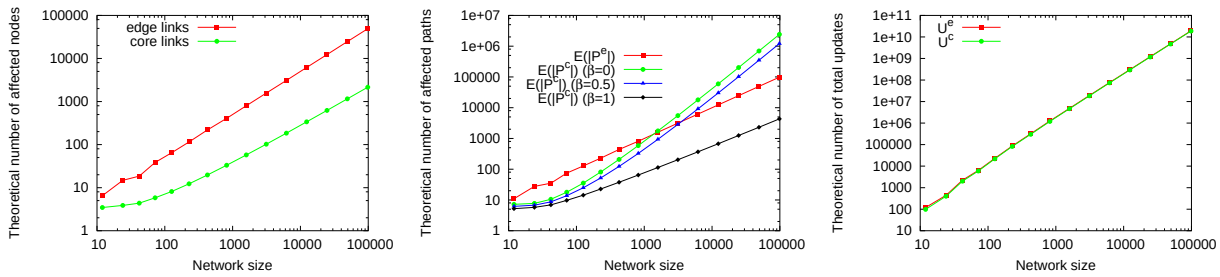
$$\beta = \frac{1}{N_e(N_e - 1)} \sum_{e \in V_e} | \bigcup_{c \in nbr(e)} nbr_e(c)|$$

Where $nbr(u)$ is de£ned as the set of neighboring nodes of u, $nbr_e(u)$ is de£ned as the set of neighboring edge nodes of u.

5.4 Results for large networks

Since simulation is practical only on relatively small topologies, to evaluate link failures in large networks, we have to rely on calculation based on our theorems. We generate a sequence of network topologies with linear increase of core nodes and exponential increase of edge nodes, and then calculate the values for all three metrics and both edge link failure and core link failure. The results are shown in Figure 4.

Observation 1. *Figure 4(a) shows that edge link failure affects more nodes than core link failure.*



(a) Average Affected Nodes (b) Average Affected Routes (c) Total Updates

Figure 4. Very Large Networks (calculation)

This implies that on average, an edge link failure affects larger scope than a core link failure. This is the result of edge nodes' limited connections compared with core nodes. For an edge node with a small number of links, all the other nodes in the network need to use this small number of links to reach the said node. Therefore, on average, there are a large number of nodes rely on an edge link. When the edge link fails, a large number of nodes will be affected. On the contrary, core nodes usually have much more links, thus there are less nodes rely on each individual core link, and its failure will have a smaller scope.

Observation 2. *Figure 4(b) shows that on average, an edge link failure affects more routes in small network size, but less routes in large network size, compared with core link failure.*

Since edge nodes grow much faster than core nodes, in large networks, there are lots of edge nodes and a relatively small core, which means a very large number of routes rely on a relatively small number of core links. This increases the average number of routes affected by core link failure, and becomes much more pronounced when the network is very large.

Observation 3. *Figure 4(c) shows that in terms of total routing updates, edge link failures have almost the same impact as the core link failures.*

The total updates generated by edge link failures and core link failures are obtained by the following two equations.

$$U^e = L_e(N-2) = L_e(N-1) - L_e = L_e(N-1) - \sum_{u \in V_c} d_e(u)$$

$$U^c = L_e(N-1) - \sum_{u \in V_c} d_e^2(u)$$

It is easy to see that the same terms $L_e(N-1)$ dominates both U^c and U^e, which explains our results in Figure 4(c). The only difference is the term being subtracted, which is determined by the number of edge nodes each core node connects to. When core nodes tend to connect to more edge nodes, U^c will become smaller than U^e.

The intuition behind the equations is the following. Updates are generated to convey routing information in network events. There are two factors determines how many updates are generated in the network. One is how many nodes generate updates, which is the number of affected nodes. The other is how many updates each node generates, which is the number of affected routes. Edge link failures affect more nodes (Figure 4(a)), but each affected node only generate updates about one pre£x, the edge pre£x being affected. On the other hand, core link failures affect fewer nodes, but each affected node generate updates about many pre£xes because the number of affected routes is large (Figure 3(a)), especially when edge nodes grows much faster than core nodes. When counting the total updates, The combination of these two factors seem to cause equally amount of updates for their respective cases.

6 Link Failure Probability

Previously, we assume a core link and an edge link share the same probability to fail. Since

$$\gamma_e = \frac{1}{1 + \eta_p \eta_u}$$

practice, because large ISPs normally have more monetary and human resources to maintain links, so it is reasonable say $\eta_p \ll 1$, which means core links are much less likely to fail than edge links. In this case, γ_e will approach to 1 which means almost all of routing dynamics will be attributed to edge link failures.

PART II: SPECIAL CONSIDERATIONS TO CORE NODES

In this part, we re-examine assumptions made on core nodes. Previously, we assume each core node only originates one pre£x and two core nodes are connected via one single link. Those assumptions are not realistic. We relax those assumptions and redo the analysis. At the meantime, we analyze the differences those assumptions made in the results.

7 Multiple Pre£xes Originated by Core Nodes

In practice, an company (edge node) is allocated a relatively small address block and may originate routes to a few pre£xes, but a large ISP (core node) is allocated large address blocks and may originate routes for a large numbers of pre£xes. To re¤ect this difference, we simply assume each core node originates M_c pre£xes.

In BGP, each path is associated with a pre£x regardless of location the pre£x is attached. Therefore, we count each path as per pre£x based. Note this assumption does not make any differences on the results for the affected nodes and total updates generated by edge link failures.

7.1 Edge Link Failure

Theorem 7.1. *The average number of paths affected by an edge link failure is*

$$E(|P^e|) = \frac{1}{L_e}(N_e N_c M_c + N_e(N + N_e - 2))$$

7.2 Core Link Failure

Theorem 7.2. *The average number of paths affected by a core link failure is*

$$E(|P^c|) = \frac{2}{N_c(N_c-1)}((N-1)(N_c M_c + N_e) - L_e(M_c+1) - \beta N_e(N_e-1))$$

Theorem 7.3. *The number of updates triggered by all core link failures is*

$$U^c = (N_c - 1)L_e M_c + N_e L_e - \sum_{u \in V_c} d_e^2(u)$$

7.3 Simulation Results and Analysis

Again, we verify our results by simulation and examine the trend in larger networks. Figure 5 veri£es our new analytical results, and the calculated results are shown in Figure 6, from which, we made the following observations.

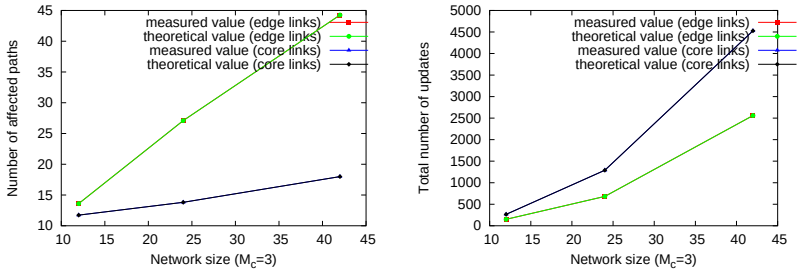
Observation 4. *When $m_c > 1$, core link failures tends get more paths affected and generate more updates than edge link failures.*

When core nodes originate larger number of pre£xes than edge nodes, which is the case in the current Internet, core links carry more pre£xes than edge links. Therefore, when core link fails, more paths will be affected since each path is de£ned at per pre£x base. Consequentially, more updates will be generated.

8 Multiple Links between Core Nodes

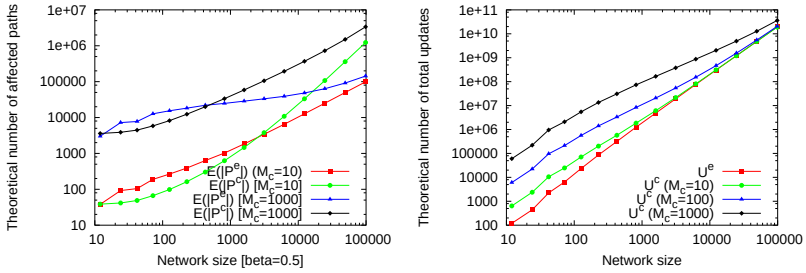
In reality, core nodes are connected at different places. Thus, two core nodes are not connected by one single link, but rather by multiple links. In this section, we reform our model and create multiple links between core nodes. For simplicity, we assume there are K links between every core nodes pair.

However, when one of the core links fails, there are two different cases to reconstruct the routes. We use an example, shown in Figure 7, to illustrate this point. Core node A is further remodeled to include two subnodes a and b, denoted by



(a) Average Affected Paths (b) Total Updates

Figure 5. Simulation results to verify the analysis.



(a) Average Affected Routes (b) Total Updates

Figure 6. Very Large Networks (calculation). Because U^e calculation is not affected by M_c, so we only show one calculation results.

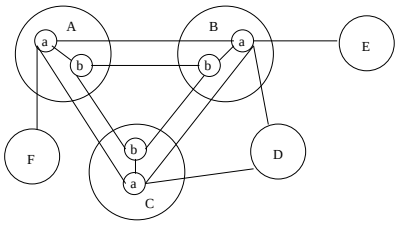


Figure 7. An example of multiple links between core nodes

be two links between A and B. Now, $A.a$ has two paths to reach B, one is via the link $(A.a, B.a)$, and the other via the link $(A.a, A.b)$ £rst, then going through the link $(A.b, B.b)$. We de£ne an *external link* as a link between two subnodes belonging to two different core nodes, while an *internal link* is de£ned as a link between two subnodes within the same core node.

When the link $(A.a, B.a)$ fails, $A.a$ has to switch to the internal link $A.a, A.b$ to reach B because there are no alternatives. Same thing for $A.a$ to reach node E. Such change will be considered as a local change because no "new" routing information is produced. Ideally, such local changes should not be propagated to the global routing system. So in such cases, there will be no nodes except u being affected, and no path changes and no updates.

However, $A.a$ will take a different approach to reconstruct the path to reach D. $A.a$ has two choices after $(A.a, B.a)$ fails. One is going through internal link $(A, a, A.b)$ £rst, then through $(B.b, B.a)$ to reach D. The other one is going through an external link $(A.a, C.a)$ £rst, then through the link $(C.a, D.a)$ to reach D. At node level, these two paths have same path length. To break the tie, in reality, a node (or an AS in BGP term) will normally try to direct traf£c out of its own network via nearest exit for the economic and monetary reasons. In our example, $A.a$ will use the external link $(A.a, C.a)$ to reach D. We document such practice explicitly as the following policy.

- When a node u has multiple paths via both external links and internal links to reach the same pre£x, u will always prefer the external links over internal links. [2]

8.1 Core Link Failures

First, we note that multiple links between core nodes will not affect the results for edge link failures, so here we only discuss the new results for core links failures.

Theorem 8.1. *When $K > 1$, the average number of nodes affected by a core link failure is*

$$E(|V^c|) = 2 + \frac{2}{N_c(N_c-1)K}(\alpha\rho(N_e-1)N_e^{''})$$

where $N_e^{''}$ is the number of multi-homed edge nodes, ρ is the percentage of paths which fall in edge-core-core-edge category and destine to multi-homed edge nodes, while α is the percentage of unique pre£xes edge-core-core- of such edge-core-core-edge paths.

The above theorem implies increasing the connectivity between core nodes will decrease the average number of affected nodes caused by core link failures. It is because the edge nodes' dependency of core links are now shared by K links. To the extreme, when $K \to \infty$, $E(|V^c|) \to 2$, which means only the two core nodes are affected in such extreme cases.

ρ is a topological factor determined by the way how the edge nodes connecting to core nodes, similar to β as we discussed before. α is also a topological factor determined by the routing preference of edge nodes. Given the same number of paths between edge nodes, if each edge node prefers to route the traf£c via the same core link to different destinations, α tends to be small. The average number affected node tends to decrease because the node will only be affected once for a particular core link failure.

Theorem 8.2. *When $K > 1$, the average number of paths affected by a core link failure is:*

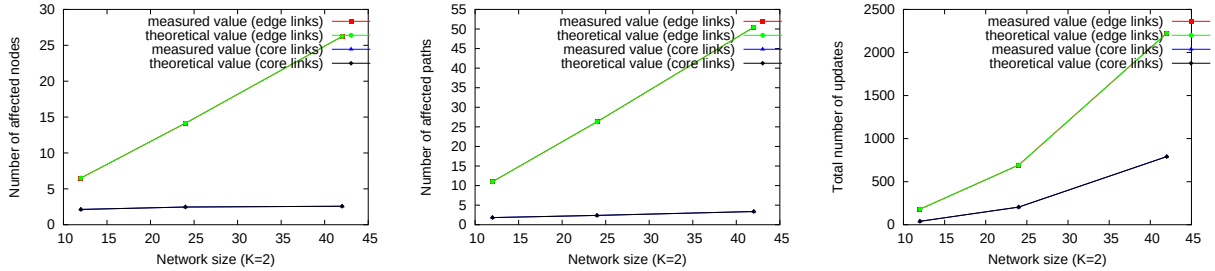
$$E(|P^c|) = \frac{2}{N_c(N_c-1)K}(K(N_eN_c - L_e) + \rho(N_e-1)N_e^{''})$$

When $K \to \infty$, $E(|P^v|) \to (2M_c + \frac{2\rho(N_e-1)N_e^{''}}{N_c(N_c-1)K})$. It implies that when core nodes increase the redundancy between them, the average number of affected paths will become independent to K. It is because large K decrease the dependency of edge nodes on core links, as shown in our proof. The paths being affected are those between two core nodes and in the class of *core-core-edge*, which is independent to K but bound to policy.

Theorem 8.3. *When $K > 1$, the number of updates triggered by all edge link failures is:*

$$U^e = (N_c-1)L_eK + N_eL_e - L_e$$

[2] In practice, it is also phrased as "prefer a path learned from an eBGP peer over the one learned from an iBGP peer".



(a) Average Affected Nodes (b) Average Affected Paths (c) Total Updates

Figure 8. Simulation results to verify the analysis

.

Theorem 8.4. *When $K > 1$, the number of updates triggered by all core link failures is:*

$$U^c = N_e L_e - \sum_{u \in V_c} d_e^2(u)$$

It implies that the number of updates caused by core link failures will not be affected by the number of links between two core nodes. U^c is merely a result determined by topological factors. (We veri£ed this by changing K for a 6-0-6 network, need more simulations.)

8.2 Simulation Results and Analysis

Figure 8 veri£es the analytical results, where α was calculated as the following: for every pair of edge nodes $(u_e, v_e)(u_e \neq v_e)$, u_e and v_e can reach each other either by a *edge-core-edge* path or a *edge-core-core-edge* path. Using the same method to compute β, we can eliminate those pairs with *edge-core-edge* path. For the rest of edge node pairs, we assign each pair with a "smallest" core link which connects two edge nodes. A core link (u_{c1}, v_{c1}) is "smaller" than (u_{c2}, v_{c2}) when $(u_{c1} < u_{c2})$ or $(u_{c1} = u_{c2}$ and $v_{c1} < v_{c2})$. After assignment, we can easily get the unique pre£xes of *edge-core-core-edge* paths, then calculate α.

Observation 5. *When $K > 1$, edge link failures tends to affects more nodes and paths than core link failures.*

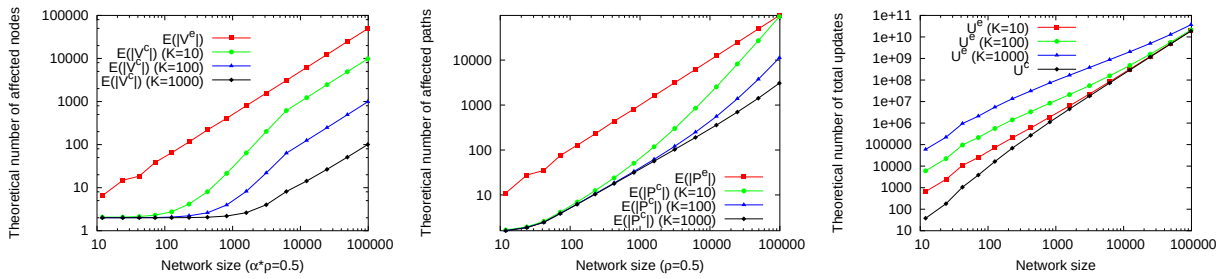
When $K > 1$, we have shown cases that core link failure will not affect some nodes and paths, which turns to be even more true for large networks.

Observation 6. *When $K > 1$, edge link failures contribute much more updates than core link failures.*

The dominant term of U^e is $(N_c - 1)L_e K + N_e L_e$. K plays a role in U^e because the information about edge link failure will be propagated through all K links between affected core nodes. For U^c, because multiple links keep core nodes connected, it cancels updates due to path changes between two core nodes. Therefore, K factor ampli£es the routing overhead when edge link fails, while reduces the routing overhead when core link fails.

9 Conclusion

The ultimate goal of this paper is to gain deep understanding of routing instability and corresponding in¤uential factors. In this paper, we examine one single factor, the link failure location, and its impact on routing instability. We £rst de£ne a simple network model, which differentiate link locations into two kinds, links at the core and links at the edge. The impacts are de£ned by three metrics: the average number of affected nodes, the average number of affected paths and the number of updates. Focusing on our network model, we conduct a formal analysis on the impacts of link failures occurring at the core versus at the edge. The analytical results are veri£ed by simulations. Then we relax some of our assumptions and re-examine the results. We found edge link failures and core link failures have different type of routing impacts. Also, we found the dominant factor which determines the number of update when the network becomes very large. This paper



(a) Average Affected Nodes (b) Average Affected Routes (c) Total Updates

Figure 9. Very Large Networks (calculation).

also serves as an example of how to use analysis approach, combined with simulation validation, to gain deep insights of a large-scale system.

However, due to our de£nition of topology and policy, slow convergence, a syndrome for real Internet, is naturally limited in our analysis. We plan to study the impacts of slow convergence in our future work. Moreover, network growth model used in this paper is quite simple. We plan to examine different network growth models later, such as power law growth model.

Appendix

Theorem 4.1

Proof. The failure of link $(u, v|u \in V_e, v \in V_c)$ affects $a(u, v)$ number of nodes, $a(u, v) = 1 + r(v, u)$. If we take the sum $a(u, v)$ over all u's links and all edge nodes, then divide it by the total number of edge links, we get

$$E(|V^e|) = \frac{1}{L_e} \sum_{u \in V_e} \sum_{(u,v) \in E} a(u, v) = \frac{1}{L_e} \sum_{u \in V_e} \sum_{(u,v) \in E} (1 + r(v, u))$$

Since every node except u must use exactly one of u's link to reach u,

$$\sum_{(u,v) \in E} r(v, u) = N - 1, \sum_{(u,v) \in E} 1 = d(u)$$

Therefore,

$$E(|V^e|) = \frac{1}{L_e} \sum_{u \in V_e} (d(u) + N - 1) = \frac{1}{L_e} (\sum_{u \in V_e} d(u) + (\sum_{u \in V_e} (N - 1))$$
$$= \frac{1}{L_e} (L_e + (N - 1)N_e) = 1 + \frac{N_e}{L_e} (N - 1)$$

□

Theorem 4.2

Proof. The failure of link $(u, v|u \in V_e, v \in V_c)$ affects $p(u, v)$ number of nodes, $p(u, v) = 2r(v, u)$. If we sum $p(u, v)$ over all u's links and all edge nodes, then divide the sum by the number of edge links, we have:

$$E(|P^e|) = \frac{1}{L_e} \sum_{u \in V_e} \sum_{(u,v) \in E} p(u, v) = \frac{1}{L_e} \sum_{u \in V_e} \sum_{(u,v) \in E} 2r(v, u) = \frac{2}{L_e} \sum_{u \in V_e} (N - 1) = 2 \frac{N_e}{L_e} (N - 1)$$

□

Proof. When $(u, v|u \in V_e, v \in V_c)$ fails, u does not send updates because it is an edge node, but v will send updates to each of its neighbors except u, thus $d(v) - 1 = (N_c - 1) + d_e(v) - 1$ updates, where $d_e(v)$ is the edge links that v has. Any other core node v' that relies on this link to reach u will also change its routing table and send updates, but only to edge nodes due to the policy, thus $d_e(v')$ updates. Let $t(u, v)$ be the number of updates triggered by this link failure,

$$t(u,v) = N_c - 2 + \sum_{w \in V_c}^{w \in r(v,u)} d_e(w)$$

Now we sum $t(u, v)$ over all links of u.

$$\sum_{(u,v)\in E} t(u,v) = \sum_{(u,v)\in E} (N_c - 2 + \sum_{w \in V_c}^{w \in r(v,u)} d_e(w)) = (N_c - 2)d(u) + \sum_{(u,v)\in E} \sum_{w \in V_c}^{w \in r(v,u)} d_e(w)$$

Since every core node must use exactly one of u's link to reach u, $\sum_{(u,v)\in E} \sum_{w\in V_c}^{w\in r(v,u)} d_e(w)$ covers all cores nodes, thus it equals to $\sum_{v\in V_c} d_e(v) = L_e$.
Next we sum over all edge nodes to get the total number of updates triggered by all edge link failures.

$$U^e = \sum_{u\in V_e} \sum_{(u,v)\in E} t(u,v) = \sum_{u\in V_e} ((N_c - 2)d(u) + L_e) = (N_c - 2)L_e + N_e L_e = (N - 2)L_e$$

□

Theorem 4.4

Proof. When a core link (u, v) fails, it affects $a(u, v)$ number of affected nodes, $a(u, v) = r(u, v) + r(v, u)$. $r(u, v)$ includes u itself and some of u's edge nodes. Let $de(u)$ be the number of edge links that u has, then $r(u, v) \leq (1 + de(u))$, because a multi-homed edge node may not choose u as the next hop to reach v. Based on how multi-homed edge nodes choose their preferred core nodes for transit service, u's edge nodes can be divided into two groups: customers, who prefer u to other core nodes they connect to, and non-customers, who prefer other core nodes to u. A customer node relies on u to reach v, unless it has a direct link to v. Let $mc(u)$ be the number of u's customers, and $nc(u)$ be the number of customers who have direct links to v, since the routing protocol breaks ties by comparing node index,

$$r(u,v) = \begin{cases} 1 + mc(u) & v < u \\ 1 + mc(u) - nc(u) & v > u \end{cases}$$

Summing $a(u, v)$ over all core links, and then dividing it by the number of core links gives us the average number nodes affected by a core link failure:

$$E(|V^c|) = \frac{1}{N_c(N_c - 1)} \sum_{u\in V_c} \sum_{v\in V_c}^{v\neq u} (r(u,v) + r(v,u))$$

Since each edge node will be counted once and only once as a core node's customer, $\sum_{u\in V_c} mc(u) = N_e$. Therefore

$$\sum_{u\in V_c} \sum_{v\in V_c}^{v\neq u} r(u,v) = \sum_{u\in V_c} (\sum_{v\in V_c}^{v\neq u} (1 + mc(u)) - \sum_{v\in V_c}^{v>u} nc(u)) = \sum_{u\in V_c} (1 + mc(u))(N_c - 1) - \sum_{u\in V_c} \sum_{v\in V_c}^{v>u} nc(u))$$

$$= (N_c - 1) \sum_{u\in V_c} (1 + mc(u)) - \sum_{u\in V_c} \sum_{v\in V_c}^{v>u} nc(u)) = (N_c - 1)N - \sum_{u\in V_c} \sum_{v\in V_c}^{v>u} nc(u))$$

The term $\sum_{u\in V_c} \sum_{v\in V_c}^{v>u} nc(u))$ actually counts all cases of (u, e, v) where u and v are core nodes and e is an edge node. Therefore,

$$\sum_{u\in V_c} \sum_{v\in V_c}^{v>u} nc(u)) = \frac{1}{2} \sum_{w\in V_e} (d(w) * (d(w) - 1))$$

$$\sum_{u \in V_c} \sum_{v \in V_c}^{v \neq u} c(u,v) = N(N_c - 1) - \frac{1}{2} \sum_{w \in V_e} (d(w) * (d(w) - 1))$$

Since $\sum_{u \in V_c} \sum_{v \in V_c}^{v \neq u} r(u,v) = \sum_{u \in V_c} \sum_{v \in V_c}^{v \neq u} r(v,u)$, The £nal result is

$$E(|V^c|) = \frac{1}{N_c(N_c - 1)}(2N(N_c - 1) - \sum_{w \in V_e} (d(w) * (d(w) - 1)))$$

□

Theorem 4.5

Proof. When a core link (u, v) fails, the number of affected paths is $2 * r(u,v) * r(v,u)$. However, summing this term over all core links gives a complex formula. Therefore we take another approach to get a more intuitive result. Since there are in total $N(N-1)$ "best" paths in the network, we just need to £nd how many paths are not affected by core link failures and subtract it from $N(N-1)$. Paths that are not affected by core link failures should not contain any core link, which means they belong to one of the following categories: core-edge, edge-core, and edge-core-edge. Paths in the £rst two categories connect an edge node to a core node, therefore their total number is $2L_e$. Paths in the third category connect two edge nodes via a core node. Let β be the percentage of all paths connecting two edge nodes that belong to the third category. The number of paths belonging to the third category is then $\beta N_e(N_e - 1)$. Therefore, we have

$$E(|P^c|) = \frac{2}{N_c(N_c - 1)}(N(N-1) - 2L_e - \beta N_e(N_e - 1))$$

□

Theorem 4.6

Proof. When a core link (u, v) fails, u will change paths to reach v and any node $w \in r(v,u)$ and $w \in V_e$. It is obvious that u and w are not directly connected. For each path change, u will send new path information only to all its edge neighbors, because policy does not allow u to announce a path destined to v and w to any of its core neighbors. Let $p(u,v)$ be the number of paths need to be changed at u's side, $m(u,v)$ be the number of updates sent by u, we have $m(u,v) = p(u,v) * d_e(u)$. Sum $m(u,v)$ for all core links u attached to, u totally need to change $(N-1) - d_e(u)$ paths. Therefore,

$$\sum_{v \in V_c}^{v \neq u} m(u,v) = \sum_{v \in V_c}^{v \neq u} p(u,v) d_e(u) = d_e(u) \sum_{v \in V_c}^{v \neq u} p(u,v) = d_e(u)(N - 1 - d_e(u))$$

Sum over all core nodes, we will have

$$U^c = \sum_{u \in V_c} d_e(u)(N - 1 - d_e(u)) = (N-1)L_e - \sum_{u \in V_c} d_e^2(u)$$

□

Theorem 7.1

Proof. The failure of an edge link $(u, v | u \in V_e, v \in V_c)$ will affect nodes in $r(u,v)$ and $r(v,u)$. First, we look at $r(u,v)$. Because no node will use u to reach v, so $r(u,v) = \{u\}$. According to routing symmetry property, u will change all paths to reach $w \in r(v,u)$. When we cut all edge links connected with u, u will change paths to all $w \neq u$. In such case, it will be $N_c M_c + (N_e - 1)$ path changes.
Then, we look at $r(v,u)$. For any $w \in r(v,u)$, w need to change its paths to u. If we cut all edge links connected to u, all nodes except u will need to change the path once. In such case, we have $(N-1)$ path changes.
Sum above two for all edge node u, we will have:

$$E(|P^e|) = \frac{1}{L_e}(\sum_{u \in V_e} (N_c M_c + (N_e - 1) + (N-1)) = \frac{1}{L_e}(N_e N_c M_c + N_e(N + N_e - 2))$$

□

Proof. Again, we use the similar approach as we used to prove Theorem 4.5. In total, we have $N(N-1)$ best paths, and there are total $(N(N-1)-2L_e-\beta N_e(N_e-1))$ Paths which will use a core link. For those paths, we can classify them into four categories: *core-core*, *edge-core-core*, *core-core-edge*, *edge-core-core-edge.* The number of paths in each category is listed in the following:

category	paths with one pre£x per node	paths with multiple pre£xes
core-core	$N_c(N_c-1)$	$(N_c(N_c-1))M_c$
edge-core-core	$N_eN_c-L_e$	$(N_eN_c-L_e)M_c$
core-core-edge	$N_eN_c-L_e$	$N_eN_c-L_e$
edge-core-core-edge	$(1-\beta)N_e(N_e-1)$	$(1-\beta)N_e(N_e-1)$

Sum all the paths, we will have

$$\begin{aligned} E(|P^c|) &= \frac{2}{N_c(N_c-1)}((N_eN_c-L_e)(M_c+1)+N_c(N_c-1)M_c+(1-\beta)N_e(N_e-1)) \\ &= \frac{2}{N_c(N_c-1)}((N-1)(N_cM_c+N_e)-L_e(M_c+1)-\beta N_e(N_e-1)) \end{aligned}$$

□

Theorem 7.3

Proof. Similar to Theorem 4.6, for core node u, when we fail all core links attached to u, there are $(N-1)-d_e(u)$ path changes when every node only originates one pre£x . Out of them, N_c-1 are paths to reach other core nodes and $N_e-d_e(u)$ are paths to reach edge nodes. Since each core node will originates M_c pre£xes, in this case, we will have $(N_c-1)M_c$ path changes for core nodes. In addition, we will have $N_e-d_e(u)$ path changes for edge nodes. For each path change, u will send $d_e(u)$ updates. So sum the results for all core nodes, we will have
Sum over all core nodes, we will have

$$U^c=\sum_{u\in V_c} d_e(u)((N_c-1)M_c+N_e-d_e(u))=(N_c-1)L_eM_c+N_eL_e-\sum_{u\in V_c} d_e^2(u)$$

□

Theorem 8.1

Proof. When one of the links between core node u and v fails, u and v will £nd another link between themselves and use that to reach each other, given no any other node provides alternative paths for u and v to use because of policy. Therefore, for each core link failure, we have at least two nodes affected.
Moreover, depending on the degree of destination edge node, node u may change paths reaching a remote edge node. For example, for single-homed edge nodes like E, when the link $(A.a, B.a)$ fails, A has no choice but using its internal link to reach E again, which is considered as a local change, and such local change will not affect any other nodes because there are no new path information being provided. Therefore, paths in category *edge-core-core-edge* but destining to a single-homed edge node will not be affected by core link failures.
When the same link fails, to reach a multi-homed destination, such as D in our example, A will choose a new path to reach D due to the policy. Such path change in turn will affect some edge nodes which depend on A to reach D. In this case, more nodes will be affected. As a matter fact, when the destination edge node is multi-homed, because the policy always prefers an external link; and such external link always is available for a core node because of our clique topology, the core node will chose a different path and such path change in turn may affect some edge nodes. It means *edge-core-core-edge* paths destining to multi-homed edge nodes will be affected. For convenience, we term such paths as $ECCE_2$ paths.
Assuming $N_e^{''}$ multi-homed edge nodes, there are totally $(N_e-1)N_e^{''}$ paths which destine to multi-homed edge nodes and in either *edge-core-edge* or *edge-core-core-edge* category. Let ρ be the percentage of the path which are in the latter category. As we analyzed above, only edge nodes which has at least one $ECCE_2$ path will be affected. For a single core link failure, if an edge node need to change its paths to multiple destinations, it will be counted as only one affected node. Hence, the total number of affected node would be equal to the number of unique pre£xes, in the form of *edge-core-core-*, of *edge-core-core-edge* paths. We de£ne $\alpha=\frac{\text{number of unique pre£xes of } ECCE_2 \text{ paths}}{\text{number of } ECCE_2 \text{ paths}}$, given we have $\frac{N_c(N_c-1)}{2}K$ core links, then we will have

$$E(|V^c|)=2+\frac{2}{N_c(N_c-1)K}(\alpha\rho(N_e-1)N_e^{''})$$

$$E(|V^c|) \leq 2 + \frac{N_e}{K}$$

Note the previous proof techniques is not applicable to $K > 1$ cases. Previously, we have $a(u, v) = r(u, v) + r(v, u)$, but for $K > 1$, it may not hold any more. For example, assume $w \in r(u, v)$ and $w \neq u$. When $K = 1$, w will be affected when u and v are disconnected. But when $K > 1$, u and v will not be disconnected, so w still can reach v via u. Consequently, there is a chance that w will not be affected. Taking an example shown in Figure 7, after link $(A.a, B.a)$ fails, $A.a$ changed the path from (A, B, D) to (A, C, D) in order to reach D, while D still kept its old path (D, B, A) to reach A since B still can reach A. In this case, A was affected but D was not.

□

Theorem 8.2

Proof. When one of core links between u and v fails, u need to change paths to reach v by failing over to another core link between u and v. Such change are purely local changes, which only affect u and v. For ef£ciency purpose, such local should be "hidden" to global routing system. So we omit those path changes as if they never occurred.
Moreover, u may change paths to reach some multi-homed remote edge nodes. So does for the edge nodes which rely on u to reach the same remote nodes. The example cases are given in Figure 7. When the link $(A.a, B.a)$ fails, A and F need to change the path to reach D. Those paths will fall into the category of *core-core-edge* and *edge-core-core-edge*.

For each path in *core-core-edge* category, we claim it will be affected once and only once when we examine each core link failure one by one. It is because such paths use a core link, which will be eventually failed during our examination. Consequently, the said path will be affected. Since there are K links between two core nodes, the number of *core-core-edge* paths will have a factor of K. It is because any of K links fail, the same *core-core-edge* path will be affected. Using our previous example, to reach D, when the link $(A.a, B.a)$ fails, $A.a$ will change path from (A, B, D) to (A, C, D). When the link $(A.b, B.b)$ fails, $A.b$ also need to change path from (A, B, D) to (A, C, D). Thus the path (A, B, D) was affected twice. Therefore in total we have $K(N_e N_c - L_e)$ paths in *core-core-edge* class.
For paths in *edge-core-core-edge* category, as we stated earlier, only $ECCE_2$ paths will be affected. Then we have totally $K(N_e N_c - L_e) + \rho(N_e - 1)N_e^{''}$ paths be affected, thus

$$E(|P^c|) = \frac{2}{N_c(N_c - 1)K}(K(N_e N_c - L_e) + \rho(N_e - 1)N_e^{''})$$

□

Theorem 8.3

Proof. When $(u, v | u \in V_e, v \in V_c)$ fails, v will send updates to each of its neighbors except u. Now, v has K links to its core node neighbors, we assume the updates from v will go through all K links [3]. Then we will have $K(N_c - 1) + d_e(v) - 1$ updates from v. Any other core node v' that relies on this link to reach u will also change its routing table and send updates, but only to edge nodes due to the policy, thus $d_e(v')$ updates. Let $t(u, v)$ be the number of updates triggered by this link failure,

$$t(u, v) = K(N_c - 1) + \sum_{w \in V_c}^{w \in r(v,u)} d_e(w) - 1$$

Now we sum $t(u, v)$ over all links of u.

$$\sum_{(u,v) \in E} t(u, v) = \sum_{(u,v) \in E} (K(N_c - 1) + \sum_{w \in V_c}^{w \in r(v,u)} d_e(w) - 1) = K(N_c - 1)d(u) + L_e - d(u)$$

Next we sum over all edge nodes to get the total number of updates triggered by all edge link failures.

$$U^e = \sum_{u \in V_e} \sum_{(u,v) \in E} t(u, v) = \sum_{u \in V_e} (K(N_c - 1)d(u) + L_e - d(u)) = (N_c - 1)L_e K + N_e L_e - L_e$$

□

[3] In reality, two large ISPs, *i.e.*, two core nodes, connect with each other via different border routers. Within one ISP, the border routers are using iBGP, which is designed to keep a consistent routing view. So the event of (u, v) failure will be propagated to all iBGP routers. Then those iBGP routers will push updates out to their respective peers in another ISP, if we assume two ISPs only have one consistent policy implemented at all iBGP routers. This will support our assumption of all K links are used to propagate the event of (u, v) failure.

Proof. When one of the K core links between core nodes u and v fails, there are four type of paths, namely, *core-core*, *core-core-edge*, *edge-core-core* and *edge-core-core-edge*, which may need change and trigger the updates. Since the path changes belonging to the £rst category are considered merely local change without global routing impacts, we ignore this category. Moreover, since edge nodes will not send updates, so we will not count the last two categories either. Therefore, the only path changes having global impacts are core nodes change paths to reach remote edge nodes, *i.e.*, *core-core-edge* paths. For those changes, according to policy, u will send updates only to all of its neighboring edge nodes. Therefore, if all core links connecting to u fail, u need to change $N_e - d_e(u)$ paths and sends $d_e(u)$ updates. Sum it for all core nodes, we have

$$\sum_{u \in V_c} ((N_e - d_e(u))d_e(u)) = N_e L_e - \sum_{u \in V_c} d_e^2(u)$$

Add above two results together, we have

$$U^c = N_e L_e - \sum_{u \in V_c} d_e^2(u)$$

□

References

[1] A. Bremler-Barr, Y. Afek, and S. Schwarz. Improved BGP Convergence via Ghost Fluching. In *Proceedings of the IEEE INFOCOM*, April 2003.

[2] T. Bu, L. Gao, and D. Towsley. On Characterizing Routing Table Growth. 2002.

[3] J. Cowie, A. Ogielski, B. J. Premore, and Y. Yuan. Global routing instabilities triggered by Code Red II and Nimda worm attacks. Technical report, Renesys Corporation, Dec 2001.

[4] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationship of Internet Topology. In *Proceedings of ACM SIGCOMM*, 1999.

[5] L. Gao. On inferring automonous system relationships in the internet. *IEEE/ACM Transactions on Networks*, 9(6), 2001.

[6] T. Grif£n and G. Wilfong. A Safe Path Vector Protocol. In *Proceedings of IEEE INFOCOMM*, March 2000.

[7] G. Huston. Analyzing the Internet BGP Routing Table. *The Internet Protocol Journal*, mar 2001.

[8] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proceedings of ACM Sigcomm*, August 2000.

[9] C. Labovitz, G. Malan, and F. Jahanian. Internet Routing Instability. In *Proceedings of ACM Sigcomm*, September 1997.

[10] C. Labovitz, G. Malan, and F. Jahanian. Origins of Internet Routing Instability. 1999.

[11] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang. Analysis of BGP Update Surge during Slammer Worm Attack. In *5th International Workshop on Distributed Computing (IWDC)*, 2003.

[12] MicroGrid: Emulation Tools for Computational Grid Research. http://www-csag.ucsd.edu/projects/grid/microgrid.html.

[13] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, F. S. Wu, and L. Zhang. Improving BGP Convergence Through Assertions Approach. In *Proceedings of the IEEE INFOCOM*, June 2002.

[14] Y. Rekhter and T. Li. Border Gateway Protocol 4. RFC 1771, SRI Network Information Center, July 1995.

[15] The SSFNET Project. http://www.ssfnet.org.

[16] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *Proceedings of the IEEE INFOCOM*, June 2002.

[17] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang. Observation and Analysis of BGP Behavior under Stress. In *Proceedings of the ACM IMW 2002*, October 2002.

[18] X. Zhao, M. Lad, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang. Understanding BGP Behavior through a Study of DoD Pre£xes. In *Proceedings of the IEEE DISCEX III*, 2003.

[19] X. Zhao, D. Pei, D. Massey, and L. Zhang. A Study on Routing Behavior of Latin America Networks. In *IFIP/ACM Latin America Networking Conference*, 2003.