On Understanding Transient Interdomain Routing Failures

Feng Wang, Jian Qiu, *Student Member, IEEE*, Lixin Gao, *Senior Member, IEEE*, and Jia Wang, *Senior Member, IEEE*

***Abstract*—The convergence time of the interdomain routing protocol, BGP, can last as long as 30 minutes. Yet, routing behavior during BGP route convergence is poorly understood. During route convergence, an end-to-end Internet path can experience a transient loss of reachability. We refer to this loss of reachability as *transient routing failure*. Transient routing failures can lead to packet losses, and prolonged packet loss bursts can make the performance of applications such as Voice-over-IP and interactive games unacceptable. In this paper, we study how routing failures can occur in the Internet. With the aid of a formal model that captures transient failures of the interdomain routing protocol, we derive the sufficient conditions that transient routing failures could occur. We further study transient routing failures in typical BGP systems where commonly used routing policies are applied. Network administrators can apply our analysis to improve their network performance and stability.**

***Index Terms*—BGP, border gateway protocol, interdomain routing, transient routing failure.**

I. Introduction

ROUTING protocols as the "control plane" of the Internet play a crucial role in the end-to-end performance of the Internet. Previous studies have shown that degraded end-to-end path performance is correlated with routing dynamics [1]–[6]. Internet routing protocols include interdomain routing protocols and intradomain routing protocols. Routing information between Autonomous Systems (ASes) is exchanged with the interdomain routing protocol, Border Gateway Protocol (BGP), while the routing information within an AS is maintained with the intradomain routing protocols such as IS-IS or OSPF. Studies have shown that intradomain routing can be fine-tuned to achieve convergence time of a few hundred milliseconds [7], [8]. In contrast, BGP convergence can last as long as 30 minutes [1], [9]. Furthermore, BGP routing events can occur very often [4], [10]–[12]. Yet, routing behavior during BGP route convergence is poorly understood. Very little is known about how routing dynamics cause the degraded end-to-end performance, and the impact of topological properties, routing policies, and routing configurations on routing behavior.

Measurements have shown that a significant number of transient forwarding loops occur during route convergence [13], [14]. When a routing loop occurs, packets can be caught in the loop, causing packet loss or packet delay. In addition to transient routing loops, BGP can experience transient loss of reachability during route convergence [15], [16]. BGP is a path vector protocol where every router announces its best path to its neighbors only. This limited route visibility makes it possible for a router to experience transient loss of reachability during the path exploration of the route convergence process. We refer to this transient loss of reachability during route convergence as *transient routing failures*.

Transient routing failures can lead to packet losses. Furthermore, prolonged packet loss bursts can make deploying applications such as voice over IP and interactive games infeasible. Therefore, it is important to understand when these transient routing failures can occur and how long these failures can last. However, analyzing and measuring transient routing failures can be challenging. Existing abstract models for BGP focus on route convergence properties or traffic engineering within an AS [17]–[19]. The BGP system is a distributed system. The occurrence and duration of transient failures depend on the order in which routing updates are propagated, which in turn depends on the timing of various events in the network (e.g., link failures, or network configuration changes). Further, the advertising of routes in routing updates for one prefix is correlated with routing updates for other prefixes since route update rate limiting timers are typically set for each BGP peering session instead of for each prefix.

In this paper, we study transient routing failures during routing change events, such as failover and recovery routing changes. Our findings can help network operators identify network configurations that might lead to transient routing failures and suggest possible configuration changes and alternative mitigation techniques. Our major contributions are summarized as follows.

- In contrast to existing BGP models [17], [18], [20]–[23], we present an abstract model to capture transient failures of BGP, which allows us to scrutinize the detailed interactions between BGP routers and thus be able to identify the conditions that a transient routing failure could occur.
- Based upon the model, we identify the sufficient conditions for the occurrence of transient routing failures.
- We further apply our generalized theorems to specify the sufficient conditions for a typical BGP system, where routing policies conforming to commercial agreements

Manuscript received June 12, 2006; revised December 28, 2006, July 02, 2007, January 02, 2008, and April 08, 2008; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor O. Bonaventure. First published September 09, 2008; current version published June 17, 2009. This work was supported in part by the National Science Foundation under Grant CNS-0626617, Grant CNS-0626618, and Grant CNS-0325868.

F. Wang is with the School of Engineering and Computational Sciences, Liberty University, Lynchburg, VA 24502 USA (e-mail: fwang@liberty.edu).

J. Qiu and L. Gao are with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003 USA (e-mail: jqiu@ecs.umass.edu; lgao@ecs.umass.edu).

J. Wang is with AT&T Labs–Research, Florham Park, NJ 07932 USA (e-mail: jiawang@research.att.com).

Digital Object Identifier 10.1109/TNET.2008.2001952

between ASes and the hierarchical iBGP configurations are deployed.

The rest of this paper is organized as follows. Sections II and III describe our models for investigating transient routing behavior of a BGP system. Sections IV and V present the sufficient conditions for transient failures at control plane and data plane, respectively. We apply our general theories to study the transient failures in a typical BGP system in Section VI. The related work is reviewed in Section VII. Finally, the paper is concluded in Section VIII.

II. Modeling BGP Transient States

In this section, we model transient states of a BGP system during a transition triggered by an event. A BGP system can go through a series of state transitions after the occurrence of a routing event. These events include link failures, BGP session resets, link additions, topological changes, or routing policy changes.

The primary aim of our model is to understand transient routing behavior and formally define transient routing failures. One of the key challenges in modeling the transient behavior is to capture the order in which routing updates are exchanged. Many factors, such as the timing of various events in the network and MRAI timer, can impact the sequences of routing updates. Our model is able to characterize all possible orders that routing updates are advertised along BGP peering sessions. This model greatly simplifies our understanding of transient routing behavior, and describes the system behavior transitions over time. Our model extends other existing frameworks [24]–[26] that capture the long-term stability of BGP.

A. State Transition Graph

In this paper, a BGP system is modeled as a graph $G = (V, E)$, where V is the set of BGP routers and E is the set of peering sessions between routers. Note that routers in the model can have both iBGP and eBGP sessions. Therefore, each BGP router belongs to one AS and an AS can have one or more BGP speakers. In the system, we choose node 0 that originate a destination d. Without loss of generality, we focus on the routing to this destination during routing events.[1] Every node has a routing table that stores routes to the destination. The routes in a routing table are sorted in the descending order of preference for this node.

In order to capture the transient behavior of a BGP system after the occurrence of an event, we introduce a state transition graph that enumerates all possible transient states of the BGP system and the transitions between these transient states. A *state transition graph* is a directed graph (S, T), where S is a set of *states*, and T is a set of *transitions*. A state is comprised of routes stored at every BGP speaker. It is presented as a vector $S = (s_1, s_2, \ldots, s_n)$, where s_i denotes the set of routes stored at router $i, i = 1, \ldots, n$.

For each state, we have a set of routing updates, U, which are going to be triggered and sent along a set of edges in a future time. A subset $T \subseteq U$ can be triggered at any time. We call $T = (t_1, t_2, \ldots, t_k)$, a *trigger set*, where t_i indicates a routing update that will be sent along an edge. Since a routing update is always sent along an edge, we also use this directed edge to represent the routing update being sent along the direction of the edge.

[1]The presence of a super-net route might help maintain the reachability when a failover event strikes the route to a destination. However, we consider one destination at a time in this paper for the simplicity of exposition. We can generalize our model to multiple destinations.

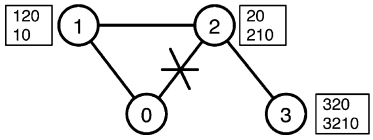


Fig. 1. A BGP system with a link failure. A box lists the paths that are allowed by the adjacent node to export to its neighbors. The paths are ordered in the descending order of preference. Note that the paths in a box represent a node's all possible routes to the destination, and might not be available at the same time. The link with a cross represents a failed link.

In a state transition graph, an edge between states S and S' represents the transition from state S to S' given a trigger set T in state S. A trigger set is activated by the path-selection process at each router. The process proceeds at every individual router independently and triggers the advertisements and withdrawals of routes. Formally, we model the BGP route decision process as follows. Once a routing update is received, it will trigger the corresponding BGP routers (or nodes) 1) to apply the import policy to receive routes; 2) to run the BGP path-selection process; and 3) to apply export policy for generating routing updates to the speaker's neighbors.

Note that the exact timing that a BGP speaker sends an update along an edge to its neighbor is determined by the *Minimum Route Advertisement Interval* (MRAI) Timer together with other factors, such as CPU load, number of BGP peers, etc. [27].

Given a state $S = (s_1, s_2, \ldots, s_n)$ and a trigger set $T \subseteq U$, the next state $S' = (s'_1, s'_2, \ldots, s'_n)$ and the next routing updates that will be triggered, U', can be derived as follows:

$$S' = \mathrm{DecisionProcess}(S, T)$$
$$U' = (U - T) \bigcup \mathrm{New}T$$

where $\mathrm{DecisionProcess}(S, T)$ derives the new state for each router by running the import policy and best path selection process if the router receives a routing update. Otherwise, routers remain in the same state. $\mathrm{New}T$ is a set of routing updates that are triggered by the decision process. In other words, $\mathrm{New}T$ contains routing updates generated by routers whose state has changed and such changes are allowed to be exported to neighbors according to the export policies. The union operation is performed on $(U - T)$ and $\mathrm{New}T$ so that the old updates in $(U - T)$ are replaced by the new ones if the relevant edges have updates to be triggered in both $(U - T)$ and $\mathrm{New}T$.

For example, Fig. 1 shows a BGP system and Fig. 2 shows the state transition graph for the BGP system shown in Fig. 1 with the link (2 0) failure event. In Fig. 1, we show both export and import routing policies in the BGP system, in which the paths that are allowed to be exported are shown in the boxes around the relevant nodes, and local preference ranking is indicated with the order of the paths. Note that this example only considers eBGP sessions to make its corresponding state graph

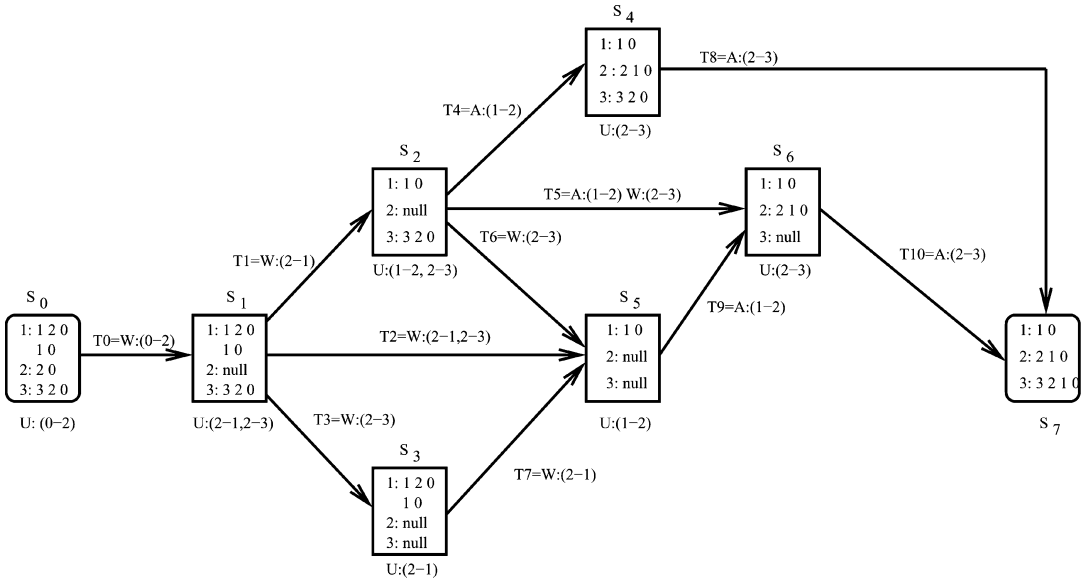


Fig. 2. The state transition graph for the BGP system described in Fig. 1. Letter "A" represents a BGP announcement, and "W" represents a withdrawal.

simple enough. All the concepts introduced so far can be applied to routers with iBGP sessions.

A *transition path* in the state transition graph is composed of a sequence of states from the initial state to the final state. A transition path is associated with a *trigger sequence*, which consists of the trigger sets that lead the state transition along the path.

For example, in Fig. 2, the transition path $(S_0S_1S_5S_6S_7)$ is triggered by the trigger sequence $(T_0T_2T_9T_{10})$, where $T_0 = (0-2)$ contains the withdrawal message from node 0 to node 2 and so on.

B. Transient Failure Routing States

A state S in a state transition graph is a *transient state* if the next state $S' \neq S$ where $S' = \mathrm{DecisionProcess}(S, T)$ and $T \subset U$. A state S in a state transition graph is a *stable state* if the next state $S' = S$ for any $T \subset U$. Griffin *et al.* have shown in [18] that in a stable state, the best paths to the destination formed from all BGP speakers is a directed tree where the direction of each edge is the same as the direction that packets traverse to reach the destination. We refer to this direct tree as *best path tree* of the stable state. In this paper, we focus on the transient behavior of a BGP system that can always reach a stable state after a given event. Here, we assume that the BGP system has a single stable state because we focus on studying transient routing behavior, not permanent routing instability, which have been focused by many other works [18], [20], [21], [25], [26]. The stable state is referred to as the *final state*. We also assume that the BGP system is in a stable state before the event. The state is called the *initial state*.

Definition 1: A state is called a *control plane failure state for a router* if the router has no route entry in this state.

If a router does not have a route entry to a destination, the router will drop all packets destined to the destination. Whether a router goes through a control plane failure state depends on the transition path from the initial state to the final state it traverses. For example, in Fig. 2, router 3 does not go through any control plane failure state for the transition path triggered by the trigger sequence $(T_0T_1T_4T_8)$, while it does for the trigger sequence $(T_0T_3T_7T_9T_{10})$. On the contrary, router 2 will definitely go through a control plane failure state no matter what the trigger sequence is.

Definition 2: A router will experience a *potential control plane failure* if there is a transition path that contains a control plane failure state for this router.

Definition 3: A router will experience a *definite control plane failure* if every transition path contains a control plane failure state for this router.

A *forwarding path* is the path that packets actually traverse from a router to a destination. The forwarding path of a router in a state can be constructed by starting from this router and iteratively appending the next hop router of every router along the path. If a router has no forwarding path that reaches the destination or the path contains a loop, the router has a *null* forwarding path. A null forwarding path implies that the packets forwarded by the router might be dropped somewhere along the path due to control plane routing failures or loops.

Definition 4: A state is called a *data plane failure state for a router* if the router has a null forwarding path in this state.

Definition 5: A router will experience a *potential data plane failure* if there is a transition path that contains a data plane failure state for this router.

Definition 6: A router will experience a *definite data plane failure* if every transition path contains a data plane failure state for this router.

It is clear that if a router goes through a control plane failure state, it is sufficient, but not necessary, for the router to experience a data plane failure state. For example, in Fig. 2, in state S_1 and S_2 router 3 has no forwarding path to the destination, while its routing table does contain a route to the destination.

Note that in most cases routing failures would lead to packet losses. However, there are some subtle situations where routing failures might not cause packet losses. Control plane failures would not result in packet losses unless the routers forward

packets based on the IP forwarding table only. If the routers employ other routing mechanisms to forward packets (e.g., in a MPLS network, routers forward packets based on the MPLS labels), the absence of routes would not necessarily mean packet losses. Meanwhile, data plane failures might not necessarily imply packet losses either. When a router experiences data plane failures, it just implies that control plane routing failures or loops present somewhere in its forwarding path. It is possible that the failures or loops are resolved before the packets arrive at the place. Packet losses would happen only if packets take no time to traverse the forwarding path. However, given the state-of-the-art data transmission technologies, IP packets usually take negligible time to traverse the network. Thus, the occurrence of a data plane failure would most likely indicate that the forwarded packets would be dropped somewhere in its forwarding path.

C. Routing Events

A BGP system can go through a series of state transition after the occurrence of a routing event. These events include link failures, BGP session resets, link additions, topological changes, or routing policy changes. To systematically analyze transient routing behavior, we focus on the routing events in which the destination is always physically connected with the network. In particular, we consider the following two routing events:

- **Failover Event:** The current route to the destination becomes unavailable and is replaced by a less preferred alternative route with different next-hop or AS path to the destination. This event may be caused by link failure, router failure, and routing policy change.
- **Recovery Event:** The current route is replaced by a more preferred route while it is still available. This event may be resulted from link repair, the addition of new routes due to policy or network failures.

We focus on transient states during the failover events and recovery events in the remaining sections.

III. Path Availability Graph

Even though our state transition graph model can precisely capture the transient routing failure states, it is challenging to derive any general conclusion based on this model. Due to its lack of scalability, a state transition graph for any network in a practical size could have an astronomical number of states, which makes it impractical to scrutinize all possible states in the graph to find the transient routing failure states. It turns out that the state transition graph can be much more complex than the original BGP system. As we described before, the state transition graph is introduced only to formally define transient routing states. Instead of designing an efficient algorithm for computing the state transition graph, we introduce the concept of path availability graph. Since whether a route will experience routing failures depends on the availability of routes, we use the path availability graph to determine whether a router might experience transient failures. The path availability graph helps us to simplify the study of the conditions that a router will experience routing failures.

Formally, we construct a directed graph $G = (V, E)$, called the *Path Availability (PA) Graph*, to model a BGP system in either a final state or an initial state. The node set V consists of all BGP routers. The edge set E encodes the path availability information between neighboring routers in the relevant state. For clarify, we use G_0 and G' to denote the PA graph in the initial state and the final state. Without explicit mentioning, a PA graph refers to either G_0 or G'.

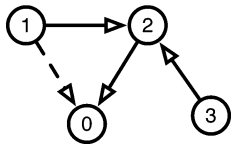


Fig. 3. An example of PA graph. The arrowed solid lines represent a link in the best path tree while the dashed lines represent the bridges.

For a destination d, a PA graph has two components: (1) a *sink tree* to the destination, and (2) *bridges* between branches of the sink tree. A link (u, v) in the sink tree has the direction from node u to node v if the link is in the best path tree and u uses the path from v. Thus, the paths from the nodes in the sink tree to the destination are the best paths to the destination.

The sink tree is used to evaluate the reachability to the destination, while bridges represent alternative paths to the destination. For two nodes u and v, if the best path of v is announced to u and installed at u as *backup*, we call the link between u and v a *bridge* and the direction of the bridge is from u to v. In a PA graph, a bridge is represented by a directed dashed line. Note that a bridge can be bi-directional if both sides announced their best path to each other as a backup path. On the contrary, a solid edge cannot be bi-directional. Since each node has one and only one best path, it should have one and only one outgoing solid edge.

Fig. 3 shows the corresponding PA graph of the routing system in Fig. 1. There is a bridge between node 1 and node 0. That means, node 1 does not use the direct path to reach node 0. Note that the bridge has a unidirectional from 1 to 0 because node 1 has the direct path from node 0, and 0 cannot install 1's best path due to routing loops.

In a PA graph, a directed line from a node u to a node v, either solid or dashed, indicates the availability of u's path in v's routing table. A *(directed) path* $v_r, v_{r-1}, \ldots, v_1, v_0$ is defined as a sequence of edges, where $v_i v_{i-1}$ is either a solid line on the sink tree or a dashed bridge in the direction from v_i to v_{i-1}. In particular, we define a directed path to the destination that contains one or more bridges as an *alternative path*.

In a PA graph, a node i is a *successor* (*predecessor*) of node j if there exists an edge from node j to node i (from i to j) in the best path tree. Each node in a PA graph can have several predecessors but only one successor.

Similarly, a node i is an *alternative successor* (*alternative predecessor*) of node j if there exists a bridge from node j to node i (from i to j).

In a failover event, the link failure partitions the nodes in the initial state G_0 into two clusters: a disconnected cluster and a connected cluster. After the partition, all nodes that cannot reach the destination through their paths in the sink tree in the initial state compose a *disconnected cluster*. On the contrary, all nodes in a *connected cluster* can reach the destination through their

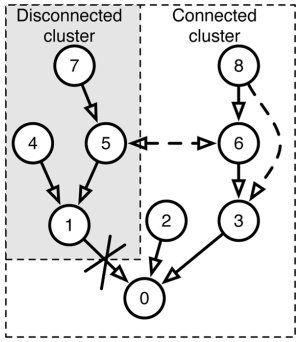


Fig. 4. Example of PA graph in a failover event.

paths in the sink tree in the initial state. At the same time, a bridge can be an *internal bridge* if it is within the disconnected cluster, or an *external bridge* if the bridge is a link between the disconnected cluster and the connected cluster. Since we are considering the failover events, in which the destination is always reachable during the events, there must be at least one external bridge in the graph. After the failure, nodes in the disconnected cluster will switch to the external bridge to reroute. Obviously, the destination belongs to the connected cluster. The two adjacent nodes of the failed link belong to the disconnected and connected cluster respectively. In particular, the adjacent node of the failed link in the disconnected cluster is called the *disconnected cluster root*, denoted by γ.

Similarly, in a recovery event, the nodes in G_0 whose best paths change after the event compose a *recovery cluster* and those unaffected nodes compose the *connected cluster*. The adjacent node of the recovered link in the recovery cluster is named for the *recovery cluster root*.

In G_0 for a failover event, according to the types of bridges contained in an alternative path, we classify the alternative path into two classes:

- *Internal alternative path*: a path that contains no external bridge but may contain internal bridges.
- *External alternative path*: a path that contains *one* external bridge and may or may not contain internal bridges.

In G_0 for a failover event, the *distance* of a node in a disconnected cluster from the disconnected cluster root is defined as the number of hops, which refers to the number of routers away from the cluster root along either the best path or an internal alternative path. Note that an internal alternative path within a disconnected cluster must traverse the disconnected cluster root. The reason is that an internal alternative path must contain a link, which connects a node in the disconnected cluster with the destination in the connected cluster. The link cannot be an external bridge. Otherwise, the path is an external alternative path. Thus, this link must traverse the disconnected cluster root, or the failed link.

Fig. 4 demonstrates an example of PA graph G_0 during a failover event, in which, after link between 1 and 0 fails, routers 1, 4, 5 and 7 fall into the disconnected cluster and the rest of the routers are in the connected cluster. Router 1 is the disconnected cluster root. Router 5 and 7 have external alternative paths while router 1 and 4 not.

In the following sections, we use the PA graphs as a language to specify the sufficient conditions that a router would experience control plane and data plane failures and estimate the upper bounds of the failure durations. Note that the PA graphs are introduced to convey ideas in the general BGP systems only. Later, we will show that the notions of AS relationships suffice to describe the sufficient conditions in the typical BGP systems.

IV. Sufficient Conditions for Transient Control Plane Failures

During a failover event, in which the routers lose their preferred paths to the destination, those routers might temporarily lose all their paths and experience control plane failures. In addition, even during a recovery event, in which the routers gain their preferred paths, control plane failures can still occur. In this section, we specify the sufficient conditions of control plane failures for both failover and recovery events.

A. Transient Failures During Failover Events

As shown in Fig. 4, nodes in the connected cluster will not lose their best paths after the link between 1 and 0 fails. However, because nodes 5 and 7 always have paths from the neighboring nodes in the connected cluster, they will not experience transient failures after the failure. Inspired by this example, we specify the following sufficient conditions for potential control plane failures.

Theorem 1 (Conditions for Potential Control Plane Failures): A node u in a BGP system will experience a potential control plane failure when a link l in the corresponding sink tree fails, if

i) u is in the disconnected cluster; and
ii) u has no external alternative path to the destination.

Proof: Suppose that node u's longest distance in the disconnected cluster is N. We will show with induction on N that we can construct the trigger sequence that leads this node to withdraw all its available paths including the best path and the internal alternative paths. Namely, the node has a chance to experience control plane failures.

Base case: according to condition (ii), the disconnected cluster root has no external alternative path. Besides, it has no internal alternative path either. Because an internal alternative path must traverse the failed link l, the root cannot have such an internal alternative path besides its best path. Thus, the root's best path through the failed link l is the only available path in the routing table. With a trigger sequence that contains the withdrawal on the failed link l sent to the disconnected cluster root, the root node will withdraw its best path and experience a control plane failure.

Induction step: Assume that any node u with longest distance $\leq N-1$ from the disconnected cluster root will experience control plane failures with a trigger sequence T_u if it satisfies condition (i) and (ii). For a node, say v, with a longest distance of N from the disconnected cluster root, all its successor and alternative successors (neighbors who provide v's best path or internal alternative paths) are denoted as $u_1, \ldots, u_i$. The longest distance from these neighbors to the disconnected cluster root is no more than $N-1$. Therefore, there must exist a state in which

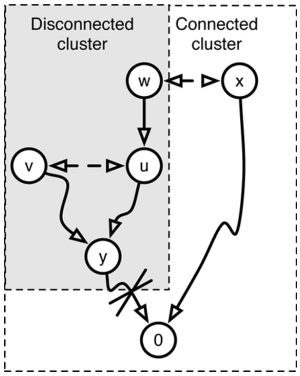


Fig. 5. Node u has an internal alternative path to the destination and it may or may not experience control plane failures during routing convergence.

$u_1, \ldots, u_i$ experience control plane failures. So with a trigger set $T_{\{u_1,\ldots,u_i\}\to v} = (u_1 \to v, \ldots, u_i \to v)$ consisting of withdrawal messages from $u_1, \ldots, u_i$ to v, this state will transit to the next state in which v withdraws all its available paths and experiences a control plane failure. ■

Unlike potential control plane failures, for which we need to show the existence of such trigger sequence that leads to control plane failures only, determining the sufficient condition for a definite control plane failure is challenging. The trigger sequence depends on the topology of a PA graph, routing policies, etc. So it is complicated to enumerate all possible sequences. For example, as shown in Fig. 5, it is not straightforward to determine if node u experiences a definite control plane failure or not. If node v receives the withdrawal message from node y before node u does, node u will lose all alternative paths. On the contrary, if node u receives the withdrawal message from node y before node v does, it will first use the alternative path from node v as a new backup path even though the path is invalid. After then, it can advertise the new path to its predecessor w. Suppose that at node w, the new path is longer than that via node x. As a result, node w will switch to use the shorter one so that node w becomes node u's successor. Finally, node u receives the alternative path from w. After that time, node u will not experience a transient routing failure.

The above example shows that a BGP system containing internal bridges in its disconnected cluster can complicate the generation of trigger sequences. There is a race between the announcements of alternative paths and the withdrawal messages. When the announcements arrive at a node first, the node will not experience a routing failure. Otherwise, the node will experience a transient failure. Therefore, if a node has neither external nor internal alternative paths, it will definitely experience control plane failures.

Motivated by the above observation, we have the following sufficient condition for definite control plane failures. ■

Theorem 2 (Conditions for Definite Control Plane Failure): A node u in a BGP system experiences a definite control plane failure when a link l in the corresponding sink tree fails, if

i) u is in the disconnected cluster; and
ii) u has no external alternative path to the destination; and
iii) there is no internal bridge in the disconnected cluster; and
iv) the removal of u and l disconnects the disconnected cluster root from the destination.

Proof: Suppose that u's longest distance to the disconnected cluster root is N. We will prove the theorem by the induction on N.

Base case: the disconnected cluster root has $N = 0$. According to condition (ii), the root node has no external alternative path. In addition, the root node has no internal alternative path. Therefore, the best path through the failed link l is the root node's only available path in the routing table. After the link failure, the root node will definitely withdraw its best path and experience a control plane failure.

Induction step: suppose that every node with a longest distance $N-1$ experiences a definite control plane failure if they satisfy conditions (i)–(iv). For a node, say u, which has N distance to the disconnected cluster root, does not have an external alternative path according to condition (ii). At the same time, condition (iii) implies that node u has a successor, say v, but has no alternative successor. Thus, u's best path through its successor v is the only available path. According to the induction assumption, v, which has a longest distance $N-1$ from the root, will experience definite control plane failure. After the failure, v will sent a withdrawal message to u. We need to show that u must receive this message. Due to condition (iii), u's predecessors will not change their path until u changes its path since their paths are learned from u. So, the withdrawal message from v is the first message that u will receive after the link failure. Further, due to condition (iv), v's alternative paths traverse u. Thus, u must change its path before v gets its alternative path. If u would not receive this withdrawal message from, would never change its path and v would never receive the alternative paths either. Therefore, after v experiences transient failure, u must receive the withdrawal message from v and experience transient failure. ■

B. Transient Failures During Recovery Events

Previous work has shown that end-to-end paths can experience packet losses and packet delay during a recovery event [1], [16]. In this section, we first use examples to demonstrate how transient failures can occur during recovery events. Then, we analyze the sufficient conditions for a node to experience transient failures during recovery events. To simplify our analysis, we emulate a recovery event as a link repair.

We first illustrate the occurrence of a control plane failure during recovery events. In Fig. 6, node 1 uses the direct path to the destination (1 0) as its best route, and nodes 2 and 3 use the paths (2 1 0) and (3 1 0) from node 1 as their best path, respectively. Once the link between node 3 and node 0 is recovered, node 3 uses the direct path (3 0) as its best path, and propagates it to node 1 and node 2. Suppose that node 1 will use the new path, and node 1 cannot propagate this new path to node 2 according to its routing policy. In this case, node 1 sends a withdrawal message to node 2 to withdraw its previously announced route. If the withdrawal message arrives at node 2 earlier than the new path sent by node 3, node 2 will lose its current route to the destination, as shown in Fig. 6.

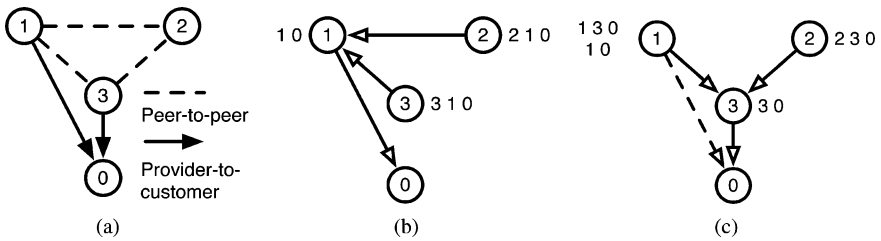


Fig. 6. A control plane failure experienced by node 2 during the link between 0 and 3 is recovered. (a) Topology. (b) PA graph before a recovery event. (c) PA graph after route converges.

Similarly, we use PA graphs to understand control plane failures during a recovery event. As shown in Fig. 6, the adding of link between nodes 3 and 0 can lead to the removal of the bridge from node 1 to node 2 in the PA graph, and cause a control plane failure at node 2. In the initial state, node 1 is node 2's successor while in the final state, it is neither node 2's successor nor alternative successor. Further, in the final state, node 1's best path is learned from node 3 rather than node 2, which ensures that node 1 can get its best path in the final state and then withdraw its path from node 2 before node 2 gets its alternative paths. As a result, a control plane failure can occur at node 2. With this observation, we derive the following lemma to identify control plane failure occurring at nodes like node 2.

Lemma 1: A node u will experience a potential control plane failure upon the recovery of a link l, if

i) u is in the recovery cluster but not the recovery cluster root; and
ii) u's successor and alternative successors in G_0 are no longer u's successor or alternative successor in G', and their best paths in G' do not contain u.

Proof: The conditions show that 1) in G', u loses its best path and all alternative paths in G_0, and 2) in G', u uses a new path from a node other than its successor and alternative successors in G_0. Suppose u's successor and alternative successors are $v_1, \ldots, v_n$ in G_0, and $x_1, \ldots, x_m$ in G'. With a trigger set $T_{v_1,\ldots,v_n \to u} = (v_1 \to u, \ldots, v_n \to u)$ which consists of withdrawal messages from $v_1, \ldots, v_n$ to u before u receives the new paths from $x_1, \ldots, x_m, u$ will lose all its routes and experience a control plane failure. We construct the trigger sequence in the following way. Since u is not in the best path trees of either v_i or x_j, with an induction procedure similar to that in the Proof of Theorem 1, we can construct trigger sequences such that each of v_i and x_j has its best path in the G' installed in its routing table while no trigger set is issued to u. Thus, u's path is unaffected by all these trigger sequences. Then, with an additional trigger $T_{v_1,\ldots,v_n \to u}$, in which $v_1, \ldots, v_n$ withdrawal their paths from u while $x_1, \ldots, x_m$ hold their route announcements, u will lose all its existing routes and experience a control plane failure. ■

Further, the nodes whose best paths are via the nodes that satisfy Lemma 1 can experience potential control plane failures as well. For example, in Fig. 6, suppose that there is a node, say node 4, which is a stub node connected to node 2 only (does not show in the figure). After node 2 experiences a control plane failure, the withdrawal message from node 2 to node 4 will cause node 4 to withdraw its only path through node 2 and experience a control plane failure. Motivated by the example, we use the following procedures to identify such stub nodes.

In a recovery event, the nodes identified with Lemma 1 are named as *recovery-induced failure nodes*. Similar to the disconnected cluster root in a failover event, based on a PA graph, those nodes, whose best path trees in the initial state traverse the recovery-induced failure nodes, compose the *recovery-induced disconnected cluster*. Similarly, bridges connecting nodes within the same recovery-induced disconnected cluster are *recovery-induced internal bridges* while those connecting one node in a recovery-induced disconnected cluster and another node outside of the cluster are *recovery-induced external bridges*. The alternative paths containing internal bridges only are called *recovery-induced internal alternative paths* while the *recovery-induced external alternative paths* contain external alternative bridges. With these terms and according to the same idea of Theorem 1, we have the following lemma to further identify nodes that might experience potential control plain failures during a recovery event. The lemma can be proved in the similar way that we prove Theorem 1, i.e., we can construct such a trigger sequence that leads the relevant nodes to experience control plane failures. Note that there can be multiple recovery-induced failure nodes in a recovery event while there is only one disconnected cluster root in a failover event.

Lemma 2: Node u will experience a potential control plane failure in a BGP system when a link l is recovered, if u is in the recovery-induced disconnected cluster, and u has no recovery-induced external alternative path to the destination.

Combining Lemma 1 and 2, we get the following sufficient conditions for potential control plane failures in the recovery events.

Theorem 3 (Conditions for Potential Control Plane Failures): Node u will experience a potential control plane failure in a BGP system when a link l is recovered, if

i) u is a recovery-induced failure node; or
ii) u is in the recovery-induced disconnected cluster and has no recovery-induced external alternative path to the destination.

V. Sufficient Conditions for Transient Data Plane Failures

In this section, we analyze the sufficient conditions for data plane failures. Any node that experiences control plane failures will definitely experience data plane failures. Therefore, the sufficient conditions for control plane failures are also the sufficient conditions for data plane failures. Moreover, even though a node does not experience any control plane failures, it is possible for the node to experience data plane failures. For example, once a node experiences a control plane failure, any node whose best

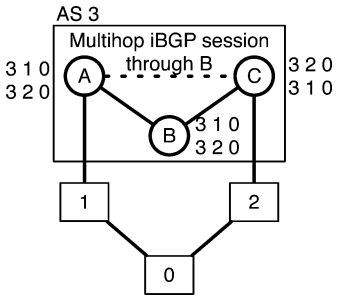


Fig. 7. Data plane failure due to a forwarding loop between node A and node B when the link between node A and AS1 is failed.

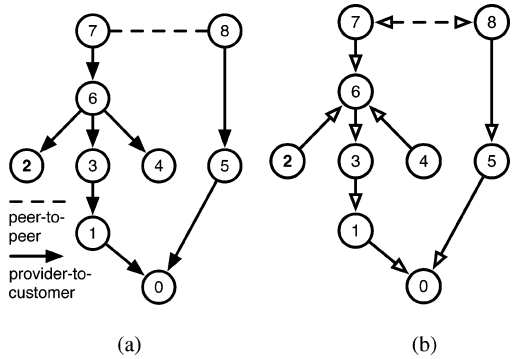


Fig. 8. Transient control plane failures take place in a hierarchical eBGP system. (a) AS relationships. (b) PA graph.

path traverses this node at that moment will experience a data plane failure.

Further, we develop the following theorem for the sufficient conditions for definite data plane failures.

Theorem 4 (Conditions for Definite Data Plane Failures): Node u in a BGP system can experience a definite data plane failure when a link l in the corresponding sink tree fails, if

i) u is in the disconnected cluster; and
ii) the disconnected cluster root has no external alternative path to the destination.

Proof: According to Theorem 2, the disconnected cluster root will experience a definite control plane failure. As soon as the disconnected cluster root experiences a definite control plane failure, every other node in the disconnected cluster still use the best path through the root node. Therefore, these nodes will experience definite data plane failure. ■

Note that besides control plane failures, forwarding loops can also lead to data plane failure. We use the following example to demonstrate that the data plane failures can be caused by forwarding loops during a failover event. Several previous works have been focused on routing loops [14], [13]. We focus on understanding transient routing failures due to lack of available routes, and studying routing loop is beyond the scope of this paper.

The example in Fig. 7 shows that a failover event can cause a forwarding loop in an iBGP system. In this example, AS3 has three BGP routers, which are connected with full meshed iBGP sessions. However, there is no direct layer-2 connection between A and C but the session has to be a multihop iBGP session through B. Nodes A and C each has an eBGP session with AS1 and AS2, respectively. Meanwhile, node B prefers path learned from node A. Suppose the link between node A and AS1 fails. Node A has to first use the alternative path learned from node C, and then inform B of the path change. Before being informed, node B continues forwarding packets to node A. Thus, a forwarding loop is formed between node A and node B. A data plane failure occurs.

Note that for the example shown in Fig. 7, MPLS can avoid those transient data plane failures. In this case, node B will forward packets according to the labels instead of route entries in its forwarding table, and will forward the packets to the right nodes indicated by the MPLS labels.

VI. Transient Failures in a Typical BGP System

In Section IV, given the topology of any BGP system and its routing policies, we use a PA graph to identify transient routing failures in the system. In this section, we show how to use our model to analyze transient failures in a typical BGP system, in which the typical routing policies that are commonly practiced by the ISPs are employed and the typical iBGP configurations are deployed. In this setting, we do not necessarily rely on the relevant PA graphs to identify the sufficient conditions for transient routing failures. We first study transient failures in a *typical hierarchical eBGP system*, in which the neighbors of an AS can be classified as providers, customers or peers according to their commercial agreements, and we assume that one AS consists of one BGP router only. Further, in a typical hierarchical eBGP system, every AS applies *typical routing policies* [17]. That is, an AS announces its customer routes to all neighbors but its peer or provider routes to its customers only. Besides, every AS prefers its customer routes over its peer routes and then over its provider routes. Second, we discuss transient failures occurring within a *typical hierarchical iBGP system*, which consists of a core with full meshed core routers, as known as route reflectors, and the edge routers which are the clients of the relevant route reflectors.

A. Transient Behavior in a Typical Hierarchical eBGP System

In this section, we assume that an AS consists of one BGP router only. This simplification helps our analysis focus on the transient routing failure caused by the eBGP configurations. Under this assumption, we examine the transient routing failures experienced by the whole AS. Apparently, if an AS would experience transient routing failures in the AS level, its routers would definitely experience transient routing failures in the router level. Therefore, for a router, this assumption does not miss any transient routing failures caused by the interactions between ASes. However, it significantly simplifies our analysis by neglecting those failures caused by the inconsistency within the ASes' iBGP systems. The transient routing failures caused by the iBGP configurations will be examined later.

Before diving into the theoretical analysis, we first use an example to show transient failures can be prevalent in a typical hierarchical eBGP system. Fig. 8(a) shows the AS relationships

for a typical BGP system. Fig. 8(b) shows the corresponding PA graph. Here, every AS prefers the route from customers to those routes from providers or peers. Now, suppose that the link between nodes 1 and 0 breaks. Nodes 6,1, and 3 satisfy the sufficient condition for definite control plane failure. That is, they do not have external alternative paths. Node 2 and node 4 satisfy the conditions for potential control plane failure, and node 7 has an external bridge. Therefore, nodes 1, 3, and 6 will definitely experience transient failure. Node 2 may experience the failure and node 7 does not experience any control plane failure.

From the above example, we observe that AS7 and AS8 do not experience any transient routing failures because they always have alternative paths. We can tell that an AS experiences control plane failures depending on whether its stable route in the final state is a customer, peer or provider route.

Lemma 3: In a typical hierarchical eBGP system, if AS u converges to a customer route after a link failure, AS u does not experience any potential control plane failure.

Proof: Suppose after the link failure, AS v_k uses path $P = (v_k v_{k-1} \ldots 0)$ to reach AS0, who originates the destination, and $(v_k v_{k-1})$ is a customer link. According to the no-valley policy, $(v_i v_{i-1})$ must be a customer link for any $i = k, k-1, \ldots, 2$. We prove the lemma by induction on i.

Base step: $i = 2$. Since after a transient failure, AS v_2 has a path $(v_2 0)$ to AS0. AS v_2 must install this path before the failure because AS0 advertises the destination to all providers. Meanwhile, due to prefer-customer routing policies, v_2's best path to AS0 should also use a customer link. When the link failure occurs, v_2 has at least one path $(v_2 0)$ to AS0. Therefore, v_2 does not experience a transient failure.

Induction step: Assume that before the transient failure, AS v_{k-1}'s best path to AS0 goes through a customer link. After the failure, its best path to AS0 still goes through a customer link and v_{k-1} does not experience any transient failure. Then v_k always has at least one path to AS0 through v_{k-1}. Therefore, v_k does not experience a potential control plane failure. ■

Lemma 4: In a typical hierarchical eBGP system, if AS u converges to a peer route after a link failure, AS u does not experience any potential control plane failure.

Proof: Suppose after the link failure, AS u uses a peer link through its peer AS v to reach the destination. Due to the no-valley policy, v must use a customer link to reach the destination after the failure. According to Lemma 3, v will not experience a transient failure, which implies that v always has a customer path during the failure. The paths must be advertised and installed at u. Thus, during the failure, AS u will always has at least one path to the destination through v. So, AS u will not experience a potential control plane failure. ■

We can apply Theorem 1 to the typical hierarchical eBGP system to identify the nodes that experience control plane failures. Further, we develop the following theorem to identify the nodes that experience potential control plane failures in a typical hierarchical eBGP system.

Theorem 5 (Conditions for Potential Control Plane Failure): In a typical hierarchical eBGP system, an AS u will experience a potential control plane failure when a link l fails if

i) u is the successor of all of its providers in the initial state; and

ii) u uses a provider route in the final state.

Proof: According to condition (i), u must use a customer route in the initial state. Further, condition (ii) shows that u changes its best path during the failover event. Thus u belongs to the disconnected cluster. Also, because u switch to a provider route in the final state. According to Lemma 3 and 4, u has no external alternative path through either customers or peers. Otherwise, u should have either a customer route or a peer route after the failure. Then u will not experience a control plane failure. Condition (i) shows that u has no external alternative path through providers. Therefore, u has no external alternative paths. According to Theorem 1, u will experience potential control plane failures. ■

According to the above lemmas, we have the following corollary for a tier-1 AS.

Corollary 1: In a typical hierarchical eBGP system, a tier-1 AS cannot experience any potential control plane failure during failover events.

Proof: Since a tier-1 AS has no provider, it has to use a customer route or a peer route after a failover event. According to Lemma 3 and 4, the tier-1 AS can not experience any control plane failure. ■

However, for the recovery events, we cannot find a node experiencing control plane failures in a typical hierarchical eBGP system.

Theorem 6: There is no control plane failure during a recovery event in a typical hierarchical eBGP system.

Proof: We need to prove that there is no such node that satisfies conditions in Lemma 1. We show this by contradiction. Assume there is a node u experiencing a control plane failure during a recovery event, and $v_1, \ldots, v_n$ are u's successor or alternative successors in G_0. We will show that there is no such v_i in the setting of a typical hierarchical eBGP system.

At first, v_i cannot be u's provider. Because in G', v_i has its best path that does not traverse u, v_i must inform u this path and thus is u's successor or alternative successor, which contradicts with the conditions in Lemma 1. Second, v_i cannot be u's peer. Since v_i is a successor or alternative successor of u in G_0, v_i must use a customer route in G_0. Otherwise, node v_i cannot advertise this route to node u. In G', v_i changes its best path. Because the recovery event cannot eliminate the existing route of v_i, v_i's new best path must be another customer route. Then v_i must inform u this new path and thus becomes u's successor or alternative successor, which again contradicts with the conditions in Lemma 1. Third, v_i cannot be u's customer. In G_0, node v_i's best path must come from a customer. Otherwise, node v_i cannot advertise a route from its provider or peer to its provider. Similar to the peer case, v_i's best path in the final state should be also a customer route, which should be announced to u and thus contradict to the conditions in Lemma 1.

Therefore, there is no node that satisfies conditions in Lemma 1. ■

In terms of the conditions for data plane failures in a typical hierarchical eBGP system, similar to the discussions in Section V, we can always identify nodes that experience data plane failures with the following two steps: 1) use the sufficient conditions for control plane failures to identify nodes experi-

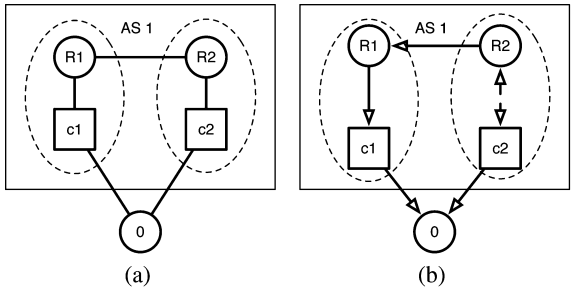


Fig. 9. A tier-1 AS with a hierarchical iBGP structure and corresponding PA graph. (a) Topology. (b) PA graph.

encing control plane failures; and 2) find other nodes that use them to reach the destination.

B. Transient Failures Within an iBGP System

In this section, we consider transient failures caused by the configurations within an AS, which consists of a set of iBGP routers. ASes usually have either a fully meshed or a 2-tiered hierarchical iBGP structure. Here we focus on the 2-tiered hierarchical iBGP system, which is used by most large ASes. Similarly, we can describe scenarios that routers in a fully meshed iBGP structure experience transient routing failures.

In a hierarchical iBGP system, there are route reflectors, which are the *backbone routers*, and a set of *edge routers*, which are the router reflectors' clients. An edge router could be an *access router* that connects to a customer network, or a *peer router* that connects to a peer network. In terms of route export and import policies, a pair of route reflectors have a peer-to-peer like relationship, i.e., a route reflector does not transfer the routes between two other route reflectors. A route reflector and its clients have a provider-to-customer like relationship, i.e., they import and export routes from and to each other without discrimination.

Transient routing failures can occur within a hierarchical iBGP system. For example, in Fig. 9(a), AS1 has two routers, $R1$ and $R2$, and two clients $c1$ and $c2$. There are two paths from two clients to d. Suppose that route reflectors $R1$ and $R2$ both select client $c1$ as the closest egress point, as shown in Fig. 9(b). Here, such routing policies could be caused by iBGP configuration. For example, the MED value of the path via $c1$ is lower than that of the path via $c2$, the length of the AS path via $c1$ is shorter than that of the path via $c2$, or at route reflector $R2$ IGP value of the path via $c1$ is lower than that of the path via $c2$. As a result, the path via $c2$ is invisible to $R1$. Based on sufficient conditions described in Theorem 2, we know that once the path via $c1$ is unavailable, router $R1$ will experience a definite routing failure.

Next, we use another more complex example to illustrate how routing failures can occur within an AS. Suppose that a multihomed customer connects to a tier-1 AS through its providers in different geographic locations. Meanwhile, tier-1 ASes are connected to each other in multiple geographic locations. For example, in Fig. 10, AS0 has two providers, AS2 and AS3. AS1 can reach a destination originated at AS0 via one of three different edge routers, ER1, ER10, or ER11. Suppose that AS2 is a customer of AS1, and AS3 is a peer of AS1. According

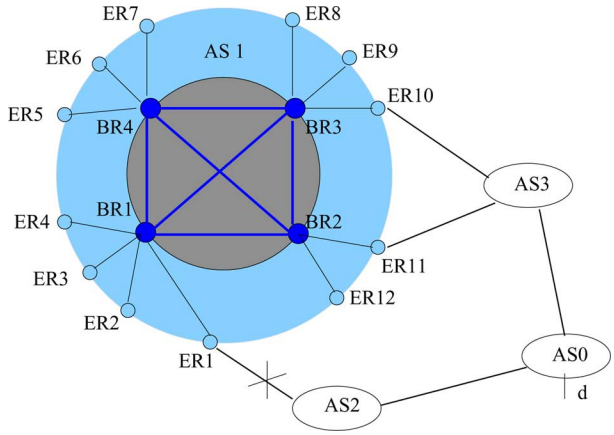


Fig. 10. A tier-1 AS with a hierarchical iBGP structure. The dark nodes represent route reflectors, and the gray nodes represent clients of route reflectors.

to prefer-customer routing policy, the path via ER1 is assigned higher local preference value than those via ER10 and ER11. As a result, all routers inside AS1 will use the path via ER1 to reach the destination. Once the link between ER1 and AS2 fails, all routers inside the AS are in the disconnected cluster. According to the sufficient condition for definite control plane failure, access router ER1, and route reflector BR1, BR2 and BR3 will experience a definite transient failure because they have only one path through ER1 to reach the destination before the failure, and all of their predecessors have only one path not via the failed path to reach the destination. All routers except ER10 and ER11 will experience potential control plane failures according to sufficient condition for potential control plane failures.

We can apply Theorem 1 to an iBGP system to identify which router can experience control plane failures. Further, motivated by the above examples, we develop the following theorem to identify the routers that experience potential control plane failures within an iBGP system.

Theorem 7 (Conditions for Potential Control Plane Failure): In an AS, if all edge routers to a destination select the same egress point to reach a destination, all other routers will experience potential control plane failures once the egress point loses its connection to the destination.

Proof: Suppose that u is a non-edge router to a destination and v is the egress router. Since all edge routers to the destination select v as their egress point, all other routers will also select v as their egress point. Once v loses its connection to the destination, all routers within the AS are in the disconnected cluster, and only those edge routers might have external alternative paths. According to Theorem 1, u will experience a potential control plane failure. ■

Note that while Theorem 5 applies to edge routers only, Theorem 7 applies to all routers within an AS including route reflectors and clients. For example, in Fig. 10, edge routers to the destination, ER10 and ER11, select ER1 as their egress point because the routes via ER1 have higher local preference values. According to Theorem 7, the routers from ER1 to ER9, ER12, and the backbone routers from BR1 to BR4 will suffer from potential routing failures when the link between ER1 and AS2 fails.

VII. Related Works

Previous studies focus on understanding the stability of interdomain routing. Several abstract models [17], [18], [20]–[23] for routing convergence properties aim to capture the long-term routing stability. For example, Griffin *et al.* show that routing policy conflicts could lead to protocol divergence and characterize sufficient conditions for BGP route convergence [18], [20], [21]. Gao and Rexford [17], [22] exploit AS commercial relationships to ensure the convergence of the BGP system. However, all of them focus on the long-term stability instead of transient routing behavior. Our model differs from these existing models in the sense that we strive to capture the transient behavior of BGP, and identify the potential transient routing failures.

Several works have focused on convergence delay of BGP. Labovitz *et al.* analyze the convergence delay of BGP and derive theoretical upper and lower bounds for the convergence delay [1], [4], [9]. Their work focuses on the convergence delay when a network prefix becomes available or unavailable. Obradovic developed a real-time BGP model to analyze the same type of convergence delay in a hierarchical eBGP system [28]. Our work focuses on routing failures that occur during the path exploration process triggered by a link failover or recovery event.

Correlation between end-to-end path failures and routing instability has been studied through measurement. Paxson identified Internet failures and discovered that routing instability can disrupt end-to-end connectivity [29]. Feamster *et al.* studied the location and duration of end-to-end path failures and correlated end-to-end path failures with BGP routing instability [3]. Their results show that most path failures last less than 15 minutes and most failures that coincide with BGP instability appear in the network core. Teixeira *et al.* [30] found that routing changes are the cause of the majority of the large traffic variations within a large ISP network. On the other hand, routing failures within an AS have been studied in [5], which shows that failures are correlated with IS-IS routing updates. Our work complements those works by focusing on interdomain routing failures.

The key to avoiding transient routing failures is to improve the visibility of alternative routes in the BGP system. BGP routers, announcing to their neighbors only the single best path for each destination, limit the visibility of alternative routes. One solution is to advertise the hidden routes. Kushman *et al.* [32] present methods to advertise hidden routes to avoid transient failures. For example, ER10 and ER11 in Fig. 10 will announce their alternative routes through AS3 to other routers. The limitation of their work is that it only focuses on providing fast recovery on AS level. Work [33] provides fast recovery for both eBGP and iBGP. In addition, an AS can adopt some mitigation techniques to alleviate potential transient routing failures. For example, Bonaventure *et al.* [31] propose a fast reroute technique by using tunnels to reroute packets when eBGP session fails. This scheme can also be used to resolve the problem of transient routing failures for iBGP. Alternatively, an encapsulation scheme, such as MPLS or IP-over-IP, can also prevent packet losses even if the routers experience transient routing failures in the iBGP system. On the other hand, in this paper, our work focuses on how and when a transient routing failure occurs.

VIII. Conclusion

In this paper, with the aid of a formal BGP model, we investigated the transient behavior of the interdomain routing protocol. We find that network changes that do not physically disconnect prefixes from the network might still force nodes to temporarily lose reachability to these prefixes. The transient routing failures can have a significant impact on the end-to-end performance in the Internet. Our analytical results show the existence of such transient behavior in today's Internet routing system. Our results have important implications for enhancing Internet reliability. We believe that the results in this paper underscore the necessity of enhancing today's interdomain routing architecture and provide insights in the future design of interdomain routing protocol.

Acknowledgment

The authors would like to thank O. Bonaventure and the anonymous reviewers for their constructive comments. They also thank Prof. B. Tucker and L. Hoglund for their valuable suggestions.

References

[1] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *IEEE/ACM Trans. Networking*, vol. 9, no. 3, pp. 293–306, Jun. 2001.

[2] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, and B. Freeman, "Combining routing and traffic data for detection of IP forwarding anomalies," *Proc. ACM SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 416–417, 2004.

[3] N. Feamster, D. Andersen, H. Balakrishnan, and M. Kaashoek, "Measuring the effects of Internet path faults on reactive routing," in *Proc. ACM SIGMETRICS*, San Diego, CA, Jun. 2003, pp. 126–137.

[4] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental study of Internet stability and backbone failures," in *Proc. FTCS*, 1999, pp. 278–285.

[5] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004, vol. 4, pp. 2307–2317.

[6] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, and C. Diot, "The impact of BGP dynamics on intra-domain traffic," in *Proc. ACM SIGMETRICS*, New York, NY, Jun. 2004, pp. 319–330.

[7] A. Shaikh and A. Greenberg, "OSPF monitoring: Architecture, design and deployment experience," in *Proc. USENIX 1st Symp. Networked Systems Design and Implementation (NSDI '04)*, San Francisco, CA, Mar. 2004, pp. 57–70.

[8] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving sub-second IGP convergence in large IP networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 35–44, 2005.

[9] C. Labovitz and A. Ahuja, "The impact of Internet policy and topology on delayed routing convergence," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, vol. 1, pp. 537–546.

[10] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 515–528, Oct. 1998.

[11] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating Internet routing instabilities," in *Proc. ACM SIGCOMM*, Portland, OR, 2004, pp. 205–218.

[12] D. F. Chang, R. Govindan, and J. Heidemann, "The temporal and topological characteristics of BGP path changes," in *Proc. IEEE Int. Conf. Network Protocols (ICNP'03)*, Atlanta, GA, Nov. 2003, pp. 190–199.

[13] U. Hengartner, S. Moon, R. Mortier, and C. Diot, "Detection and analysis of routing loops in packet traces," in *Proc. 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW'02)*, 2002, pp. 107–112.

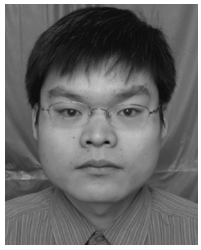
[14] D. Pei, X. Zhao, D. Massey, and L. Zhang, "A study of BGP path vector route looping behavior," in *Proc. Int. Conf. Distributed Computing Systems (ICDCS'04)*, Tokyo, Japan, 2004, pp. 720–729.

[15] F. Wang, L. Gao, J. Wang, and J. Qiu, "On understanding of transient interdomain routing failures," in *Proc. IEEE Int. Conf. Network Protocols (ICNP'05)*, Boston, MA, 2005, pp. 30–39.

[16] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A measurement study on the impact of routing events on end-to-end Internet path performance," in *Proc. ACM SIGCOMM*, Pisa, Italy, 2006, pp. 375–386.
[17] L. Gao and J. Rexford, "A Stable Internet routing without global coordination," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, pp. 681–692, Dec. 2001.
[18] T. G. Griffin and G. T. Wilfong, "An analysis of BGP convergence properties," in *Proc. ACM SIGCOMM*, Cambridge, MA, Aug. 1999, pp. 277–288.
[19] N. Feamster, J. Winick, and J. Rexford, "A model of BGP routing for network engineering," in *Proc. ACM SIGMETRICS*, New York, NY, 2004, pp. 331–342.
[20] T. Griffin, F. B. Shepherd, and G. T. Wilfong, "Policy disputes in path-vector protocols," in *Proc. IEEE Int. Conf. Network Protocols (ICNP'99)*, Toronto, Ontario, Canada, Nov. 1999, pp. 21–30.
[21] T. Griffin and G. T. Wilfong, "A safe path vector protocol," in *Proc. IEEE INFOCOM*, 2000, pp. 490–499.
[22] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, pp. 733–745, Dec. 2001.
[23] L. Gao, T. Griffin, and J. Rexford, "Inherently safe backup routing with BGP," in *Proc. IEEE INFOCOM*, Anchorage, AK, 2001, pp. 547–556.
[24] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. Networking*, vol. 10, no. 2, pp. 232–243, Apr. 2002.
[25] T. G. Griffin, A. D. Jaggard, and V. Ramachandran, "Design principles of policy languages for path-vector protocols," in *Proc. ACM SIGCOMM*, Aug. 2003.
[26] J. Sobrinho, "Network routing with path vector protocols: Theory and applications," in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 49–60.
[27] A. Feldmann, H. Kong, O. Maennel, and A. Tudor, "Measuring BGP pass-through times," in *Proc. Passive and Active Measurement Conf. (PAM'04)*, Antibes Juan-les-Pins, France, Apr. 2004, pp. 267–277.
[28] D. Obradovic, "Real-time model and convergence time of BGP," in *Proc. IEEE INFOCOM*, New York, NY, 2002, vol. 2, pp. 893–901.
[29] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. Networking*, vol. 5, no. 5, pp. 601–615, Oct. 1997.
[30] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan, "Traffic matrix reloaded: Impact of routing changes," in *Proc. Passive and Active Measurement Conf. (PAM'05)*, Boston, MA, Mar. 2005, pp. 251–264.
[31] O. Bonaventure, C. Filsfils, and P. Francois, "Achieving sub-50 milliseconds recovery upon BGP peering link failures," *IEEE/ACM Trans. Networking*, vol. 15, no. 5, pp. 1123–1135, Oct. 2007.
[32] N. Kushman, S. Kandula, D. Katabi, and B. Maggs, "R-BGP: Staying connected in a connected world," in *Proc. 4th USENIX Symp. Networked Systems Design and Implementation*, Cambridge, MA, 2007, pp. 341–354.
[33] F. Wang and L. Gao, "A full route aware routing protocol—Fast recovery from transient routing failures," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 2333–2341.

Feng Wang received the B.E. degree from Zhejiang University, China, the M.S. degree from Yanshan University, China, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts at Amherst.

He is an Assistant Professor with the School of Engineering and Computational Sciences at Liberty University, Lynchburg, VA. His research interests include networked computer systems, Internet routing, and wireless networks.



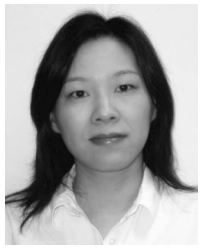
Jian Qiu (S'06) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, China, in 2001 and 2004, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of Massachusetts at Amherst.

His research interests include Internet routing and topology.

Lixin Gao (M'98–SM'07) received the Ph.D. degree in computer science from the University of Massachusetts at Amherst in 1996.

She is a Professor of electrical and computer engineering at the University of Massachusetts at Amherst. Her research interests include multimedia networking, and Internet routing and security. Between May 1999 and January 2000, she was a visiting researcher at AT&T Research Labs and DIMACS.

Dr. Gao is an Alfred P. Sloan Fellow and received an NSF CAREER Award in 1999. She has served on number of technical program committees including SIGCOMM2006, SIGCOMM2004, SIGMETRICS2003, and INFOCOM2004, and is on the Editorial Board of IEEE/ACM TRANSACTIONS ON NETWORKING.



Jia Wang (S'97–M'01–SM'06) received the Ph.D. degree in computer science from Cornell University, Ithaca, NY, in 2001.

She is currently a Senior Technical Specialist member of the Network Measurement and Engineering Research Department in the Internet and Networking Systems Research Center at AT&T Labs—Research, Florham Park, NJ. Her research interests include network measurement and management, routing and topology analysis, network security, overlay networks and applications, and other Internet-related research work. She has published over 40 research papers in leading journals and conferences including IEEE/ACM TRANSACTIONS ON NETWORKING, ACM SIGCOMM, ACM SIGMETRICS, IEEE INFOCOM, ICNP, NSDI, IMC, WWW, and USENIX. She has given tutorials at ACM SIGMETRICS 2005 and IEEE INFOCOM 2004. She has also served on several Technical Program Committees (most recently ACM SIGMETRICS 2005, IEEE INFOCOM 2005–2007, and PAM 2007).