INTERNET INEFFICIENCIES AND INSTABILITY

James C. Lu

INTRODUCTION

In 1876, Alexander Graham Bell invented what he called his “electrical speech machine,” modernly known as the telephone [2]. For several years after its invention, the telephone was known only to be nothing more than a laboratory experiment. Shortly after the turn of the century, however, came the advent of circuit switched networks, allowing for the efficient global deployment of telephones. The ability to communicate long distances fundamentally impacted the world socially and economically, while the very concept of communication was transformed before our very eyes.

In the latter half of the 20th century, the concept of communication was to undertake another revolutionary leap when a graduate student of MIT named Leonard Kleinrock published a paper on an innovative concept known as *packet-switching*. This was a breakthrough proposal for an alternative means to establishing a large communications network, contrasting the networks already established in the worldwide telephony infrastructure which utilized *circuit-switching*. The major difference between these two underlying concepts is that a packet-switched network does not require the reservation of network resources (bandwidth) for communication, while a circuit-switched network does. While a packet-switched network allows for more efficient utilization of resources, a circuit-switched network allows for guarantees in service quality. Both methods have their own pros and cons, but packet-switched networks ultimately prevailed as the chosen underlying framework for the communications medium of the next generation: the Internet.

The Internet, similar to the telephone, did not enter society immediately upon invention, and spent ample time as a mere research project. After Leonard Kleinrock’s paper was published in 1961, several other researchers began investing time and effort in packet-switching. By 1972, a small packet-switched network was established, consisting of approximately 15 nodes. Research and development continued, and by the end of the 1970s, there were 200 nodes, by the end of the 1980s, 200,000. However, the network was still only mainly used for linking universities together and establishing computer networks between them. The release of the World Wide Web in the 1990s changed this forever by bringing the Internet into the homes and businesses of millions of people worldwide. The Internet transformed from a specialized university tool to an instrument of our everyday lives.

Presently, there are over 560 million people worldwide who use the Internet. By 2003, this figure is estimated to rise to 760 million [1]. Although these figures are quite staggering, the Internet has proven to scale quite well. Concepts upon which the Internet was built appear to work as well for 560 million nodes as it did for a couple thousand. Nevertheless, the question of how big the Internet can grow supported by the technical and theoretical foundations established decades ago is emerging. Will the Internet ever fail on a global scale?

The ”death of the Internet” [3] would have devastating effects worldwide. The Internet is no longer merely a convenience, but an integral part of society, and its failure would at the very least crush the global economy. As the world becomes more dependent on the Internet, the greater is the threat of an Internet collapse for two reasons. The first reason is obvious: more individuals would be affected by an Internet collapse if there are more individuals who depend on it. The second reason is less obvious: as more individuals begin to use the Internet, the closer we come to an Internet collapse. This is due to the fact that each additional individual that begins to use the Internet adds to the total number of messages that are traveling through the network at any given point in time. Since the devices in the Internet that deliver these messages (routers) have limited capacities, as overall network traffic increases, the likelihood of exceeding the Internet’s total capacity for handling messages similarly increases. Should this occur, messages that attempt to traverse an already saturated Internet may simply be dropped and never reach their intended destinations. In the extreme, the Internet would be so saturated with messages that no messages ever reach their destinations, resulting in an Internet collapse.

In reality, the likelihood of a global Internet collapse occurring anytime soon is near zero. The reason for this is that hardware technology is currently fast enough to allow the Internet to handle an incredibly large amount of traffic. If this is the case, then why even worry about the possibility of Internet failure? First, it is clear that an Internet failure would have devastating consequences. This by itself makes the possibility of an Internet failure an important consideration. Second, although the theoretical behavior of routing algorithms has been thoroughly studied, the deployed, or actual behavior of routing protocols is surprisingly poorly understood [5]. In particular, it has been shown that a great deal of inefficiencies exist throughout the Internet. If the Internet were to ever collapse, it is unlikely that it would be a direct result of having too many users on the Internet. It is more likely that inefficient algorithms upon which the Internet is built would be the true cause of a potential global failure.

Several questions can now be posed. Are there any inefficiencies in the Internet as it currently stands? If so, where? What, if anything, can be done about these inefficiencies? The answer to these questions can improve the Internet, and is the topic of discussion for the remainder of this paper. In particular, we will discuss the possibility of existing inefficiencies and pathologies in global Internet routing.

The remainder of this paper is organized as follows. In the 2nd section, we provide background information regarding the Internet and global Internet routing. Since routing data is required for our study, a detailed description of the data that was examined is described in the 3rd section. Next in the 4th section we show some related work that has already been conducted in this area and on the same data set. We provide our own analysis in the 5th section. After the analysis, some observations are described in section 6. Section 7 discusses the limitations to our method and section 8 suggests further work to work beyond these limitations. We end with some concluding remarks in section 9. Attached to this paper are 10 CDs and their contents are described in section 10. Finally there are acknowledgements and references in sections 11 and 12.

BACKGROUND

The Internet is comprised of millions and millions of hosts, each of which can potentially communicate with each other. Hosts can be anything from personal computers and WWW servers, to nontraditional computing devices such as cell phones, mobile computers, and Web TV. With a physical connection to the worldwide Internet, applications that are running on these hosts can communicate to the applications running on other hosts. Communicating hosts in the Internet exchange data by sending messages to each other, also called packets. The rules for how these packets are to be formatted is specified in a protocol. The predominant protocol for passing messages between hosts on the Internet is called IP, for Internet Protocol. Following the rules of IP, hosts that wish to send messages to each other will format their messages accordingly, and provided that the messages have followed the rules of IP exactly, they make a best-effort attempt to reach the specified destination.

Specified by the rules of IP, when a host sends a message into the Internet, it must include the address of its destination. All hosts on the Internet have unique addresses called IP addresses, and this is the address that is included in each IP packet that is sent. Once the message is sent inside the Internet, intermediate devices known as routers (or packet-switches) will read the destination IP address in each of these packets and forward the packet (hopefully) to another router closer to the destination. Eventually, after passing through several routers, the IP packet should reach its destination host.

When a router receives an IP packet, in order to determine where to forward it to next, the router consults its routing table. All routers in the Internet possess a unique routing table specifying how to forward any IP packet that contains any particular IP address. Thus, the job of a router is inherently quite simple. It simply reads the destination address of the IP packet, looks up the address in the routing table, finds the corresponding address of the next router that will lead closer to the destination (next hop), and forwards the packet on the appropriate outgoing link. But where does this routing table come from?

In order for a router to construct a suitable routing table, it must have some view of the Internet topology. The term Internet topology refers to the structure of the Internet, which includes the location of hosts and the costs of sending messages between them. Abstractly, the entire Internet topology can be modeled as an undirected graph where the nodes are specific hosts and the edges have associated costs. The cost of traversing an edge could be based on any suitable metric such as the physical distance between nodes or the current amount of traffic on that edge. Assuming that every router knows some aspect of the Internet topology, algorithms to determine the best "next hop" for each possible destination host can be executed at every router. The end result is a routing table uniquely calculated for every router.

This description of Internet routing would work perfectly for any network, provided that there are not too many hosts that are a part of it. The Internet, however, is by no means a small network. With nearly 200 million hosts worldwide [4], it would be impossible for each router in the Internet to maintain a suitable "next hop" for each of those possible destinations. Not only would the size of the routing table itself be unmanageably huge, the amount of time and information required to obtain sufficient information regarding the Internet topology to execute the appropriate algorithms would be intolerable.

In order for these algorithms to scale to the size of the Internet, we cannot simply view the Internet as a bunch of connected hosts. Instead, the Internet is divided in to a large number of distinct regions of administrative control, each of which is known as an *Autonomous System* (AS). Examples of an autonomous system can be a network service provider and its customers, a large corporate network, or a college campus. In general, any large group of individual hosts that are administered by a common authority comprise of an autonomous system.

Routing can now be divided into two major forms: intra-AS routing, and inter-AS routing. The former is routing of packets within a particular autonomous system, and the latter is the routing of packets between autonomous systems. Intra-AS routing is very much like how it was described above, with each router in the autonomous system having some view of the entire topology. Note that the topology is no longer of the entire Internet, but just of the router's autonomous system. Thus, it is no longer necessary for a router to maintain a routing table with entries specifying the appropriate "next hop" for every possible destination. Entries need to exist only for destination hosts that reside within the same autonomous system, allowing for smaller, manageable routing tables. Algorithms that derive these routing tables can now have realistic times to convergence.

If a host wishes to send a message to another host that is in another autonomous system, a router that is handling this particular message will know that it must be sent to one of the *gateway routers* in this autonomous system. This is where Inter-AS routing commences. All autonomous systems have one or more gateway routers that communicate with the gateway routers of other autonomous systems. From a gateway router's perspective, each autonomous system is merely another possible destination. It does not view all of the hosts within an autonomous system as possible destinations, but aggregates them all as one possible target. Thus, the gateway router sends the message to the appropriate target autonomous system's gateway router. It does so by consulting its routing table that contains the appropriate "next hop" autonomous system for every associated possible target autonomous system. It is important to notice that the "next hop" is no longer to a host, but to an autonomous system (more specifically, to an autonomous system's gateway router). Once the target gateway router receives the message, Intra-AS routing commences once again to finally get the message to the destination host. This form of hierarchical routing allows for routing to occur at the scale of the Internet.

Protocols exist that specify how routers build routing tables just as protocols exist to define the rules of packet formatting (IP). For intra-AS routing, example protocols include OSPF, ISIS, and IGRP. Administrators of individual autonomous system have the authority to choose what intra-AS routing protocol to utilize. Each protocol has its specific pros and cons, and the choices are left to the administrators. However, for inter-AS routing, all gateway routers must use a standard protocol to exchange routing information. BGP (Border Gateway Protocol) has come to be that standard.

As of date, very little formal study has been conducted as to how well these protocols work in practice. Arguably, it is not important, because the fact that the Internet has grown to its immense size is living proof that these protocols work and do scale remarkably well. Nevertheless, the lack of formal study in the actual behavior of routing protocols becomes increasingly dangerous as the Internet grows in size. True, these protocols and routing principles have proven to scale, but for how long and how large? Some studies have already shown that the Internet is full of inefficiencies, and the only thing keeping it alive is the fact that the underlying hardware is capable of processing all of the extraneous data at satisfactory speeds. However, the rapid growth of the number of hosts on the Internet makes an Internet collapse more and more a possibility, and has caused some researchers to wonder whether these inefficiencies will eventually destroy the Internet.

Theoretically, BGP operates as follows. After convergence of the initial algorithm to determine the optimal routing table of each of the gateway routers, BGP updates are sent between routers in order to propagate news of a change in Internet topology. Such changes can occur due to network failures or policy changes that affect the availability of a path to a given autonomous system. The gateway router topologically closest to the to the failure detects the failure, and sends BGP messages to all of its neighboring gateway routers (or peers) to inform them of this change. This is an example of a BGP withdrawal message. The recipients of the BGP message in turn propagate the news to its neighbors as well, and the entire Internet is updated. Similarly, when a gateway router learns of a new path to a particular autonomous system, it sends BGP announcement messages to all neighboring gateway routers announcing the new addition, and this message is similarly propagated.

From this logic, it is clear that BGP messages should only be sent between routers when there is a change in Internet topology or routing policy. Studies show that this is not the case [5, 6], and the Internet is flooded with extraneous BGP update messages that do not have any specific purpose. Routing instability, defined to be the rapid fluctuation in the availability of specific destinations, would certainly result from these extraneous BGP updates. High levels of routing instability can lead to packet loss, increased network latency, and time to convergence [5]. In extreme cases, high levels of instability can lead to wide area network failures. In order to better ensure that the Internet continues to scale without a threat of an Internet collapse, it is important to locate areas of routing instability, determine the cause, and find a solution.

DATA

In order to study Internet routing instability, it is necessary to first have a large collection of BGP data so that formal analysis can be conducted on it. Several tools developed by the MRT project [7] enable the collection and processing of this data. The IPMA project [8] has been utilizing these tools for the past several years in order to continuously log all of the BGP routing messages exchanged with their probe machines located at several major U.S. network exchange points: Mae-East, Sprint, AADS, PacBell, and Mae-West. All of this data is archived and available for public usage.

The data is located at ftp.merit.edu/statistics/ipma/data/ and between March 1, 1997 and December 31, 2000 is organized as,

```
<yyyy.mm>/<ixp>/[bgp.yymmdd.hh:ss.gz | rsd_dump.yymmdd.gz]
```

where yyyy represents the four digit year, yy the two digit year, mm the month, dd the day, hh the hour, ss the second, and ixp the network exchange point. Data exists for dates outside this range, however the format varies, and so we omit it from consideration.

The rsd_dump files are ASCII routing table dumps at a given point in time. Since we care more about unstable BGP update messages, it is easier to study the BGP updates directly rather than to attempt extrapolation of inefficiencies by looking at changes in the routing tables. Thus, we only consider the BGP update files and omit the rsd_dump files from consideration.

The BGP files are compressed (gzip) binary logs of all BGP update packets exchanged with the probe machines located at the network exchange points. In order to make sense of the data, it must be converted into ASCII format. The Multi-Threaded Routing Toolkit (MRT) includes a utility called route_btoa which takes as input one of the binary BGP logs and outputs an ASCII conversion of it. Conveniently, the output can be specified to be machine-readable, making it easy to write scripts to parse through the ASCII data.

The format of each line of the machine-readable output is,

```
Protocol | Time | Type | PeerIP | PeerAS | Prefix | <update dependent info>
```

where Protocol is either BGP or BGP4, Time is the number of seconds since epoch that the packet was recorded, Type is either A for announcement or W for withdrawal, PeerIP and PeerAS are the IP address and AS number from where the update was received, and Prefix is the route prefix being announced or withdrawn.

For withdrawals, the <update dependent info> is irrelevant, but for announcements it is structured as,

```
ASPath | Origin | NextHop | Local_Pref | MED | Community
```

where ASPath is the autonomous system path of the update, Origin is IGP, EGP, or Unknown, and Local_Pref, MED, and Community are some other BGP specific attributes that are interpreted uniquely per autonomous system. These attributes are usually used to influence BGP path selection.

The following is an example of a BGP update message:

```
BGP | 884831402 | W | 204.70.7.53 | 3561 | 198.163.111.0/24
```

In this example, a withdraw message was received by the probe machine from a peer located at 204.70.7.53 which resides in autonomous system 3561. The message specifies that it is no longer possible to reach the prefix 198.163.111.0/24 through 204.70.7.53.

Another example:

```
BGP | 884831401 | A | 144.228.107.1 | 1239 | 205.113.0.0/16 | 1239 6453 5769 |
        IGP | 192.41.177.241 | 0 | 91
```

Since this is an announcement message, significantly more information is provided as part of the <update dependent info>. In this example, a withdraw message was received by the probe machine from a peer located at 144.228.107.1 which resides in autonomous system 1239. The message specifies that it is possible to reach the prefix 205.113.0.0/16 by going through autonomous systems 1239, 6453, and 5769. The IP address of the next destination along the path (next hop) is 192.41.177.241.

RELATED WORK

In *Experimental Study of Internet Stability and Wide-Area Backbone Failures* [5], several major Internet failures and their probable origins were examined. It was emphasized that the interactions between the underlying components of the Internet are fundamentally poorly understood. Furthermore, specific paths between some Internet Service Providers were examined for stability, which utilized some of the BGP routing data recorded by IPMA. The key result was that instability does exist in the Internet at a high level, and exists globally throughout the Internet backbone. While the death of the Internet is not imminent, instabilities clearly exist and will become more of an issue as the Internet grows in size.

In *Origins of Internet Routing Instability* [6], detailed study has been conducted on the available BGP data at the Mae-East network exchange point considering a 28-month period from March 1996 to June 1998. Pathological routing was defined to be routing information that is redundant and extraneous. Although redundant routing data inherently is not erroneous, it is necessary to identify sources of it so that in an attempt to maintain the Internet’s scalability and robustness, pathological routing can be identified and eliminated. Included in their analysis was a detailed description of gross trends during this 28-month period, specifically, the quantity of BGP announcements and withdrawals were being made on average per day.

According to [6], the total number of BGP updates per day dropped from a staggering 3 to 5 million throughout 1996, to just several hundred thousand per day by the summer of 1998. The decline was attributed to a significant drop in the number of pathological withdrawals, an end result of several vendor hardware and policy changes suggested by some of their previous research. Prior to these changes, 99 percent of all BGP updates were pathological, duplicate BGP withdrawal messages. After full implementation of the recommended software changes, by the summer of 1998, pathological withdrawals dropped from over 2 million to less than ten thousand per day. However, the number of BGP announcements rose steadily from 275,000 to 427,000 per day during the 28-month period. This rise was attributed to specific Internet provider policy changes and ongoing changes in the Internet’s topology.

Despite a dramatic reduction of pathological BGP updates during the 28-month study, it was shown that pathological information was still dominant in Internet routing, and that the pathologies were still derived from router vendor software implementations. In particular, pathologies that were not simply redundant in nature were identified. Examples include identified BGP updates that repeatedly caused a fluctuation in the apparent availability of a given destination, usually caused by alternating withdrawals and announcements of that destination. For several of the remaining pathologies that were identified, further vendor software implementation changes were suggested to remedy the problem.

NEW ANALYSIS

Our analysis begins with a quantitative description of the data that is available, specifically data between the dates of January 1998 to December 2000. The analysis in [6] considered the available BGP data at the Mae-East network exchange point during the 28-month period from March 1996 to June 1998. In a sense, we wish to continue some of their analysis to see if previous trends are still ongoing, or if in fact there are new emerging trends to be further studied.

Below is a table describing the available data:

Time Period	Amount of Data (megabytes) *	Special Characteristics
January 1998	722	
February 1998	416	
March 1998	1,052	

April 1998	1,163	
May 1998	946	
June 1998	770	
July 1998	889	
August 1998	0	No data was available.
September 1998	948	
October 1998	992	
November 1998	1,025	
December 1998	934	
January 1999	948	
February 1999	852	
March 1999	775	
April 1999	0	No data was available.
May 1999	0	No data was available.
June 1999	740	Missing data: 6/8-6/21 and 629-6/30
July 1999	1,648	
August 1999	1,593	
September 1999	0	No data was available.
October 1999	176	Missing data: 10/1-10/28
November 1999	More than 5,000	Too much data to process.
December 1999	More than 5,000	Too much data to process.
January 2000	2,220	
February 2000	1,568	
March 2000	0	No data was available.
April 2000	0	No data was available.
May 2000	1,525	
June 2000	630	Missing data: 6/8-6/30
July 2000	1,819	
August 2000	1,373	
September 2000	346	
October 2000	253	Missing data: 10/16-10/20
November 2000	319	
December 2000	639	

* - Values are for uncompressed data after conversion to ASCII from binary.

As can be seen from the table, some months had missing data or no data whatsoever. Also, the data in November 1999 and December 1999 was incredibly large in quantity and so we could not do the binary to ASCII conversions due to lack of resources. With these constraints, it was decided that only months where there was not any missing data would be considered for analysis.

The method used for analysis was to write a Perl script to parse through the converted ASCII data and to extrapolate some meaningful information from it. Much of the work done in [6] was repeated for the months following the time period after which the analysis in [6] ended. The Perl script would take as input any number of converted ASCII BGP data files that were continuous (i.e. there were no gaps in the data) and output the following for each time segment:

- Total number of withdrawals
- Total number of announcements
- Total number of BGP updates (withdrawals, announcements, or other)
- Most withdrawn prefix
- Number of withdrawals of that prefix
- Average number of seconds between each withdrawal of that prefix
- Most announced prefix
- Number of announcements of that prefix
- Average number of seconds between each announcement of that prefix
- Average number of autonomous systems per announced route

Furthermore, at the end of the output, the above is also displayed for all of the input data files in entirety rather than considering each time segment alone.

Here is an example of the output generated by the script:

```
Day 1 -
Most withdrawn prefix: 205.185.132.0/24
 (withdrawn 800 times averaging 107.8 seconds between each occurrence)
Most announced prefix: 205.185.132.0/24
 (announced 1130 times averaging 76.34 seconds between each occurrence)
 (averaging 3.400 autonomous systems per announced route)
56433 withdrawals
790955 announcements
850267 total updates

Day 2 -
Most withdrawn prefix: 206.79.140.0/24
 (withdrawn 851 times averaging 101.4 seconds between each occurrence)
Most announced prefix: 206.79.140.0/24
 (announced 1528 times averaging 56.52 seconds between each occurrence)
 (averaging 3.410 autonomous systems per announced route)
73376 withdrawals
800534 announcements
876666 total updates

.
.
.

For the month -
Most withdrawn prefix: 205.185.132.0/24
```

```
 (withdrawn 26264 times averaging 101.9 seconds between each occurrence)
Most announced prefix: 212.111.0.0/19
 (announced 64754 times averaging 41.35 seconds between each occurrence)
 (averaging 3.604 autonomous systems per announced route)
2529540 total withdrawals
17180179 total announcements
19768418 total updates
```

For our analysis, we considered one day (a 24 hour period) to be the time segment size that the above results would be calculated for. Since each BGP data file is data for 15 minutes, 96 (4 * 24) files would be considered for each day. For each run of the script, we set as input data an entire month. Thus, for a 30-day month, 2880 (4 * 24 * 30) files were considered. These variables can be altered easily if needed. To change the time segment size, the Perl script can be easily modified to accommodate. If it is necessary to consider a total time span of greater than one month, simply use as input the quantity of BGP data files corresponding to the time period desired.

Below is some portions of the output of running our script for several representative months during the 3-year period. Here we provide the aggregate number of announcements and withdrawals during an entire month. Again, only months that did not have missing data were considered.

Time Period	Total Number of Withdrawals	Total Number of Announcements
January 1998	8,110,247	3,209,765
April 1998	3,650936	9,747,232
July 1998	82,247	207,305
October 1998	3,360,038	8,314,801
January 1999	2,820,381	8,087,600
March 1999	2,447,778	6,558,777
July 1999	3,115,996	15,008,895
August 1999	3,575,527	14,233,127
January 2000	2,473,107	21,520,259
May 2000	3,390,615	13,764,566
July 2000	2,529,540	17,180,179
November 2000	1,311,630	2,450,105

We do not provide the day-by- day statistics in this report since that would span numerous pages. However, they will be provided in CDs containing the ASCII data.

As stated above, the output contains other interesting statistics. Some examples:

Time Period	Most Withdrawn Prefix	Average Time Between Each Withdrawal
January 5, 1998	195.17.162.0/24 (2,851 times)	18.04

April 12, 1998	203.14.240.0/22 (772 times)	111.7
July 20, 1998	203.14.240.0/22 (1,634 times)	52.80
October 4, 1998	207.137.193.0/24 (3,194 times)	9.942
January 30, 1999	207.137.193.0/24 (7,819 times)	11.04
March 22, 1999	207.137.193.0/24 (504 times)	25.76
July 28, 1999	192.100.94.0/24 (738 times)	47.67
August 5, 1999	158.107.0.0/16 (383 times)	225.2
January 4, 2000	192.106.128.0/21 (663 times)	129.7
May 1, 2000	148.225.0.0/16 (692 times)	124.5
July 31, 2000	204.79.190.0/24 (1,375 times)	62.73
November 23, 2000	203.23.38.0/24 (792 times)	91.86

Time Period	Average Number of Autonomous Systems Per Advertised Route
January 2, 1998	3.719
April 21, 1998	3.479
July 27, 1998	3.238
October 20, 1998	3.628
January 3, 1999	4.181
March 16, 1999	3.779
July 12, 1999	3.684
August 22, 1999	3.668
January 14, 2000	3.579
May 12, 2000	3.661
July 21, 2000	3.591
November 24, 2000	3.529

Note that data exists beyond the examples given above. Even though there is no example of “most withdrawn prefix” for a month in aggregate, it is available for all the months we considered. The same holds for “most announced prefix.” We provide no examples of it here, but it is available for each day of each month we considered, as well as for each month in aggregate. In fact, output exists for each day of each month we considered, as well as for each month in aggregate for all output criteria described above for the Perl script. Here are just some representative examples, since providing all of the output would take too much space.

OBSERVATIONS

Immediately, it can be observed that our results are in accordance with [6] by considering the aggregate number of announcements and withdrawals for January 1998 and July 1998. Pathological withdrawals were dominant prior to the implementation of vendor software implementations suggested in [6], but afterwards, the number of withdrawals greatly decreased in number. In fact, by 1998, the number of both announcements and withdrawals seemed to decrease in number by an order of magnitude. Thus, at the end of their analysis, situations were looking quite positive.

Unfortunately, this does not seem to be the case in the months to come. By 1999, the values were back up to the numbers prior to 1998, and by 2000, even higher! However, it is necessary to note that we cannot immediately conclude that this is the result of a recurrence of pathological announcements and withdrawals. Several other factors can contribute to this increase in BGP update traffic. For example, overall Internet usage may have risen (and it has) during the same time period, naturally resulting in a need for more routers and more updates. There may also have been policy changes within autonomous systems that could directly result in a greater number of BGP updates. Without the ability to keep all other factors constant, it would be incorrect to attribute the rise in BGP updates to pathological routing alone. Nevertheless, based on the dramatic rise in BGP updates, it is tempting to make that conclusion.

Looking at the “most withdrawn prefix” examples makes it even more tempting. How can it be that it is necessary to withdraw a prefix 1,375 times in a day averaging a withdrawal every 63 seconds (July 31, 2000)? Without the ability to take a snapshot of the entire Internet topology during July 31, 2000, one still cannot scientifically say that the topology did not warrant 1,375 withdrawals of the same prefix. Perhaps it really was necessary and optimal to do so however unlikely that may seem.

Regarding the average number of autonomous systems on a given advertised route, this figure appears to have remained quite stable over the years. From this, one can infer that if there is pathological routing in the Internet, it is not the actual routes that are advertised that are erroneous, but the excessive number of times that they are advertised that is in question.

LIMITATIONS

There were some limitations to our approach for analyzing the data. First, we only considered a subset of the total data available. A more comprehensive analysis could have included all of the available data while interpolating the numbers for the incomplete months.

We also considered a time span to be a 24-hour period and used a span of 1 month in each input to the Perl script. To be complete, one should consider various time spans. For example, the values should be calculated for 12-hour periods or 48-hour periods. Rather than considering the aggregate data for 1 month, perhaps it should be considered for 2-weeks or 2 months. One can be more confident with the results if they are similar for various segmentations of the 3-year period.

Although the data is available for 5 separate network exchange points throughout America, we only considered Mae-East. It would obviously be more comprehensive to do the same analysis for all of the exchange points.

Finally, as stated in the previous section, a simple observation of the numbers cannot yield any definitive conclusions. One must take all other factors into account and compensate accordingly.

SUGGESTIONS FOR FUTURE WORK

Listed in the previous section are several limitations to the approach used in this paper's analysis. A starting point for future work would be to repeat the analysis in a more comprehensive sense without the limiting factors. Immediately, one can repeat the analysis with all of the data from all network exchange points. Furthermore, the analysis can be repeated considering several different time-segment lengths.

Ultimately, as in [6], the actual topology of the Internet should be studied and understood to see how much of the routing data observed was necessary and how much was extraneous. For example, a single route in particular can be isolated for observation. Based on the policies of the autonomous systems at each endpoint of the route as well as the state of the Internet topology, analysis can be conducted on whether the BGP updates are actually what they should be.

CONCLUSION

Although the main topic of this paper revolved around the Internet and its instabilities, our goal was not to downplay the Internet as a poorly designed infrastructure. Quite the contrary, the Internet has been remarkable in its scalability and adaptability to recent surges in Internet demand. Beginning as a simple network between universities operating with less than 200 nodes, the fact that these protocols and underlying technologies are able to satisfactorily service millions of individuals today reliably is simply amazing.

One of the major challenges of implementing a packet-switched network is that there are no guarantees in how the network will behave. Unlike circuit-switched networks, packets make a best-effort attempt to reach their destinations. No guarantee is given for reliability or throughput. For these reasons, the behavior of the Internet is extremely hard to predict as the number of variables that determine the actual behavior of the deployed Internet protocols is too great.

Our goal is to show the reader that there may be an obvious discrepancy between the theoretically predicted behaviors of Internet protocols and the actual deployed behaviors. BGP updates should only be sent to reflect Internet topology changes or changes in Internet policy. This is clearly may not be the case, as pathological updates appear to flood the Internet. While we have not rigorously proven this fact, we hope that we have shown the reader enough to convince him/her that Internet instability and inefficiencies very possibly do exist, and are definitely worthy of further study.

ATTACHMENTS

Included are 10 CDs that provide the following:

- All binary BGP data from Mae-East from January 1998 to December 2000
- All ASCII BGP data (after conversion using route_btoa) from Mae-East from January 1998 to December 2000 excluding November 1999 and December 1999
- This paper in MS Word 2000 format
- The Perl script used to parse through the ASCII BGP data
- Output of running the Perl script 12 representative months from January 1998 to December 2000

ACKNOWLEDGEMENTS

I thank Jeanna Neefe Matthews for her continued support throughout this endeavor.

REFERENCES

[1] Global Reach, www.glreach.com.
[2] TelecomWriting.com home page, www.privateline.com.
[3] B. Metcalf, “Predicting the Internet’s Catastrophic Collapse and Ghost Sites Galore in 1996,” *InfoWorld*, December 4, 1995.
[4] NetSizer, www.netsizer.com.
[5] C. Labovitz, A. Ahuja, F. Jahanian, “Experimental Study of Internet Stability and Wide-Area Backbone Failure,” University of Michigan Technical Report, December, 1998.
[6] C. Labovitz, G.R. Malan, F. Jahaniam, “Origins of Pathological Internet Routing Instability,” *Proceedings of the ACM SIGCOMM ’98*, Vancouver, Canada, September 1998.
[7] Multi-threaded Routing Toolkit, www.mrtd.net.
[8] Internet Performance Measurement and Analysis Project, www.merit.edu/ipma.