Reliability Models and Evaluation of Internal BGP Networks

Li Xiao and Klara Nahrstedt
Department of Computer Science
University of Illinois at Urbana-Champaign, Urbana, IL 61801
Email: {lixiao, klara}@cs.uiuc.edu

***Abstract*— The performance of global Internet communication is significantly influenced by the reliability and the stability of Internet routing systems, especially the Border Gateway Protocol (BGP), the de facto standard for inter-domain routing. In this paper, we investigate the reliability of BGP sessions and the Internal BGP (IBGP) networks in the environment of unreliable physical and routing layers.**

The reliability analysis of IBGP networks is difficult, because IBGP sessions may be correlated to each other by the shared underlying physical links and TCP enables IBGP sessions to tolerate certain level of network failures. In this paper, we first investigate the failure probability of IBGP sessions and its relation to BGP timers and TCP retransmission behaviors. The result of this investigation is a simple modification of TCP that increases the robustness of IBGP sessions significantly. Second, we present a novel reliability model to measure the resilience of the whole IBGP networks. This model is of great importance for studying the function loss of IBGP operations and it also provides the theory basis for IBGP network optimization in terms of reliability.

I. Introduction

Border Gateway Protocol (BGP) [1] is the widely used inter-domain routing protocol. Two BGP routers, which communicate with each other directly, are called BGP peers. BGP peers exchange routing information via *BGP sessions* which are running over TCP. According to the relation of BGP peers, BGP itself can be divided into two parts: External BGP (EBGP) and Internal BGP (IBGP). An EBGP session connects two BGP routers which reside in different Autonomous Systems (AS); an IBGP session links two BGP routers which belong to the same AS. In this paper, we focus on reliability modeling for the *IBGP network* which consists of BGP routers in one AS and the IBGP sessions among them.

In the traditional IBGP network, IBGP sessions form a full mesh over all BGP routers in a domain. Furthermore, two hierarchical IBGP structures, route reflection [2] and confederation [3], have been proposed to solve the scalability limitation in the full mesh design. Fig. 1 shows an example of two-level IBGP route reflection network. BGP routers are divided into three clusters. In each cluster, at least one router is chosen as a route reflector (A, B, E and I), and other routers are route-reflector clients. In cluster I, redundant reflectors are used for better reliability. All reflectors establish a full mesh via IBGP sessions. A client is only required to share IBGP sessions with the reflectors in its cluster. The sessions between clients of the same cluster are optional. A client is the traditional BGP router and it only needs to communicate with its reflectors. A reflector is responsible for: (1) reflecting routes (routing information) from its client to the peer reflectors and the other clients; (2) reflecting routes from its peer reflectors to its clients. CLUSTER_LIST loop detection mechanism prevents routes being reflected back to the clusters where the routes originate.

This work was supported by NSF under contract number NSF ANI 00-73802. Any opinions presented in this material do not necessarily reflect the views of the National Science Foundation.

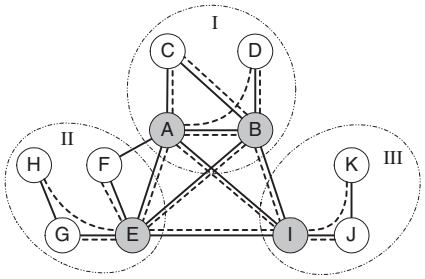


Fig. 1. An example of IBGP route reflection network. Routers are grouped into three clusters. Solid lines stand for physical links; dotted lines stand for IBGP sessions. Shaded nodes represent route reflectors.

Route reflection network, confederation and full mesh IBGP network are actually overlay networks on top of the underlying TCP/IP network. Each IBGP session is an overlay edge supported by a TCP connection. Some IBGP peers are not adjacent physically. They depend on IGP (Internal Gateway Protocol) to provide multi-hop communication. For example, in Fig. 1, the IBGP session between router A and D is routed along an IGP path through router B.

The reliable and stable IBGP operation is very important to the quality of Internet routing. When an IBGP session is lost, all related routes in the BGP routing tables have to be deleted. The route withdrawal messages may trigger huge amount of route recomputations and also result in route flaps and even unreachable network addresses. It takes a long time and lots of network resource to reestablish this session.

On the other side, the reliability and stability of IBGP operation also depend on the quality of underlying TCP and IGP routing. BGP sessions are sensitive to transport layer stability and routing layer reliability [4][5], especially the IBGP sessions which may cross multiple network hops.

However, the impact on the reliability of IBGP sessions and IBGP networks, which is from the IGP routing and the physical

layer, has not been well studied yet. Iannaccone et al. [6] show that link failures occur as part of everyday operation and about half of the failures last longer than one minute. Also, extremely high CPU utilization or memory shortage can cause a router to hang [7][8]. Most of these failures in physical networks are transient and can be recovered eventually. But, IGP routing service is temporarily interrupted, i.e., packet forwarding is stopped. Thus, BGP messages between some routers are lost until the IGP routing resumes. Though some rerouting-based fast IP recovery enhancement is proposed, aiming to decrease the recovery time to sub-second level, a notable fraction of routing failures are caused by the errors in software and equipment implementations and configurations, which makes IGP routing recovery time unexpectedly prolonged. For example, minute level (even greater than 10 minutes) routing outage is reported from a real measurement in an IP backbone [9][1]. This shows that long outage of IGP routing exists in practice, and we need to study its impact on time critical network protocols, such as BGP.

TCP provides reliable communication support to BGP by packet retransmissions. However, the delivery of BGP messages is delayed due to the network failure. Because any BGP router should receive at least one message from its peer in certain period of time (controlled by `Hold Timer`) to keep the session alive, the delayed message delivery may cause BGP session reset and those two BGP routers thus lose contact. Therefore, it is necessary to quantitatively study the BGP session reliability and its relations to BGP timer configurations and the recovery time of network failures. Also, TCP retransmits packets in an exponential backoff manner and the retransmission interval is up to 64 seconds. We find that, in some scenarios, this conservative behavior actually exacerbates the delay of BGP message delivery and thus makes IBGP sessions more fragile if network failures happen.

Furthermore, based on the results of IBGP session reliability, we can investigate the resilience of the whole IBGP network. In the route reflection or confederation IBGP networks, the full mesh of IBGP sessions is substituted by hierarchical structures to solve the scalability problem. However, we do not have a precise understanding on the impact of this transition with respect to network reliability. For example, does the network reliability become worse due to the smaller number of IBGP sessions? How much reliability can we gain by introducing one redundant reflector? And, moreover, is it possible to design a route reflection network even more reliable than the full mesh IBGP? Therefore, a proper reliability metric needs to be developed to evaluate the system resilience of IBGP networks.

The reliability analysis in IBGP networks is complicated due to the correlations among different IBGP sessions. For example, in Fig. 1, IBGP session between H and E is not statistically independent of the session between G and E, because they share one physical link. Cui et al. [10] give an approximate calculation on the probability that two overlay links fail simultaneously. In this paper, we perform a much more extensive study on the reliability of the overlay networks in IBGP. By using the dependent network failure analysis model [11] (based on independent failure-causing events), we investigate the reliability of IBGP networks and propose a novel metric for network resilience. This model can also be applied to other types of overlay networks.

The main contributions of this paper are: (1) We propose a reliability model for IBGP sessions, which gives quantitative relations between the session failure probability and the influential factors: BGP timers, network failure recovery time and TCP retransmission behaviors. Based on these results, we can configure BGP timer appropriately, so that BGP sessions can tolerate certain level of transient network failures. (2) A simple modification to TCP is presented, which can improve the robustness of IBGP sessions without introducing extra message overhead. (3) We propose a novel metric to measure the resilience of IBGP networks. We show that by using redundant reflectors or sessions appropriately, IBGP reflection networks can become considerably more reliable. Moreover, it can be more resilient than the traditional full mesh IBGP networks that do not have route reflection deployed. Therefore, our work provides the theory basis for further research on IBGP network optimization in terms of reliability.

The rest of the paper is organized as follows: In Section II, we define the network models and describe the reliability analysis framework for overlay networks. In Section III, we present the reliability models for IBGP sessions, validate the models by simulations, and give a modification to TCP to improve the robustness of IBGP sessions. In Section IV, the resilience metric for IBGP route reflection network is proposed, followed by case studies of route reflection network reliability. Section V concludes the paper.

II. System Model

A. Network Models and BGP Timers

We denote a typical physical network as graph $G(V, E)$, where V and E are the sets of routers and physical links, respectively. (u, v) represents the physical link from router u to router v. IGP path from u to v is denoted as P_{uv}, which is a sequence of routers and links on the path.

IBGP Route reflection network $G_r(V_r, E_r)$ is overlaid on top of the physical network, where V_r is the set of IBGP routers and E_r is the set of IBGP sessions. IBGP session between u and v is denoted by $\langle u, v\rangle$. IGP paths P_{uv} and P_{vu} are used to support this session. In a two-level route reflection network, E_r includes the full mesh of IBGP sessions among all reflectors and the sessions between clients and their reflectors. IBGP sessions between the clients in one cluster are optional, which can be used to improve the reliability of IBGP networks.

In order to detect misbehaving peers, every BGP router maintains a `KeepAlive Timer` and a `Hold Timer` for each BGP session it possesses. When the `KeepAlive Timer` expires, a `KEEPALIVE` message is sent to the

[1]The long delay is caused by the router oscillation between the 'faulty' state and the normal state without being detected by peer routers, and the failure to set 'Infinity Hippty Cost' bit during router reboot, which makes packets being forwarded to the rebooting router.

peer router associated with the session. When receiving a KEEPALIVE message or an UPDATE message, the Hold Timer is cleared. When an UPDATE message is sent out, the KeepAlive Timer is also cleared. If the Hold Timer expires, the BGP router assumes that the peer router can not respond correctly, and thus resets the BGP session.

Let us denote the period of KeepAlive Timer and Hold Timer as T_k and T_h, respectively. Thus, in any period of T_k, at least one message is sent to the peer; a BGP router expects to receive at least one message from the peer in any period of T_h. Any reasons, which cause BGP message delays or losses, may further lead to the related BGP session reset. In IETF RFC, the default timer values are: $T_h = 90$ seconds and $T_k = 30$ seconds.

B. Reliability Analysis Framework for IBGP Networks

In route reflection network $G_r(V_r, E_r)$, the failures of different edges are not necessarily independent. Two logical overlay sessions may share the same physical routers or links in their IGP paths. There are several approaches to study the network reliability with dependent component failures. In this paper, we make use of the cause-based reliability analysis model [11] in IBGP networks.

First, we identify all failure-causing events, each of which can cause some components in physical network $G(V, E)$ to fail. These failure events are independent to each other. The probability that an event happens can be derived from the historical network operation information. We define that the network is in a *failure state* when one or more than one failure-causing event happens. Moreover, we use $\mathcal{S}$ to denote the set of all network states, which includes all failure states and the state without any failure. F_s is the set of physical components that fail in state s ($s \in \mathcal{S}$), and $F_s \subseteq V \cup E$. Other components, which are not in F_s, work properly. The probability that state s happens is r_s, which can be easily calculated because the causing events are independent. It is obvious that $\sum_{s\in\mathcal{S}} r_s = 1$.

Second, in network failure state s, the failure probability of each IBGP sessions can be calculated. We denote p_{se} as the conditional probability that IBGP session e fails in network failure state s, i.e.,

$$p_{se} = P\,[\text{session } e \text{ fails} \mid F_s \text{ fails and other components are up}]$$

where $e \in E_r$, $s \in \mathcal{S}$, and F_s is the set of failed physical components in state s. p_{se} is related to BGP timers, network recovery time and TCP retransmission behaviors (its calculation will be shown in Section III). Also, in a given failure state, all components in physical network $G(V, E)$ are in deterministic states, and therefore the conditional failures of IBGP sessions are independent. Reliability analysis techniques in networks with independent failures can be applied. For each state, we can calculate the resilience of the IBGP network and denote it as R_s. We will show the definition of R_s in Section IV.

Third, the overall resilience of IBGP network is $\sum_{s\in\mathcal{S}} r_s R_s$.

Generally speaking, the number of the network failure states could be a very large number. However, in practice, we can get a satisfying statistical coverage (i.e., $\sum_{s\in\mathcal{S}} r_s$ is very close to 1) by only analyzing the most probable failure states. For instance, the possibility that multiple physical components fail simultaneously in one administrative domain is extremely small [12]. We can safely assume that at most one physical component fails at any time in one AS. The failed links or routers can be repaired before the next failure event takes place. Under this assumption, the number of network failure states is $|V|+|E|$. Note: some IP links may share one segment of fiber, and thus they may fail coincidently when the fiber is cut. In this scenario, we can include each single fiber-cut as a failure state in $\mathcal{S}$, and analyze the simultaneous failures of the related IP links. The same reliability model can thus be applied.

Table I summaries the major notations used in this section and the following sections.

TABLE I
TABLE OF NOTATIONS

(u, v)	Physical link from router u to router v
$\langle u, v\rangle$	IBGP session between router u and router v
$G(V, E)$	Physical network with router set V and link set E
$G_r(V_r, E_r)$	IBGP network with router set V_r and IBGP session set E_r
P_{uv}	IGP path from router u to router v
$\mathcal{S}$	Set of all network states
F_s	Set of failed components in state s
r_s	Probability that network state s occurs
p_{se}	Failure probability of session e in state s
q_{sv}	Probability that Hold Timer expires at router v in state s
R_s	IBGP resilience in state s
$\bar{R}_s$	IBGP resilience loss in state s $\quad(\bar{R}_s = 1 - R_s)$
T_h	Hold Timer expiration period
T_k	KeepAlive Timer expiration period
t_f	Occurrence time of network failures
T_c	IGP recovery time
i^*	The last admissible TCP retransmission
$t_r(i)$	The time of the i^{th} TCP retransmission
R_0	TCP retransmission timeout value
R_m	TCP maximum retransmission timeout value

III. RELIABILITY OF IBGP SESSIONS

In network failure states, IBGP sessions may be influenced by the failed physical components. In this section, we present a model to calculate the reliability of IBGP sessions in network failure state s and discuss how to tune network parameters and TCP retransmissions to increase the robustness of IBGP sessions. This model also provides a basis for IBGP network resilience analysis in Section IV.

Given that the components in F_s fail, session failure probability p_{se} is determined by the IGP paths used by e, TCP retransmission behavior and BGP timers. Also, the failures of different sessions are conditionally independent, because, in each network failure state, the states of physical components are deterministic and the correlations between IBGP sessions

are thus removed. The calculation of p_{se} can be divided into three cases as follows. Suppose session e is shared by router u and router v, i.e, $e = \langle u, v \rangle$. (1) If $u \in F_s$ or $v \in F_s$, then $p_{es} = 1$, i.e., an IBGP session definitely fails if its owner router fails. (2) If $F_s \cap P_{uv} = \emptyset$ and $F_s \cap P_{vu} = \emptyset$, then $p_{es} = 0$. That is, the IBGP sessions, that do not pass the failed components, will not be influenced. (3) Otherwise, IBGP session e could survive with certain probability, because the IGP routing could be recovered by either IP rerouting or physical layer repairing before the session is reset. In this scenario, p_{se} depends on the period (T_h) of `Hold Timer`, the period (T_k) of `KeepAlive Timer`, *IGP routing recovery time* T_c, and TCP retransmission behavior.

We have several comments on the recovery time T_c. IGP routing failure and recovery are complex processes influenced by protocol design, routing software and hardware implementations and configurations. To be precise, the recovery time is specifically determined in different failure scenarios. Thus, T_c should be taken as an average value for typical failure cases. Moreover, when we investigate the reliability of IBGP sessions and IBGP networks, T_c can represent possible degrees of network failures. For instance, BGP timers and reflection networks can be configured based on a T_c that could appear in the worst failure scenarios. We assume T_c is known for IBGP reliability analysis.

In the following parts of this section, we will focus on the analysis of p_{se} of the third scenario. Before that, we discuss the probability that a BGP `Hold Timer` expires.

A. *IBGP* `Hold Timer` *Expiration Probability*

Let us suppose the IGP path used by IBGP session e is $(u, \ldots, r_1, r_2, \ldots, v)$. In failure state s, the physical links between r_1 and r_2 fail, i.e., $F_s = \{(r_1, r_2), (r_2, r_1)\}$. After time T_c, the IGP routing between u and v recovers. Because of the delayed delivery of the `KEEPALIVE` message, the `Hold Timers` at u and v may expire. We denote their expiration probability as q_{su} and q_{sv}, respectively.

We only consider the `KEEPALIVE` messages between u and v in the following analysis. A typical packet transmission process, which is interfered by network failures, is shown in Fig. 2. Router u sends `KEEPALIVE` message $ka1$ to v successfully and receives the TCP acknowledgment $ack1$ after one round trip time RTT. Links (r_1, r_2) and (r_2, r_1) fail at time t_f. Message $ka2$, which is sent out at time T_k, is lost. TCP tries to recover the packet loss by retransmitting $ka2$. At time $t_f + T_c$, IGP routing recovers. u delivers $ka2$ finally at the third retransmission. Thus, since $ka1$ is received, it takes t_d for v to receive $ka2$. If there is no network failure at all, t_d should be around T_k, because `KEEPALIVE` messages are sent every T_k seconds. However, in a failure state, t_d is prolonged. If t_d is greater than T_h, the `Hold Timer` at v expires. Thus, $q_{sv} = P[t_d > T_h]$.

The IGP routing between u and v is interrupted from time t_f to $t_f + T_c$. t_f is a random variable at the time line. We divide the time line into the following intervals: $(H_1, H_2, H_3 \ldots)$, by the events that `KEEPALIVE` messages leave router r_2. Therefore, the network failures should occur in one of these intervals, which is marked by H in Fig. 2. According to renewal theory [13], t_f is distributed uniformly in H, because the length of each time interval ($|H_i|$) is a fixed value T_k. Next, we will calculate q_{sv} based on the distribution of t_f.

In the renewable interval H, in which the network failure happens, we further divide the time into two segments: A_r and A_l (as shown in Fig. 2). If $t_f \in A_r$, $ka1$ and $ack1$ are both delivered successfully; if $t_f \in A_l$, $ka1$ is received by v, but $ack1$ is lost in the network. We will analyze these two cases as follows.

(1) Case of A_r: This case is shown in Fig. 2. $ka1$ is sent out at time 0, it is delivered and acknowledged successfully. u sends $ka2$ out at time T_k. Due to the network failure, $ka2$ is lost in transit, and it is retransmitted by u. Since TCP guarantees the ordered packet delivery, no other BGP messages will be delivered to v before $ka2$. We focus on analyzing the retransmission of $ka2$.

In TCP implementations [14], the retransmission timeout (RTO) value R_0 is calculated as $\max(RTT + 4RTTVAR, minrto)$, where $RTTVAR$ is the variance of RTT estimation and $minrto$ is the minimum value of R_0. Packets are retransmitted in an exponential backoff manner, i.e., $RTO' = \min(2RTO, R_m)$, where R_m is the maximum retransmission timeout limit. The default values of $minrto$ and R_m are 1 second and 64 seconds.

Denote the time of the i^{th} retransmission of $ka2$ as $t_r(i)$. u begins the first retransmission at time $T_k + R_0$, i.e., $t_r(0) = T_k + R_0$. Based on the exponential backoff property, we have the equations for $t_r(i)$:

$$\begin{aligned} t_r(i) &= \sum_{k=1}^{i} \min(2^{k-1}R_0, R_m) + T_k \qquad (1) \\ &= \begin{cases} (2^i - 1)R_0 + T_k & : \text{ If } i \le \rho \\ (2^\rho - 1)R_0 + (i - \rho)R_m + T_k & : \text{ otherwise} \end{cases} \end{aligned}$$

where $\rho = 1 + \lfloor \log_2 \frac{R_m}{R_0} \rfloor$.

In order to avoid the `Hold Timer` expiration, v must receive $ka2$ before $T_h + RTT/2$. So, u must send out $ka2$ successfully before T_h. We call the retransmissions, which can arrive before the `Hold Timer` expires, the *admissible retransmissions*. In this case, the admissible retransmissions must be sent out before T_h. Let i^* be the last admissible retransmission iteration, i.e.,

$$i^* = \max\{i : t_r(i) \le T_h\} \qquad (2)$$

It is easy to derive i^* and $t_r(i^*)$ as follows:
If $T_h \le (2^\rho - 1)R_0 + T_k$, then

$$i^* = \left\lfloor \log_2 \left(\frac{T_h - T_k}{R_0} + 1 \right) \right\rfloor \qquad (3)$$

$$t_r(i^*) = T_k + (2^{i^*} - 1)R_0 \qquad (4)$$

otherwise,

$$i^* = \left\lfloor \frac{T_h - T_k - (2^\rho - 1)R_0}{R_m} \right\rfloor + \rho \qquad (5)$$

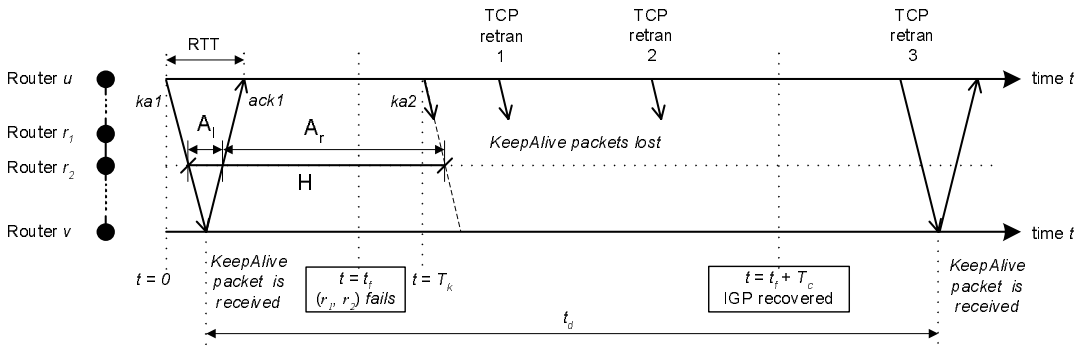


Fig. 2. A typical case of transmitting KEEPALIVE messages when network failures occur.

$$t_r(i^*) = (i^* - \rho)R_m + (2^\rho - 1)R_0 + T_k \quad (6)$$

Based on the concept of the last admissible retransmission, we have the following lemma.

Lemma 1: If $t_f \in A_r$, then the probability of the Hold Timer expiration at v is

$$q_{sv}^{A_r} = \begin{cases} 0 & : \text{ if } T_c \le t_r(i^*) - T_k - d_{ur_2} \\ 1 & : \text{ if } T_c \ge t_r(i^*) - |A_l| - d_{ur_2} \\ \frac{T_c + T_k + d_{ur_2} - t_r(i^*)}{|A_r|} & : \text{ else} \end{cases} \quad (7)$$

where d_{ur_2} is the delay of IGP path P_{ur_2}. $|A_r|$ and $|A_l|$ are the lengths of time interval A_r and A_l, respectively.

Proof: The expiration of the Hold Timer at v is equivalent to the fact that the last admissible retransmission is sent before the recovery of IGP routing, i.e., $t_r(i^*) < t_f + T_c$. Therefore, $q_{sv} = P[t_f > t_r(i^*) - T_c]$. Because t_f is uniformly distributed in the interval of A_r,

$$q_{sv} = \frac{T_k + d_{ur_2} - [t_r(i^*) - T_c]}{|A_r|}$$

By limiting the value q_{sv} into $[0, 1]$ interval, we obtain Equation 7. ■

(2) Case of A_l: If $t_f \in A_l$, $ka1$ is received by v, but $ack1$ is lost due to the network failure. Thus, $ka1$ is not successfully delivered in the view of router u. u begins to retransmit $ka1$ at time R_0. We can calculate the retransmission time sequence as follows:

$$t_r(i) = \begin{cases} (2^i - 1)R_0 & : \text{ If } i \le \rho \\ (2^\rho - 1)R_0 + (i - \rho)R_m & : \text{ otherwise} \end{cases}$$

where $\rho = 1 + \lfloor \log_2 \frac{R_m}{R_0} \rfloor$.

Similarly, an admissible retransmission is the retransmission that can arrive before the Hold Timer expires. In this case, the admissible retransmission of $ka1$ must be sent before $T_h - RTT$, so that $ka2$ can be delivered in time. Thus, the last admissible retransmission $i^* = \max\{i : t_r(i) \le T_h - RTT\}$ and we have following equations for $t_r(i^*)$:

If $T_h \le (2^\rho - 1)R_0 + RTT$, then

$$i^* = \left\lfloor \log_2 \left(\frac{T_h - RTT}{R_0} + 1 \right) \right\rfloor \quad (8)$$

$$t_r(i^*) = (2^{i^*} - 1)R_0 \quad (9)$$

otherwise,

$$i^* = \left\lfloor \frac{T_h - (2^\rho - 1)R_0 - RTT}{R_m} \right\rfloor + \rho \quad (10)$$

$$t_r(i^*) = (i^* - \rho)R_m + (2^\rho - 1)R_0 \quad (11)$$

The following lemma gives the probability that the Hold Timer expires in the case of A_l.

Lemma 2: If $t_f \in A_l$, then the probability of the Hold Timer expiration at v is

$$q_{sv}^{A_l} = \begin{cases} 0 & : \text{ if } T_c \le t_r(i^*) - |A_l| - d_{ur_2} \\ 1 & : \text{ if } T_c \ge t_r(i^*) - d_{ur_2} \\ 1 - \frac{t_r(i^*) - T_c - d_{ur_2}}{|A_l|} & : \text{ else} \end{cases} \quad (12)$$

where d_{ur_2} is the delay of IGP path P_{ur_2}.

Proof: Similar to Lemma 1, $q_{sv} = P[t_f > t_r(i^*) - T_c]$. Because t_f is uniformly distributed in the interval of A_l,

$$q_{sv} = \frac{|A_l| + d_{ur_2} - [t_r(i^*) - T_c]}{|A_l|}$$

By limiting the value of q_{sv} into $[0, 1]$ interval, we obtain Equation 12. ■

Because t_f is uniformly distributed in the interval H, we combine the results from previous two cases to obtain the probability of the Hold Timer expiration as follows.

$$q_{sv} = (|A_l| q_{sv}^{A_l} + |A_r| q_{sv}^{A_r})/T_k \quad (13)$$

The calculation of q_{sv} can be simplified when RTT is small, especially in an intradomain backbone network, where the IBGP overlay network is applied. Compared with T_h and T_k (the default values are 30 seconds and 90 seconds, respectively), RTT, $|A_l|$ and d_{ur_2} are small enough and can be ignored. Thus, the case of A_l does not have significant influence on the calculation of IBGP session reliability. We can thereafter simplify the calculation of q_{sv}.

$$q_{sv} = \begin{cases} 0 & : \text{ if } T_c \le t_r(i^*) - T_k \\ 1 & : \text{ if } T_c \ge t_r(i^*) \\ 1 - \frac{t_r(i^*) - T_c}{T_k} & : \text{ else} \end{cases} \quad (14)$$

Our simulation results show that the simplification obtains satisfiable precision when RTT is small.

B. IBGP Session Reliability

IBGP session e (shared by u and v) is terminated, if any of the two `Hold Timers` associated with e expires. Let us suppose both u and v send `KEEPALIVE` message to each other at the same period of T_k. Without the loss of generality, the time, when u sends out the message in each period, is θ seconds earlier than that of v, where θ is the phase difference and $\theta \in [0, T_k)$. Fig. 3 shows the time sequence of sending `KEEPALIVE` messages from u and three corresponding sequences of v based on different values of θ. Each vertical arrow represents one sending of `KEEPALIVE` message. For the `Hold`

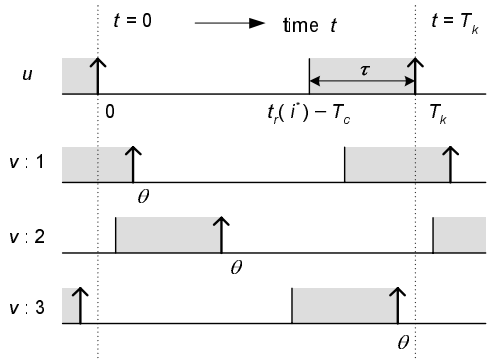


Fig. 3. Time sequence of sending `KEEPALIVE` messages.

`Timer` expiration probability, we only consider the case of A_r, which dominantly influences the calculation. Thus, in the renewable time interval $[0, T_k)$, if the network failure happens in the shaded region on the time line of either u or v, IBGP session e will fail. Because the shaded regions are influenced by θ, the session failure probability p_{se} is the function of θ, written $p_{se}(\theta)$. By analyzing the three scenarios in the figure, we can derive the formula for $p_{se}(\theta)$ as follows.

$$p_{se}(\theta) = \frac{\tau + \min(\theta, T_k - \tau) - \max(0, \theta - \tau)}{T_k} \tag{15}$$

where τ is the length of a shaded region and $\tau = T_k - t_r(i^*) + T_c$.

`KeepAlive` timers are reset whenever a `KEEPALIVE` or `UPDATE` message is sent out. Because u and v send `UPDATE` packets independently and θ changes randomly. We can safely model θ by the uniform distribution in interval $[0, T_k)$. Thus, we have the following theorem for IBGP session reliability.

Theorem 1: Let us assume that IGP routing between node u and v is interrupted in network failure state s. T_c is the routing recovery time after the failure. If the communication latency between u and v is small enough, then the failure probability of IBGP session $\langle u, v \rangle$ is

$$p_{se} = \begin{cases} 0 & : \text{ if } T_c \le t_r(i^*) - T_k \\ 1 & : \text{ if } T_c \ge t_r(i^*) \\ 1 - \left[\frac{t_r(i^*) - T_c}{T_k}\right]^2 & : \text{ else} \end{cases} \tag{16}$$

where T_k and T_h are the expiration periods of the `KeepAlive Timer` and `Hold Timer`, respectively. $t_r(i^*)$ is the time of the last admissible retransmission, which can be calculated from Equations 3-6.

Proof: When RTT is small, Equation 15 can be used to calculate the session failure probability. Because θ is uniformly distributed in $[0, T_k)$,

$$p_{se} = \frac{1}{T_k} \int_0^{T_k} p_{se}(\theta)\, \mathrm{d}\theta$$

By plugging in $p_{se}(\theta)$ and simplifying the above equation, we get the IBGP session failure probability in Equation 16. ■

The above theorem (Equation 16) shows that the `Hold Timers` at u and v expire independently under the assumption that the phase difference is uniformly distributed. Thus, p_{se} can also be calculated as follows.

$$p_{se} = q_{su} + q_{sv} - q_{su} q_{sv} \tag{17}$$

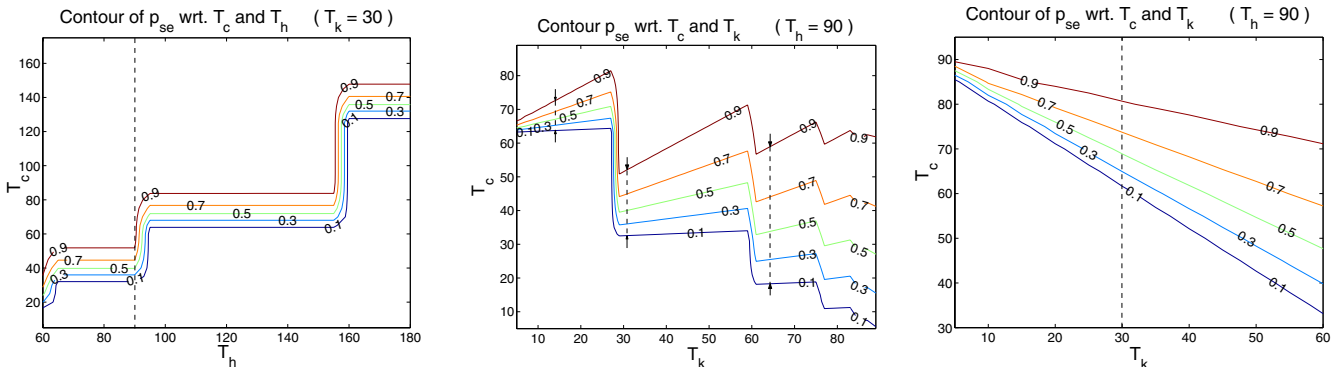
This equation is also applicable to the scenario where RTT is very large and thus the influence of A_l case is non-negligible, though in most of scenarios of IBGP networks, Equation 16 can be applied with satisfying precision.

Theorem 1 is derived based on the single link failures. It is also applicable to the failures of multiple links or routers. T_c can be obtained from the historical network operation information. Furthermore, p_{se} can also be viewed as a function of T_c, standing for the robustness of IBGP sessions that are subjected to various routing service interruptions.

Next, we investigate some characteristics of the IBGP session reliability, as well as the influences from BGP timers and IGP recovery time. In our numerical experiments, the failure probabilities of IBGP sessions (p_{se}) are calculated by the formulas derived above. The round trip time between node u and v is 40 milliseconds. $R_0 = 1$ second and $R_m = 64$ seconds.

In Fig. 4(a), the contour of p_{se} as a function of T_h and T_c is presented, where $T_k = 30$ seconds. The curves are the level set of p_{se}. This figure shows that a larger T_h or a smaller T_c results in a lower session failure probability, which is intuitively correct. In the default BGP timer configurations ($T_k = 30$ seconds and $T_h = 90$ seconds, as indicated by the dotted line in the figure), when T_c increases from 33 to 55 seconds, p_{se} changes from 0.1 to 0.96. If we want to prevent session failures, IGP path recovery process has to finish in less than 31 seconds and we call this time interval the *repairing window*. The interesting thing in this figure is that the curves of p_{se} exhibit a 'staircase' pattern behavior, i.e., the increase of T_h may lead to invariable p_{se}. This is because p_{se} is determined by $t_r(i^*)$, when T_c and T_k are fixed. Due to the time interval between two consecutive TCP retransmissions, the increase of T_h may not change the time of the last admissible retransmission $t_r(i^*)$. For example, when $T_h \in [93, 157)$, $t_r(i^*)$ is fixed at 93 seconds. Therefore, p_{se} does not change, if the time of the last admissible retransmission keeps constant.

The impact of T_k on the IBGP session failure probability is shown in Fig. 4(b), where $T_h = 90$ seconds and the contour of p_{se} with respect to T_c and T_k is displayed. The middle section



(a) Contour of p_{se} wrt. T_c and T_h, when $T_k = 30$.

(b) Contour of p_{se} wrt. T_c and T_k, when $T_h = 90$.

(c) Contour of p_{se} in aggressive packet retransmission, when $T_h = 90$.

Fig. 4. IBGP session failure probability. (Time unit: second)

of p_{se}, where $0 < p_{se} < 1$ (shown in the figure as the dotted lines), narrows when T_k decreases (i.e., the contour curves are closer to each other at smaller T_k). The intuitive explanation for this characteristic is that the decrease of T_k shrinks the length of the renewable interval H and thus reduces the randomness of the system. When T_k is small enough, the fate of IBGP sessions becomes deterministic. In addition, we can observe from the figure that p_{se} is not a monotonic function of T_k. Equation 1 shows that the retransmission time sequence $(\{t_r(i)|i \geq 0\})$ is shifted as we change T_k. This shift action could increase or decrease $t_r(i^*)$, and thus the session failure probability may become larger or smaller. But, in a large scope, a smaller T_k leads to more reliable IBGP sessions.

C. Aggressive Packet Retransmission

The reliability of an IBGP session is influenced by the retransmission behavior of TCP. TCP retransmits packet in a conservative way (exponential backoff) and the time interval between two retransmissions is up to 64 seconds. This behavior may miss some critical opportunities to get BGP message delivered in time. Thus, it is possible to improve the session robustness by using different transport protocols. As an extreme case, when the packet loss is detected, an aggressive transport protocol can retransmit the lost packet as frequent as possible, so that the packet is guaranteed to be delivered successfully as soon as the IGP paths are recovered. Thus, in the scenario of aggressive retransmission, $t_r(i^*) = T_h$. From Equation 16, the IBGP session failure probability can be calculated as follows:

$$p_{se} = \begin{cases} 0 & : \text{ if } T_c \leq T_h - T_k \\ 1 & : \text{ if } T_c \geq T_h \\ 1 - \left[\frac{T_h - T_c}{T_k}\right]^2 & : \text{ else} \end{cases} \quad (18)$$

Obviously, the method of aggressive packet retransmission is not practical, because it incurs too much message overhead. However, it provides the lower bound for the IBGP session failure probability, i.e., the highest robustness that can be achieved by any transport layer protocol. Fig. 4(c) shows the contour of p_{se}, where the aggressive packet retransmission is used. First, different from Fig. 4(b), p_{se} in Fig. 4(c) decreases monotonously as T_k decreases. Because of aggressive packet retransmission, $t_r(i^*)$ is fixed at T_h, and thus p_{se} is a monotonic function of T_k. Second, the session reliability is much improved due to the aggressive packet retransmission. In the default BGP timer configurations (indicated by the dotted line in Fig. 4(c)), when T_c increases from 62 to 84 seconds, p_{se} changes from 0.12 to 0.96. The repairing window is 60 seconds, which is almost twice as large as that of normal TCP (31 seconds) in Fig. 4(a). This shows that the conservative retransmission behavior of TCP does exacerbate the reliability of IBGP sessions when IBGP network is in a failure state.

In next section, we will present a simple modification of TCP, which achieves the lowest IBGP session failure probability without introducing extra message overhead.

D. Improving Robustness of IBGP Sessions

Three controllable factors influence IBGP session reliability: `Hold Timer`, `KeepAlive Timer` and TCP retransmission behavior. `Hold Timer` expiration period T_h is set to detect unhealthy BGP peers. Equation 18 demonstrates that a larger T_h enables BGP sessions to tolerate longer IGP routing interruptions. However, it also makes BGP routers insensitive to detect misbehaving peers. Similarly, a very small T_k (the `KeepAlive Timer` expiration period) can reduce session failure probability, but it may incur a large amount message overhead. Therefore, we have to make a compromise between the IBGP session reliability and other issues when adjusting the values of T_h and T_k. Our model can therefore be used as a reference to IBGP session reliability in various circumstances. For example, Fig. 4(c) can be used to choose an appropriate T_k so that the IBGP session can tolerate certain level of network failures, when T_h is fixed at 90 seconds.

TCP retransmission behavior can also be tuned to make IBGP sessions more robust against transient network failures. From Equation 16, p_{se} is a decreasing function of the last admissible retransmission time $t_r(i^*)$. If $t_r(i^*)$ reaches its maximum value T_h, the session failure probability p_{se} is

minimized, which is equivalent to the scenario of the aggressive packet retransmission. Thus, in order to improve the session robustness, we can regulate $t_r(i^*)$ by modifying the TCP implementations on BGP routers as follows, so that $t_r(i^*)$ is as close to T_h as possible.

We enforce a TCP retransmission right before the `Hold Timer` expires at the peer BGP router, which is T_h seconds after the successful delivery of the previous BGP message. In TCP, the time of receiving the ACK (including acknowledgments for both `KEEPALIVE` and `UPDATE` messages) from the peer router is recorded as t_p. Then, the time interval between the k^{th} and the $(k-1)^{th}$ retransmissions is $\min(rto, T_h + t_p - RTT - t_r(k-1))$, where rto is the retransmission timeout value in the original TCP, i.e., $rto = \min(2^{k-1}R_0, R_m)$.

By the above modifications of TCP, the `KEEPALIVE` packet can be delivered right before the `Hold Timer` expires, if IGP routing is recovered in time. The shortcoming of the large retransmission timeout in TCP is thus avoided. Moreover, because $t_r(i^*)$ is controlled to be T_h, IBGP session failure probability can be calculated using Equation 18. Therefore, we achieve the performance of the aggressive packet retransmission, without extra message overhead. We will show the improvement on IBGP session reliability by simulations in the next section.

E. Validation of IBGP Session Reliability Models

In this section, we validate the previous reliability models by simulations. The simulator, SSFNET [15] with BGP4 implementation of version 1.4.15, is used in our experiments. Four nodes in an AS, u, r_1, r_2 and v, are connected in sequence and each physical link has 20 millisecond propagation delay. IBGP session e is set up between u and v. We use default BGP timer parameters: $T_k = 30$ seconds and $T_h = 90$ seconds. In order to test the failure probability of session e, we inject network failures by bringing down the link between r_1 and r_2 at a random time in simulations. The link is recovered T_c seconds after the failure. T_c ranges from 30 to 95 seconds. For each value of T_c, we iterate the fault injection process 1000 times. By counting the times of `Hold Timer` expirations and IBGP session failures, we can obtain the percentages of the timer expiration and session failures, which stand for the simulated results of q_{sv} and p_{se}.

Fig. 5(a) compares the expiration probabilities of the `Hold Timer` from the simulation results and from our analytical models. Because the most often used retransmission timeout (corresponding to R_0 in the analytical model) in SSFNET TCP is 1.5 seconds for a small round trip time, we let R_0 equal 1.5 in the analytical models. The figure shows that the timer expiration probability given by the simulations is quite close to the results from the analytical models (Equation 13). Also, the simplified model (Equation 14), which only takes the case of A_r into calculation, generates almost the same results as the exact model does, because the round trip time is very small compared to the timer parameters.

Fig. 5(b) validates the model of IBGP session failure probability. p_{se} is calculated using Equation 16. The 'Simulation Results I' is obtained with normal TCP. Its value is a little greater than that of the analytical model. This is because, in simulation, R_0 alternates between 1.5 and 2.0 seconds due to the estimations of RTT and its variance, but, in the analytical model, R_0 is fixed at 1.5. Thus, some difference exists. In order to show that R_0 is the reason which causes the difference, we fix R_0 in TCP implementation to be 1.5 and get 'Simulation Results II' which conforms to the analytical model very well. This experiment shows that R_0 in the analytical model needs to be calibrated based on the specific TCP implementations to make the analysis result more precise. Usually, $minrto$ is 1 second and RTT is small values in backbone network. Thus, we can choose R_0 to be $1 - 2$ seconds.

The modified TCP, described in Section III-D, is implemented in the SSFNet simulator. We keep track of the last acknowledgment time t_p and modify the packet retransmission code in TCP. The simulation results are shown in Fig. 5(c), as well as the results from the aggressive packet retransmission and the original TCP. Our modification of TCP (by enforcing a retransmission right before `Hold Timer` expires) significantly increases the robustness of IBGP sessions. It achieves almost the same results as the aggressive packet retransmission, but no extra message overhead is incurred. In original TCP, the repairing window is 30 seconds; while, in the modified version, it is improved to 60 seconds. The length of the repairing window is doubled due to the modification.

Furthermore, in Fig. 5(c), the curves are different only in the interval $[30, 90]$ of T_c. This means that only if T_c falls into interval $[t_r(i^*) - T_k, T_h]$, our modification on TCP or aggressive retransmission can improve the reliability of IBGP sessions; in other range of T_c, the failure probabilities are uniformly 0 or 1, no matter if the extra retransmissions are used.

F. Extension for `UPDATE` *Messages*

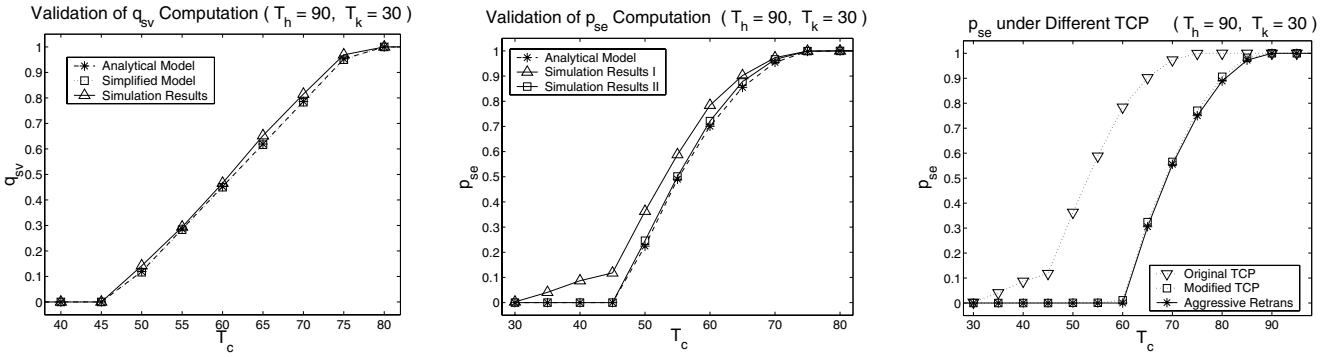
Our previous analysis only considers `KEEPALIVE` messages. In this section, we extend the model to include `UPDATE` messages. We assume that arrivals of the `UPDATE` messages from u to v and from v to u are both Poisson processes with rate λ.

By combining the `KEEPALIVE` and `UPDATE` messages, we get the cumulative distribution function of the inter arrival time T_a of BGP messages from u to v.

$$F_{T_a}(t) = \begin{cases} 1 & : \text{ if } t \geq T_k \\ 1 - e^{-\lambda t} & : \text{ if } 0 \leq t < T_k \end{cases} \tag{19}$$

By applying renewal theory, the density function of failure time t_f is as follows.

$$\begin{aligned} f_{t_f}(t) &= (1 - F_{T_a}(t)) / E[T_a] \\ &= \begin{cases} 0 & : \text{ if } t \geq T_k \\ \frac{\lambda e^{-\lambda t}}{1 - e^{-\lambda T_k}} & : \text{ if } 0 \leq t < T_k \end{cases} \end{aligned} \tag{20}$$



(a) `Hold Timer` expiration probability. (b) IBGP session failure probability. (c) Improving IBGP session reliability.

Fig. 5. Validation of IBGP session reliability models by SSFNet BGP Simulator. (Time unit: second)

Our previous TCP modification also works in the scenario where `UPDATE` messages are considered. That is, $t_r(i^*) = T_h$ and the IBGP session failure probability can reach its lower bound by enforcing one retransmission right before `Hold Timer` expires. Similar to Theorem 1, we can derive the theorem for p_{se} based on the distribution of t_f. Because of the space limitation, we skip the proof.

Theorem 2: If the arrival of `UPDATE` messages related to BGP session e is Poisson process and the rate is λ, the IBGP session failure probability in the modified TCP implementation (in Section III-D) is as follows.

$$p_{se} = \begin{cases} 0 & : \text{ if } T_c \le T_h - T_k \\ 1 & : \text{ if } T_c \ge T_h \\ 1 - \left[\frac{1-e^{-\lambda(T_h - T_c)}}{1-e^{-\lambda T_k}}\right]^2 & : \text{ else} \end{cases} \quad (21)$$

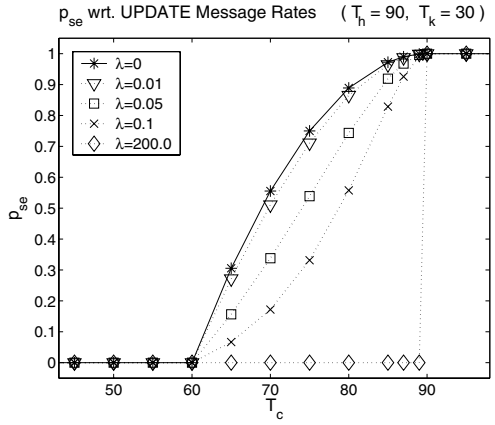


Fig. 6. Impact of `UPDATE` messages on IBGP session failure probability with the modified TCP. (Time unit: second)

Fig. 6 shows the IBGP session failure probability as a function of the rates of `UPDATE` messages. When $\lambda = 0$, it is equivalent to the previous case without taking `UPDATE` messages into consideration. When λ increases, the session becomes more robust. When λ is large enough, p_{se} is either 0 or 1. Note that the rate of `UPDATE` messages does not influence the repairing window.

IV. Reliability of IBGP Networks

In previous discussions, we have presented the reliability analysis framework for IBGP networks and the models for IBGP session reliability. In this section, based on those results, we complete the IBGP reliability model by giving a metric to characterize the resilience of the whole IBGP networks. Though our methods focus on route reflection IBGP networks (and the full mesh IBGP), it can be applied to the confederation IBGP networks with very few modifications.

A. IBGP Network Resilience R

Network resilience means that a network system can still perform some part of functions, even when failures happen. The function of IBGP network is to distribute the external routing information, which is learned from outside of the AS, to all other IBGP routers in this domain after necessary processing. The loss caused by IBGP session failures is the invalidation of the routing entries, which are directly or indirectly related to those failed IBGP sessions, and the consequent route flaps or unreachable network addresses. We use w_i to denote the number of external routes that are obtained by router i from its EBGP peers and are further injected into the IBGP network.

In a healthy network, these w_i routes from router i can potentially[2] be advertised to all other IBGP routers directly or through reflectors, and these routes are involved in the path selection process of the local router. On the other hand, if some IBGP sessions fail, the related routes are invalidated and withdrawn.

In network failure state s, if router i can not exchange routing information with router j directly or indirectly due to IBGP session failures, i and j are *isolated* from each other and we denote this relation as $i \overset{s}{\nleftrightarrow} j$. Likewise, if i and j are reachable to each other in failure state s, it is denoted as $i \overset{s}{\leftrightarrow} j$. Please note that two routers may be isolated from each other even if they are connected in the route reflection graph. For example, if session $\langle A, D\rangle$ fails in Fig. 1, A and D are isolated. Though A and D both share an IBGP session with

[2]Some routes may be filtered out because of path selection policies of reflectors.

B, B does not reflect routes between A and D, because B is in the same cluster as A does. CLUSTER_LIST loop detection policy [2] prevents this type of route reflection.

In network failure states, if i and j are isolated from each other, router i loses w_j routes which were previously advertised by j; similarly, j loses w_i routes. Thus, the loss of IBGP function can be quantified as $w_i + w_j$. In the full mesh IBGP network, any two routers communicate directly to each other using a dedicated IBGP session. If session $\langle i, j\rangle$ fails, only i and j are isolated from each other and the loss is $w_i + w_j$. In a route reflection network, the routers are organized hierarchically. External route information may be reflected multiple times until it arrives at a destination router. The termination of one IBGP session may cause multiple pairs of IBGP routers to lose contact. Thus, the loss of IBGP function may be severe. For example, in Fig. 1, the external routing information learned by router H has to be reflected by E, by I, and then it can be received by J. If IBGP session $\langle H, E\rangle$ fails, H is isolated from any other BGP routers in the domain. Also, the failures of different IBGP sessions or routers have different impacts on IBGP operation. For instance, the termination of session $\langle A, B\rangle$ only makes A and B isolated from each other. However, if session $\langle E, I\rangle$ breaks, the routers in cluster II and III are isolated.

In a general IBGP network, based on the above descriptions about IBGP function loss, we define the *network resilience* in state s as R_s, and

$$R_s = \frac{\sum_{i,j\in V_r}(w_i + w_j)Pr[i \overset{s}{\leftrightarrow} j]}{\sum_{i,j\in V_r}(w_i + w_j)} \tag{22}$$

The network *resilience loss* $\bar{R}_s$ is

$$\bar{R}_s = 1 - R_s = \frac{\sum_{i,j\in V_r}(w_i + w_j)Pr[i \overset{s}{\nleftrightarrow} j]}{\sum_{i,j\in V_r}(w_i + w_j)} \tag{23}$$

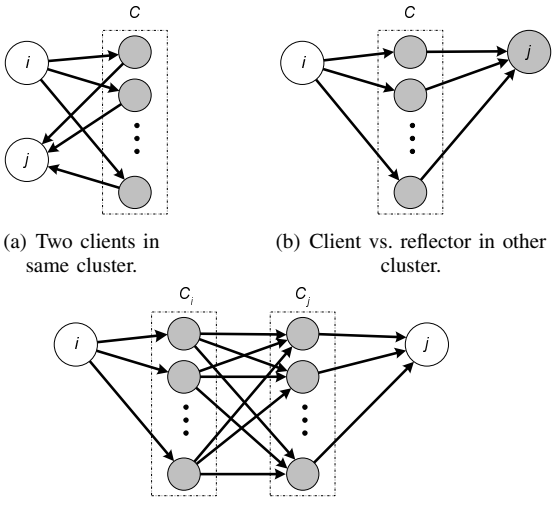
Therefore, the resilience over the entire state space is $R = \sum_{s\in\mathcal{S}} r_s R_s$, and $\bar{R} = \sum_{s\in\mathcal{S}} r_s \bar{R}_s$, where r_s is the probability that the network is in state s. It is easy to verify the following facts: $0 \le R, \bar{R} \le 1$ and $R + \bar{R} = 1$.

The IBGP resilience defined above is related to the network resilience studied by Colbourn in [16]. The network resilience in [16] is the expected number of node pairs that can communicate in a network of independent failures. There are three major differences between this definition and our IBGP resilience: (1) Because the session failures in IBGP networks are not independent, we calculate IBGP resilience in each network state and summarize them together based on the probability of each state. (2) We make a weighted summation of node pairs by the number of the introduced EBGP routes. (3) We normalized the summation into a value between 0 and 1, so that the IBGP resilience of different IBGP networks can be compared. A larger R or a smaller $\bar{R}$ means that the IBGP network is more robust. For convenience, in the following discussion, we focus on the resilience loss $\bar{R}$.

B. Calculation of IBGP Network Resilience

In order to calculate the resilience loss $\bar{R}$, parameters r_i and w_i can be obtained from historical network operation information. Moreover, we need to obtain the isolation probability $Pr[i \overset{s}{\nleftrightarrow} j]$ for any pair of nodes, i and j, in any failure state s. This probability calculation can be reduced to the s,t-*connectedness* problem [17] in the directed acyclic graph G^s_{ij} which has perfect nodes and independent edge failures. The s,t-connectedness problem aims to compute the probability that at least one path from i to j does not fail in the directed graph.

G^s_{ij} is generated based on the route reflection graph $G_r(V_r, E_r)$ with the following three modifications: (1) The IBGP routers that are failed in network failure state s and the IBGP sessions they possess are removed from V_r and E_r. (2) The edge failure probability in G^s_{ij} is the IBGP session failure probability which can be calculated by the models in Section III. (3) Irrelevant edges are deleted and the directions of the remaining edges are determined so that all paths from i to j in G^s_{ij} are valid route advertising paths in the reflection network. The valid advertising paths are explained as follows. In a route reflection network, if the routing information is sent from a client to its reflector, we define this advertising relationship as C-R; similarly, R-C and R-R stand for sending routing information from a reflector to its client and from a reflector to its peer reflector in different clusters, respectively. Thus, according to IETF RFC [2], the valid route advertising path is the subsequence or the whole of the following sequence: C-R $\Rightarrow$ ... C-R $\Rightarrow$ R-R $\Rightarrow$ R-C... $\Rightarrow$ R-C. Next, we will show how to calculate the isolation probability of i and j, i.e., $Pr[i \overset{s}{\nleftrightarrow} j]$, in a two-level route reflection network in detail.



(a) Two clients in same cluster.

(b) Client vs. reflector in other cluster.

(c) Two clients in different clusters.

Fig. 7. G^s_{ij} for calculating $Pr[i \overset{s}{\nleftrightarrow} j]$.

In a two-level route reflection network, the routing information can be reflected at most twice. We thus divide the calculations into the following three cases based on the relationship between router i and router j. For convenience of

explanation, we denote the failure probability of IBGP session $\langle u, v\rangle$ in failure state s simply as p_{uv}.

First, if both i and j are reflectors or i is a client of the reflector j, because of route reflection rules and CLUSTER_LIST loop detection, the routes from i can not be reflected to j by any other reflector and vice versa. Thus, $Pr[i \overset{s}{\nleftrightarrow} j] = p_{ij}$.

Second, if i and j are clients in the same cluster or i is a client and j is a reflector in other cluster, graphs G^s_{ij} of these two scenarios are shown in Fig. 7(a) and Fig. 7(b), respectively. There are $|C|$ independent paths from i to j, where C is the set of reflectors in the cluster of i. Thus,

$$Pr[i \overset{s}{\nleftrightarrow} j] = \prod_{c \in C} (p_{ic} + p_{cj} - p_{ic}p_{cj}) \tag{24}$$

Third, if i and j are clients in different clusters, graph G^s_{ij} of this scenario is shown in Fig. 7(c). C_i and C_j are the sets of reflectors in the cluster of i and j, respectively. There are $|C_i||C_j|$ different paths from i to j. Thus, $Pr[i \overset{s}{\nleftrightarrow} j]$ is the probability that all these paths fail. However, because these paths are not independent, the probability calculation could be a difficult problem. In general, the following lemma shows that it is unlikely to find efficient solutions to calculate the isolation probability of this scenario.

Lemma 3: If i and j are clients in different clusters, the problem of computing $Pr[i \overset{s}{\nleftrightarrow} j]$ is $\#\mathbf{P}$-complete.

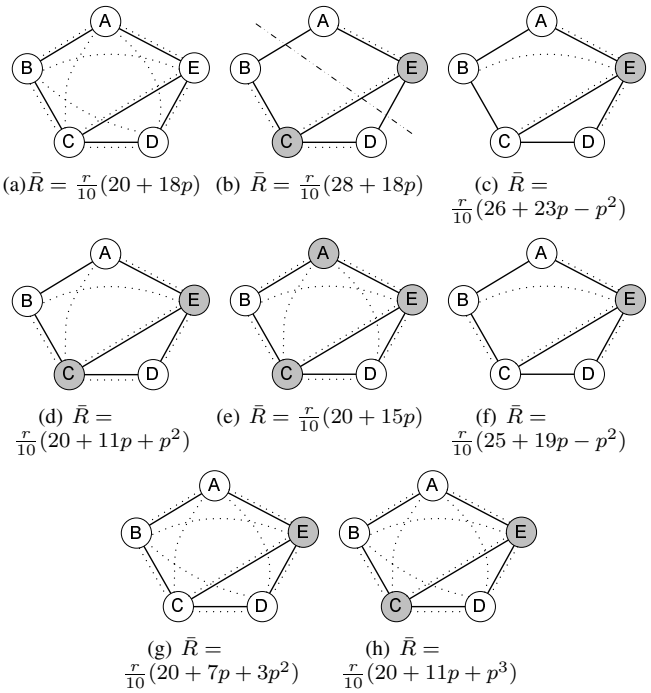
Proof Sketch: The problem of computing $Pr[i \overset{s}{\nleftrightarrow} j]$ is equivalent to find the probability that all paths from i to j fail in graph G^s_{ij} (shown in Fig. 7(c)). Though C_i and C_j form a complete bipartite, it is a more general case than a general bipartite, because if one edge does not exist, the corresponding edge in G^s_{ij} can have 1 as the failure probability. Thus, the reduced result of Corollary 3.4 in chapter 3.2 of [17] by using the proof technique of Theorem 3.2 is a special case of G^s_{ij}. Bipartite Independent Set problem, which is $\#\mathbf{P}$-complete, can be reduced to the problem of computing $Pr[i \overset{s}{\nleftrightarrow} j]$, and therefore the result in the lemma follows. ■

In practice, the number of IBGP sessions that have nonzero failure probabilities in a failure state is small. The number of redundant reflectors ($|C_i|$ and $|C_j|$) is also quite limited. If the sessions with zero failure probabilities contain a path from i to j in G^s_{ij}, then $Pr[i \overset{s}{\nleftrightarrow} j] = 0$; otherwise, the isolation probability can be computed fast enough by traditional network reliability analysis methods, such as the factoring algorithms [18].

C. Case Studies - functional reliability analysis

In this section, we perform a functional reliability analysis on eight IBGP networks which are overlaid on top of the same physical network $G(V, E)$. The functional reliability analysis means to analyze the reliability of the IBGP network in which the failure probabilities of all components (including IBGP sessions) are uniform. Let us denote p_f as the failure probability of all physical components (routers and physical links) and denote p as the failure probability of the influenced IBGP sessions in all network failure states (calculated by Equation 16). Though our reliability model of IBGP networks allows dependent physical component failures, in order to simplify the following analysis, we assume that each physical component fails independently. We also consider that single router failure and single physical link failure have enough statistical coverage on the space of failure states. Thus, the total number of failure states is $|V| + |E|$ and the probability that the network is in one failure state is $r = p_f(1-p_f)^{|V|+|E|-1}$. Note that r is only determined by the physical network. Every IBGP router receives identical number of external routes from its EBGP peers and we normalize each w_i to 0.5 .

Fig. 8 shows eight IBGP reflection networks and their resilience losses $\bar{R}$. Solid lines represent physical links, dotted lines represent IBGP sessions, and the shaded nodes represents reflectors. The resilience of the IBGP networks is determined by the configuration of route reflection, such as the number of clusters, redundant reflectors, redundant IBGP sessions, etc. We discuss their impacts on the IBGP reliability as follows.



(a) $\bar{R} = \frac{r}{10}(20 + 18p)$ (b) $\bar{R} = \frac{r}{10}(28 + 18p)$ (c) $\bar{R} = \frac{r}{10}(26 + 23p - p^2)$

(d) $\bar{R} = \frac{r}{10}(20 + 11p + p^2)$ (e) $\bar{R} = \frac{r}{10}(20 + 15p)$ (f) $\bar{R} = \frac{r}{10}(25 + 19p - p^2)$

(g) $\bar{R} = \frac{r}{10}(20 + 7p + 3p^2)$ (h) $\bar{R} = \frac{r}{10}(20 + 11p + p^3)$

Fig. 8. Eight IBGP route reflection networks and resilience losses $\bar{R}$. $r = p_f(1-p_f)^{10}$, and r is a small value. p is from Equations 16.

In each of the eight cases, we need to compute resilience loss $\bar{R}_s$ for every failure state (by Equation 23). For example, in case (c), if E fails, $\bar{R}_s = \frac{10}{10}$, because all routers are definitely isolated; if A fails, A is isolated and B loses contact with others with probability p, so $\bar{R}_s = \frac{4+3p}{10}$; etc. By combining $\bar{R}_s$ of all network failure states, we obtain the resilience loss of case (c): $\bar{R} = \frac{r}{10}(26 + 23p - p^2)$. The calculation results of other IBGP networks are all shown in Fig. 8.

Case (a) is a traditional full mesh IBGP network with 10 sessions and no route reflector is deployed. Case (b) and (c) are route reflection networks, which have two and one cluster, respectively. Both of them have only four IBGP sessions, less

than the full mesh IBGP network. Thus, cases (b) and (c) are less reliable than case (a). Especially, case (c) suffers from the single point of failure problem. If E fails, all routers are isolated.

There are two ways to increase IBGP network resilience: using redundant reflectors and adding redundant IBGP sessions between clients.

Case (d) uses two reflectors in one cluster. It is much more resilient than case (c), due to the redundant reflectors and 3 additional sessions. For a small network, where the number of BGP sessions is not a big concern, this design is quite preferable. It is even more reliable than case (a) which has the maximum number of sessions. The reason is that route reflections by the redundant reflectors can avoid some cases of router isolation. For example, if link (C, D) fails, in case (a), $Pr[B \overset{s}{\nrightarrow} D] = p$, because other routers do not reflect routes between B and D. While, in case (d), the redundant reflectors, C and E, both reflect routes between B and D, i.e., there are two independent paths from B to D in graph G^s_{BD}. Thus, the communication between B and D is not affected by the failure of (C, D).

However, using more redundant reflectors does not necessarily guarantee higher reliability. Case (e) uses one more reflector and 2 more sessions than case (d), but it still performs worse. This is because a reflector can not reflect routes between its redundant reflectors and their clients (CLUSTER_LIST loop detection). In case (e), there is only one path in graph G^s_{AD} from A to D. If link (D, E) fails, $Pr[A \overset{s}{\nrightarrow} D] = p$. While, in case (d), two independent paths exist, because both A and D are clients and they can exchange routes via reflector C and E. Therefore, if link (D, E) fails, $Pr[A \overset{s}{\nrightarrow} D] = 0$.

Using redundant sessions between clients of the same cluster can also improve resilience. Based on case (c), we introduce one more session between node B and node C in case (f). This improves the resilience slightly, because B and C are not isolated from each other when router E, link (E, C), (A, E), or (A, B) fails. Case (g) even constructs a full mesh among all clients, and it is most reliable among all these IBGP networks. However, in some scenario, using too many redundant sessions among clients can not improve resilience significantly. For example, case (h) only obtains very slightly higher resilience than case (d), thus these three additional sessions seem not worthwhile.

Summary: This case study shows that, without incurring much additional overhead, we can make IBGP networks more reliable (even better than traditional full mesh IBGP networks) by introducing redundant reflectors and sessions appropriately. Our models can measure IBGP reliability quantitatively and thus provide a theory basis for further IBGP network optimization in terms of network resilience.

V. Conclusion

The reliability of IBGP networks has remarkable impact on the stability of Internet routing. Based on the existing network infrastructure, to model and to improve the resilience of IBGP networks is of significant importance. In this paper, we first investigate the reliability of IBGP sessions, which is closely related to BGP timers and TCP retransmission behaviors. We also develop a simple modification of the TCP retransmission. The simulation results show that it can considerably improve the robustness of IBGP sessions and the repairing window almost doubles. Second, we propose a novel reliability model for IBGP networks, which takes into account the dependent failures of IBGP sessions and quantifies the resilience of IBGP networks in various network failure scenarios. Through an extensive case study, we show that our model is effective in characterizing the resilience of IBGP networks and it also provides a theory basis for further IBGP network optimization research.

Acknowledgment

The authors would like to thank Guanghui He, Jun Wang, Kai Chen and King-Shan Lui for many helpful discussions.

References

[1] Y. Rekhter and T. Li, *A Border Gateway Protocol 4 (BGP-4). IETF RFC 1771.*, March 1995.

[2] T. Bates, R. Chandra, and E. Chen, *BGP Route Reflection - An Alternative to Full Mesh IBGP. IETF RFC 2796*, April 2000.

[3] P. Traina, D. McPherson, and J. Scudder, *Autonomous System Confederations for BGP. IETF RFC 3065*, February 2001.

[4] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "Observation and analysis of BGP behavior under stress," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, 2002.

[5] A. Shaikh, A. Varma, L. Kalampoukas, and R. Dube, "Routing stability in congested networks: Experimentation and analysis," in *Proceedings of ACM SIGCOMM*, 2000.

[6] G. Iannaccone, C. nee Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an IP backbone," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, 2002.

[7] Cisco Systems Inc., "Troubleshooting high cpu utilization on cisco routers," in *http://www.cisco.com/warp/public/63/highcpu.html*.

[8] Cisco Systems Inc., "Troubleshooting memory problems," in *http://www.cisco.com/warp/public/63/mallocfail.shtml*.

[9] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of link failures on VoIP performance," in *Proceedings of ACM NOSSDAV*, 2002.

[10] W. Cui, I. Stoica, and R. H. Katz, "Backup path allocation based on a correlated link failure probability model in overlay networks," in *Proceedings of IEEE ICNP*, 2002.

[11] K. V. Le and V. O. Li, "Modeling and analysis of systems with multimode components and dependent failures," *IEEE Transaction on Reliability*, vol. 38, April 1989.

[12] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE Network*, vol. 14, December 2000.

[13] R. Grimmett and D. R. Stirzaker, *Probability and random processes*. Oxford, 2001.

[14] G. R. Wright and W. R. Stevens, *TCP/IP Illustrated Volume 2 - The Implementation*. Addison Wesley, 1995.

[15] "Scalable Simulation Framework Network Models (SSFNet)," in *http://www.ssfnet.org/homePage.html*.

[16] C. J. Colbourn, "Network resilience," *SIAM Journal on Algebraic and Discrete Methods*, vol. 8, July 1987.

[17] C. J. Colbourn, *The Combinatorics of Network Reliability*. Oxford University Press, 1987.

[18] L. B. Page and J. E. Perry, "Reliability of directed networks using the factoring theroem," *IEEE Transaction on Reliability*, vol. 38, December 1989.