Position Statement for Workshop on Internet Routing Evolution and Design

Lan Wang, Lixia Zhang
Computer Science Department
University of California, Los Angeles

Daniel Massey
Information Sciences Institute
University of Southern California

September 22, 2003

Because its design was based on a simple fault model, the current BGP is vulnerable to many unexpected faults. In this position statement, we consider the impact of this limited fault model on BGP's operation resiliency. We use two examples in neighbor BGP communication to illustrate the general problem: (1) it is possible for BGP routing table to contain undetected errors, and (2) it is possible for BGP to collapse under self-induced heavy congestion. These examples show that design based solely on a few common faults cannot lead to a truly resilient infrastructure. We believe that design against unexpected faults is both possible and feasible, as evidenced by our proposed solution to the neighbor communication problem.

1. Consequences of Undetected Faults

BGP was designed to handle, and *only* handle, topological failures and packet losses. However, this fault model is too simplistic for today's Internet. The sheer scale and heterogeneity of the Internet makes all kinds of faults possible. Memory faults may corrupt a router's routing table, causing it to announce false routes (such unexpected faults occurred as far back as during the ARPANET operation in 1970's). The forwarding table of a router may be corrupted while the routing protocol is functioning correctly (see [NANOG 2003]). Given that not all the faults are predictable, a robust routing system should be able to recover from any transient faults. Formally speaking, it should be self-stabilizing, i.e. "regardless of its initial state, it is guaranteed to arrive at a legitimate state in a finite number of steps" ([Dijkstra 1974]).

Unfortunately, BGP, as it is currently designed, is far from self-stabilizing. It assumes that neighboring routers will have consistent routes as long as all the route changes are carried by a reliable transport protocol. When unexpected faults make this assumption invalid, BGP cannot recover from the resulting incorrect state without human intervention. For example, if one or more routers fail to process a withdrawal message after it is received, those routers and their downstream routers will keep the stale route forever because there is no built-in mechanism in BGP to remove them other than the withdrawal message. Since manual detection and correction may take a long time, invalid routes often stick in the Internet long after they have been withdrawn by the originating router (see [NANOG 1998]). Moreover, the routes stored at a router may be accidentally corrupted or intentionally changed, leading to inconsistent routing state between neighboring routers. However, these invalid routes will not be detected by BGP.

2. Danger of Unstable Peering Sessions

The impact of unexpected faults is not limited to BGP neighbor consistency issues alone. Based on the assumption of stable and long-lasting peering sessions, the current BGP design dictates that two neighboring routers must exchange their *entire* routing tables after each session reset. However, analysis of routing measurements collected by RIPE NCC shows that some BGP monitoring sessions were frequently reset during the Nimda worm attack in September 2001 ([Wang 2002]). Most of the sessions failures may have been caused by the severe network congestion during the worm attack. Although the BGP peering setting at the monitoring point is not typical of the operational Internet and thus the excessive session resets are a monitoring artifact, it is evidence that BGP peering does not work well under adverse conditions. Conditions that can destabilize *single-hop* BGP sessions in the operational Internet include the following:

- **Misconfiguration:** misconfigured filters can introduce a large number of invalid routes, causing congestion and disabling routers with less CPU power and memory.
- **Spoofed routing traffic**: attackers can send a large amount of spoofed routing packets ([Gill 2003]). These packets can exhaust link capacity as well as exhaust the victim router's CPU processing power.
- **Amplification of local changes:** the routing protocol itself may amplify a local connectivity change to a large volume of global routing updates.
- **Hardware failure**: the interface cards of router boxes very often have transient failures ([Diot 2002]);

Since a default free routing table typically consists of over 100,000 routes, the routing table exchange can further exacerbate the existing congestion or crash a low-end router, thus starting a vicious cycle of session reset and re-establishment. Session reset is especially problematic when a BGP router has hundreds of peers.

In hindsight, such a poor response to a session reset fault should not be a surprise. The original design listed only a few well-known failure types and engineered the protocol for those events only. No further thought was given to how the protocol might behave when the set of faults deviates from the expected set.

3. Proposed Solutions

BGP's limited ability to detect inconsistencies and its session instability problem impose a potential threat to the dependable data delivery over the global Internet. It is not simply the case that the BGP design overlooked some additional failure modes. Instead, we claim it is impossible to enumerate all the potential faults a protocol may face during its operations. The routing protocol should therefore assume that inconsistencies caused by undetectaed faults do occur and check the consistency between neighboring routers persistently. However, the large size of BGP's global routing table makes the typical soft-state solution of periodic updates infeasible. As a first step towards adding resiliency into inter-domain routing, we have proposed a Fast Routing Table Recovery (FRTR) mechanism that encodes routing table state in small Bloom filters to effectively detect and recover from otherwise unnoticeable errors ([Wang 2003]). Furthermore, FRTR can help reduce the bandwidth overhead of routing table re-synchronization after a BGP session reset. In addition to be a potentially useful mechanism in improving BGP, FRTR demonstrates the feasibility of designing protocols that can be at once resilient, effective, and efficient.

References

[Dijkstra 1974] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. Communications of the ACM, 17:643-644, 1974.

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

[Gill 2003] V. Gill. Lack of priority queuing on route processors considered harmful. NANOG 26, http://www.nanog.org/mtg-0302/gill.html, Oct. 2002

[NANOG 1998] NANOG Mailing List Archive. EU.net/Supernet leaking routes to Sprint. http://www.merit.edu/mail.archives/nanog/1998-10/msg00841.html, Oct. 1998

[NANOG 2003] NANOG Mailing List Archive. Router crash unplugs 1m Swedish Internet users. http://www.merit.edu/mail.archives/nanog/2003-06/msg00491.html, Jun. 2003

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

[Wang 2003] L. Wang, D. Massey, L. Zhang. FRTR: A scalable mechanism to restore routing table consistency. Submitted to Infocom 2004.