A Finite State Model Update Propagation for Hard-State Path-Vector Protocols

Timothy G. Griffin*

April 2, 2019

Abstract

The Border Gateway Protocol (BGP) implements dynamic routing on the global Internet. BGP is often referred to as a *path vector* protocol since it performs a distributed route computation where best route selection is based on *path attributes* attached to routes as they propagate from router to router. For reasons of scale, BGP is an *incremental* protocol — it sends updates only when required to by a change in a best route. This is in contrast to protocols that periodically resend their entire routing tables to all peers. The incremental nature of BGP suggests that we may be able to analyze a set of BGP updates and *infer* the events that caused them to be generated. Solution of this inference problem would have many applications, including *post mortem* analysis of network failures, automatic alarm generation, automatic correlation of routing changes with other end-to-end metrics. We present an abstract model of BGP update propagation based on Communicating Finite-State Machines. The model suggests that the BGP inference problem is much more difficult problem than previously believed.

Keywords: Internet Routing, update propagation, Border Gateway Protocol, BGP, passive monitoring, root cause analysis.

1 A BGP Update Propagation Model is Needed

The Border Gateway Protocol (BGP) [15, 7, 16] is currently the only interdomain routing protocol used in the Internet to maintain connectivity between Autonomous Systems (ASes). Today the Internet is made up of over 32,000 ASes using BGP to exchange reachability information for over 300,000 IP address ranges (see for example [8]). There is growing interest in the analysis of the *dynamic* behavior of BGP on a global scale. This is due in a large part to important research on global BGP routing stability [12, 11, 9, 10], the availability of software that can be used for passively monitoring BGP sessions (for example, the *zebra* software [4]), and to publicly accessible archives of a large number of BGP update streams from the RIPE Routing Information Services [2] and from Oregon Route-Views [3]. It is very difficult to estimate the number of BGP speaking routers in the global Internet, but a rough estimate is in the range of 100,000 to 200,000, exchanging routing information over several million BGP sessions. Thus, the BGP routing system surely ranks among the largest and most geographically dispersed distributed systems — and its behavior is critical to the stability of the entire Internet.

*Computer Laboratory, University of Cambridge, email: timothy.griffin@cl.cam.ac.uk

BGP is an *incremental* protocol — it only sends updates in response to a changes in its best routes — which suggests that we may be able to analyze a trace of BGP updates and *infer* the events that caused them to be generated. We will dub this the *BGP update interpretation problem*. Solutions of this problem would have many applications, including *post mortem* analysis of network failures, automatic alarm generation, and automatic correlation of routing changes with end-to-end traffic metrics.

However, the author's experience with attempting to solve this problem, in the context of a large backbone network (AT&T's North American IP network), demonstrated that the incremental nature of the BGP update process, together with complex routing policies, makes the BGP update interpretation problem an extremely difficult and perplexing one.

The approach of the current paper is not to solve the BGP update interpretation problem, but instead to develop a model of BGP update propagation as a first step toward this ultimate goal. Such a model will assume complete knowledge of the structure of the network and all routing policies. As a basic requirement, the model should be able to enumerate all possible observations obtainable at a given point of view that could be generated from a given root cause change. Such a model should help us calibrate the complexity of BGP update propagation and better understand the difficulties inherent in the BGP update interpretation problem — especially in the common situation where only partial information is known about network topology and routing policies.

This paper presents such a model based on representing a network of BGP speaking routers as a set of Communicating Finite-State Machines (CFSMs) [5]. The model is a very abstract one — it attempts to ignore all of the inessential details of BGP routing. A network of BGP speaking routers is treated as a large transducer, nondeterministically transforming input signals (representing root cause changes) to output signals at observation points. The model allows us to rigorously define the set of all possible observed signals given a set of input signals. It represents a *phenomenological* rather than a *predictive* model. The model clearly demonstrates that even for networks of BGP speaking routers where *everything is known* (input, topology, and routing policies), the input/output behavior of the network can nondeterministically distort, amplify, or attenuate input signals in combinatorially complex and nonintuitive ways.

This contrasts sharply with a commonly held intuition about BGP update propagation — that updates are essentially *flooded* through a network of BGP speaking routers. In fact, the BGP mechanism of *route flap damping* (RFC-2439 [17]) seems to assume this simple model implicitly. Damping routes is a means of punishing "badly behaved routes" — with the implicit assumption that bad behavior at an observation point means bad behavior at the route's origination point. And this implies a simple flooding propagation model. However, it has recently been shown that route flap damping can actually punish "well behaved" routes [14, 13], causing undesirable loss of connectivity. These results are entirely consistent with the model presented here.

2 A Peek at BGP Updates

BGP is a *path vector* protocol, which means that if a BGP speaking router receives a route announcement from a neighbor, then that neighbor must actually be using that route. Each router is free to use, or not, routes sent from its neighbors. The choice of best routes in BGP is based on locally configured routing policies. BGP is also an *incremental* protocol, which means that it only announces changes (deltas) to its set of current best routes. Routing updates either announce a new route for a given prefix (CIDR block), or they withdraw a route for a previously announced

prefix.

Figure 1 presents a sequence of 22 updates (announcements and withdrawals) received from AS 7018 for a specific prefix, 193.230.191.0/24, over a short period of time (about 1/2 hour). The trace presented ignores all BGP attributes but the prefix and the AS path. The line

```
A 19:03:54 193.230.191.0/24 <7018 701 702 8708 13181>
```

means that at time 19:03:54, an announcement of prefix 193.230.191.0/24 was received with AS path 7018 701 702 8708 13181. This AS path indicates that the prefix was originated by AS 13181, which sent an announcement to AS 8708, which in turn sent the announcement to AS 702, which in turn sent the announcement to AS 701, which finally sent the announcement to AS 7018. The line

```
W 19:04:19 193.230.191.0/24
```

means that at time 19:04:19 a withdrawal from AS 7018 for prefix 193.230.191.0/24 was received. Two announcements in a row, such as

```
A 19:04:48 193.230.191.0/24 <7018 1299 5483 8708 13181>
A 19:05:51 193.230.191.0/24 <7018 3320 5483 8708 13181>
```

represent a router "changing its mind" about its best route. It firsts announces a route with AS path 7018 1299 5483 8708 13181, and then a few seconds later announces a new route with a different AS path, 7018 3320 5483 8708 13181 (note these paths differ in the second ASN from the left). This second announcement represents an *implicit withdrawal* of the previous announcement.

3 Basic Modelling Ideas

We can simplify our modelling task by focusing on the observed behavior of a single prefix. Before presenting an example, let us make two simplifications that are used throughout this paper. We will ignore the attribute values of updates entirely and we will ignore the discreet values of arrival times, preserving only relative time. Suppose we replace each announcement and withdrawal of prefix 193.230.191.0/24 with a single symbol from this table:

route with AS path	A	W
7018 701 702 8708 13181	a_1	b_1
7018 1299 5483 8708 13181	a_2	b_2
7018 3320 5483 8708 13181	a_3	b_3
7018 1239 3320 5483 8708 13181	a_4	b_4
7018 701 1299 1299 1299 5483 8708 13181	a_5	b_5

Then the sequence of updates of in Figure 1 can be represented by the sequence of symbols

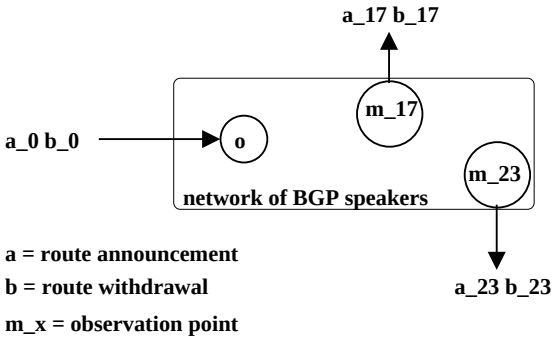
$$a_1b_1a_2a_3a_1a_2a_3a_1a_2a_3a_1a_4b_4a_1a_2a_3b_3a_5a_2a_3a_1b_1.$$

We will call this sequence the *observed output signal*. Note that this signal ignores the discrete arrival times of the updates, and preserves only the relative order of the arrival times of the updates.

What *input signal* could have caused this observed output signal? The naive view of BGP route propagation is illustrated here:

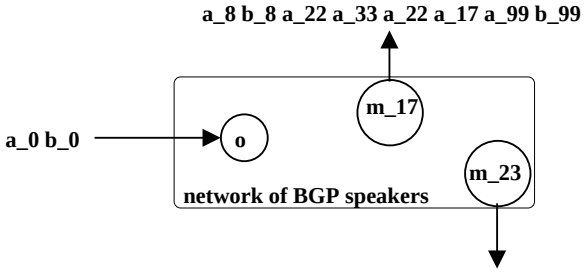
```
A 19:03:54 193.230.191.0/24 <7018 701 702 8708 13181>
W 19:04:19 193.230.191.0/24
A 19:04:48 193.230.191.0/24 <7018 1299 5483 8708 13181>
A 19:05:51 193.230.191.0/24 <7018 3320 5483 8708 13181>
A 19:08:59 193.230.191.0/24 <7018 701 702 8708 13181>
A 19:09:28 193.230.191.0/24 <7018 1299 5483 8708 13181>
A 19:09:59 193.230.191.0/24 <7018 3320 5483 8708 13181>
A 19:12:51 193.230.191.0/24 <7018 701 702 8708 13181>
A 19:13:12 193.230.191.0/24 <7018 1299 5483 8708 13181>
A 19:14:00 193.230.191.0/24 <7018 3320 5483 8708 13181>
A 19:18:06 193.230.191.0/24 <7018 701 702 8708 13181>
A 19:18:24 193.230.191.0/24 <7018 1239 3320 5483 8708 13181>
W 19:18:48 193.230.191.0/24
A 19:19:16 193.230.191.0/24 <7018 701 702 8708 13181>
A 19:19:44 193.230.191.0/24 <7018 1299 5483 8708 13181>
A 19:20:08 193.230.191.0/24 <7018 3320 5483 8708 13181>
W 19:22:24 193.230.191.0/24
A 19:23:23 193.230.191.0/24 <7018 701 1299 1299 1299 5483 8708 13181>
A 19:23:48 193.230.191.0/24 <7018 1299 5483 8708 13181>
A 19:24:08 193.230.191.0/24 <7018 3320 5483 8708 13181>
A 19:26:04 193.230.191.0/24 <7018 701 702 8708 13181>
W 19:26:30 193.230.191.0/24
```

Figure 1: A 23 minute update sequence for the prefix `193.230.191.0/24` extracted from RIPE data.



This naive view pictures the network of BGP speaking routers as a fairly passive set of "route propagation medium" that allows an originating node to "flood" a route throughout the Internet. Each router in this view simply changes some attributes, in accordance with the protocol rules (such as growth of AS path) or and local policies (such as setting of local preference). The figure illustrates this by showing the the input signal a_0b_0 (an announcement followed by a withdrawal) observed as $a_{17}b_{17}$ at observation point 17 and as $a_{23}b_{23}$ at observation point 23.

Experience has shown that reality is much more complex than this naive model would suggest. This is illustrated by a picture closer to "reality" is illustrated here:



The input signal a_0b_0 results in the observed signal $a_8b_8a_{22}a_{33}a_{22}a_{17}a_{99}b_{99}$ at observation point 17, and results in no observation at observation point 23.

Let's try to understand this by considering a very simple example, illustrated in Figure 2. Suppose that router R is sending BGP updates to monitor (observation point) M, and that there are two possible BGP routes, r_1 and r_2, that R can receive from some origin o. We ignore here any intermediate nodes along paths from the origin o to R, and only consider r_1 and r_2 as the final outputs of a route propagation along two distinct paths. When R passes route r_1 to M, it becomes route r_3, signifying the fact that r_3 is in fact a different route from r_1, since r_3 can be used by M to send traffic to R and then on the path associated with r_1. Similarly, when R passes route r_2 to M, it becomes route r_4.

For each route r_1 or r_2, R can receive either an announcement or a withdrawal, and it in turn sends announcements and withdrawals on to M. If we ignore all of the details (attribute values) of the route r_1, then we can denote its announcement or withdrawal by a single symbol, as was discussed above. For example, let the symbol a_1 represent the announcement to R of the route r_1 and let symbol b_1 represent a withdrawal of r_1 received by R. Similarly, we can use a_2, a_3, and a_4 to denote the announcement and b_2, b_3, and b_4 to denote the withdrawals of routes r_2, r_3, and r_4, respectively.

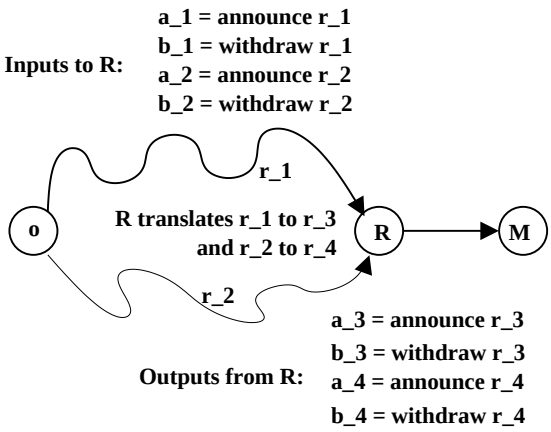


Figure 2: Router R viewed as a signal translator.

For us, a routing signal will be a word over some finite set of symbols (an alphabet). Assume that router R is receiving signals in $\Sigma_1^* = \{a_1,\ b_1\}^*$, from one neighbor and signals in $\Sigma_2^* = \{a_2,\ b_2\}^*$ from the another. Then the BGP process at R will receive signals in $\Sigma_{in}^* = \{a_1,\ a_2,\ b_1,\ b_2\}^*$, since the signals from the neighbors will be interleaved due to different delays in processing and transmission. Note that this interleaving is an inherently non-deterministic process.

With these abstractions, router R can be seen as a translator of input words in $\Sigma_{in}^* = \{a_1,\ b_1,\ a_2,\ b_2\}^*$ to output words in $\Sigma_{out}^* = \{a_3,\ b_3,\ a_4,\ b_4\}^*$. However, this translation is not completely straightforward. Rather, it is a function of the state of router R (which symbols in $\{a_1,\ a_2\}$ it knows about), and R's local routing policies. For the sake of illustration, let us suppose that router R prefers the route r_1 over route r_2. If we ignore BGP rate limiting (modelled in the next section), then the translation at R can be modelled using a deterministic Finite State Transducer (a type of finite-state machine that can produce an output symbol on each state transition). Figure 3 presents this FST. The states of the FST (circles) correspond to the subset of $\{a_1,\ a_2\}$ learned at any time. A state transition arc are annotated with labels of the form $I : O$, where $I \in \Sigma_{in}$ and $O \in \Sigma_{out} \cup \{\epsilon\}$, where ϵ is the null symbol. A state transition from state q_1 to state

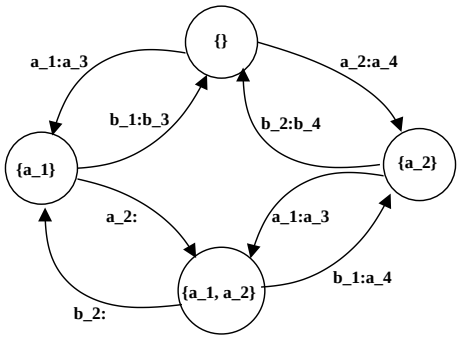


Figure 3: Translation at router R modelled as a Finite State Transducer.

q_2 annotated with $I : O$ means that the transducer in state q receiving input symbol I can change to state q_2 and output O.

The following table presents some input/output pairs for the transducer of Figure 3.

	input signal	output signal
1	a_2b_2	a_4b_4
2	$a_2a_1b_1b_2$	$a_4a_3a_4b_4$
3	$a_1a_2b_2a_2b_1$	a_3a_4
4	$a_1a_2b_2a_2b_2 \cdots a_2b_2a_2b_1$	a_3a_4

This extremely simple example illustrate that it is not impossible to determine the input (modelling the root cause routing change) of such a FST given only the output. Not also that this example can translate a withdrawal to an announcement in some cases. In general, every BGP speaking router in a network will be modelled by an FST, and the global behavior of the network as a signal transducer will be modelled as a set of Communicating Finite-State Machines, which can lead to highly complex interactions as the next section illustrates.

4 A Simple Example

Instead of starting with a set of BGP configurations, we will use instances of the Stable Paths Problem (SPP) [6]. The SPP model was introduced in order to study policy interactions in path vector protocols such as BGP. Here we use it to eliminate much of the complexity of BGP — an SPP instance is obtained from a collection of BGP router configurations by tracing every possible signalling path in the network, then composing all export and import policy functions along every path, and then ranking all routes so obtained at each node using the (rather complex) BGP route selection process. When finished, we simply retain the signalling graph, and the rankings of all (router level) paths, discarding all of the BGP-specific details.

Figure 4 (a) presents a simple SPP with five nodes. The directed arcs denote the flow of routing information (this network is a simple DAG, which most routing networks are not.) Tne node 1 is the *origin*, and each node has a list of *permitted paths* to the origin (the origin is associated with the origination of a single BGP prefix). For example, the path 2 1 next to node 2 indicates that node 2 will accept only this path to node 1. The paths 4 2 1 and 4 3 2 next to node 4 indicate that that node will accept either of these paths, and the order of listing indicates that node 4 prefers path 4 2 1 over path 4 3 2. There are many BGP configurations that could map down to this SPP. For instance, node 4 does not accept the path 4 5 3 1. This could be because it the BGP import policies at node 4 filter out the route associated with this path. Or it could be that the export policies of node 5 never allow the associated route to be sent to node 4. This could be node 5's local decision, or it could be the result of some BGP community value attached at node 3, or even at node 1. All such BGP-specific details are "compiled away" in the translation to an instance of the SPP.

The translation of the SPP SIMPLE to a CFSM will use the enumeration of all permitted paths shown in Figure 4. For example, the announcement of path 5 3 1 will be denoted by the symbol a_6, and the withdrawal of path 4 2 1 will be denoted by b_4.

We will build a distinct CFSM $M(w_1)$ for each possible input word w_1, where w_1 is in the set $\{a_1,\ a_1b_1,\ a_1b_1a_1,\ a_1b_1a_1b_1, \ldots\}$. The finite-state machine associated with node 1 will simple send word w_1 to all neighbors.

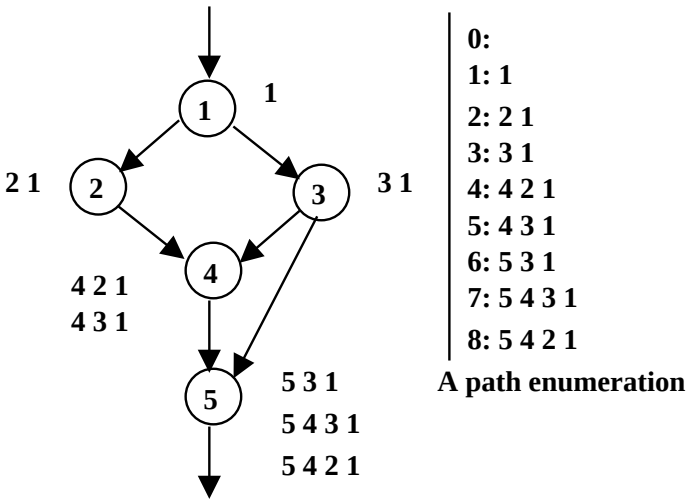


Figure 4: The SPP SIMPLE

The FST for node 4 is presented in Figure 5, and resembles the FST presented in Figure 3. Actually, each state also has a six self loops labled with the announcements and withdrawals that can be send from node 5. Each produces no output. These have been eliminated to reduce clutter. A more complex FST is required for node 5, and is presented in Figure 6.

Let's see how the CFSM for SIMPLE works when $w_1 = a_0b_0$. For CFSM $M(w_1)$ and for $i \in \{2,\ 3,\ 4,\ 5\}$, let L_i be the set of all possible words output at node i. Node 2 simply translates w_1 to $w_2 = a_2b_2$, while node 3 translates w_1 to $w_3 = a_3b_3$. So the possible inputs to node 4 are all possible shuffles (interleavings preserving order) of the words w_2 and w_3. For each, we can compute the output of node 4:

	inputs to node 4.	outputs of node 4
1	$a_3a_2b_2b_3$	a_4b_4
2	$a_2a_3b_2b_3$	$a_5a_4b_4$
3	$a_3a_2b_3b_2$	$a_4a_5b_5$
4	$a_2a_3b_3b_2$	$a_5a_4a_5b_5$
5	$a_2b_2a_3b_3$	$a_5b_5a_4b_4$
6	$a_3b_3a_2b_2$	$a_4b_4a_5b_5$

Finally, we now take all interleavings of the outputs of nodes 3 and 4, we arrive at 71 possible inputs to node 5. Some of these inputs produce identical outputs at from node 5. For example,

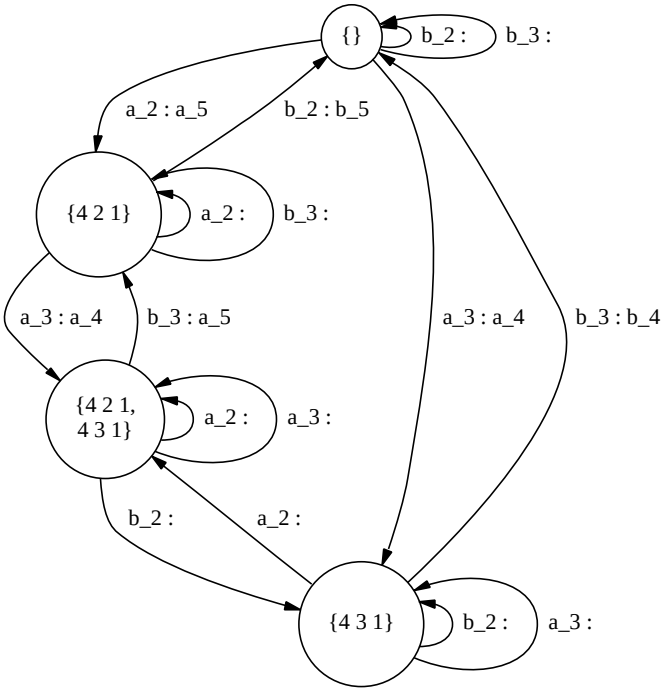


Figure 5: FST at node 4 in SIMPLE.

	node 5 input	node 5 output
1	$a_3a_4b_4b_3$ $a_3a_5a_4b_4b_3$ $a_3a_4a_5b_5b_3$ $a_3a_5a_4a_5b_5b_3$ $a_3a_5b_5a_4b_4b_3$ $a_3a_4b_4a_5b_5b_3$	a_6b_6
2	$a_3a_4a_5b_3b_5$ $a_3a_5a_4a_5b_3b_5$ $a_3a_4b_4a_5b_3b_5$	$a_6a_7b_7$
3	$a_3a_4b_3a_5b_5$ $a_3a_5a_4b_3a_5b_5$	$a_6a_8a_7b_7$

Figure 7 presents the 52 possible outputs from node 5.

The following table shows the sizes of $\mathcal{L}_4$ and $\mathcal{L}_5$ for a few values of w_1.

w_1	$\mid \mathcal{L}_4 \mid$	$\mid \mathcal{L}_5 \mid$
a_0	2	4
a_0b_0	6	52
$a_0b_0a_0$	16	344
$a_0b_0a_0b_0$	55	6813

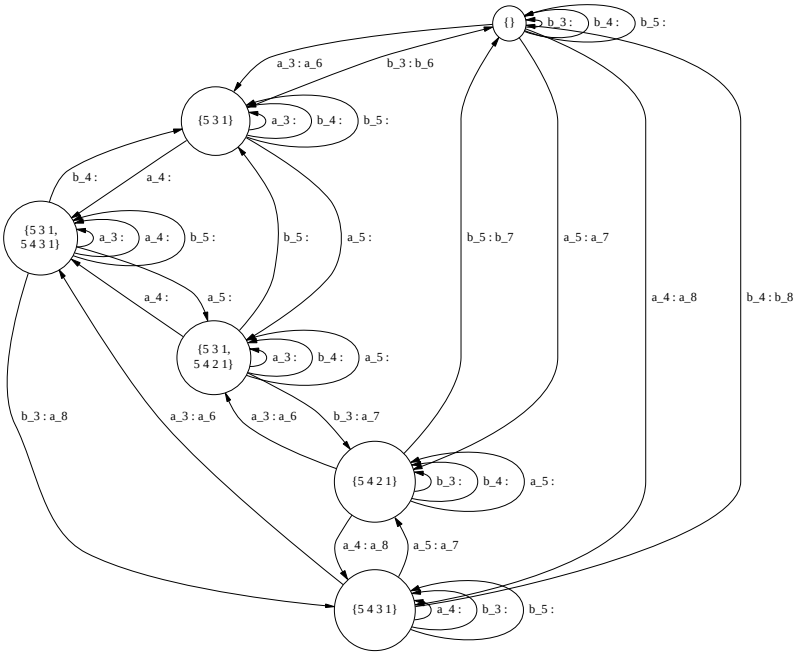


Figure 6: FST at node 5 in SIMPLE.

Note that given most of these 52 words as observed output, it would be very difficult to infer the input w_0.

4.1 Extensions to the Basic Model

In encoding a network of BGP speaking routes as CFSMs there are several adjustments that can be made to the model developed thus far. The most significant is that BGP speakers normally implement rate limiting so that a router can change its state multiple times within a window of time and then only inform its neighbors about the net effect of its changes. The window is normally 30 seconds for EBGP (MinRouteAdvertTimer), and much smaller (perhaps 4 seconds) for IBGP. So far, out model captures BGP when this timer is set to 0.

The impact of BGP rate limiting can be modelled with a *non-deterministic* FST at each node of a CFSM. Since we do not have a notion of time in the model, we will instead fix some integer k and construct FSTs that can non-deterministically consume between 1 and k input symbols before emitting a single symbol that represents its change of state. Figure 8 illustrates this for SIMPLE node 4 with $k = 2$. For example, going from state $\{4\ 2\ 1\}$ to state $\{4\ 2\ 1,\ 4\ 3\ 1\}$ always involves generating the output symbol a_4. However, in this non-deterministic FST, the transition can be made by consuming one of four distinct sequences of input symbols: a_3, a_3a_2, a_2a_3, or b_3a_3.

We now compare two models of the SPP system DISAGREE shown in figure 9 — with and without rate limiting. Figure 10 shows the network state transition diagram for DISAGREE without rate limiting using solid transition arcs. The left and right boxes contain the two distinct stable

a_6b_6	$a_7a_6b_6$	$a_8a_6b_6$	$a_8a_6a_8b_8a_7b_7$
$a_6a_7b_7$	$a_7a_6a_7b_7$	$a_8a_6a_7b_7$	$a_8a_7a_6a_7a_8b_8$
$a_6a_8b_8$	$a_7a_6a_8b_8$	$a_8a_6a_8b_8$	$a_8a_7a_8a_6a_8b_8$
$a_6a_7a_8b_8$	$a_7a_8a_6b_6$	$a_8a_7a_6b_6$	$a_8a_7a_8b_8a_6b_6$
$a_6a_8a_7b_7$	$a_7b_7a_6b_6$	$a_8b_8a_6b_6$	$a_8b_8a_6b_6a_7b_7$
$a_6b_6a_7b_7$	$a_7a_6a_7a_8b_8$	$a_8a_6a_7a_8b_8$	$a_8b_8a_7a_6a_7b_7$
$a_6b_6a_8b_8$	$a_7a_6b_6a_8b_8$	$a_8a_6a_8a_7b_7$	$a_8b_8a_7b_7a_6b_6$
$a_6a_7b_7a_8b_8$	$a_7a_8a_6a_8b_8$	$a_8a_6b_6a_7b_7$	
$a_6a_8a_7a_8b_8$	$a_7a_8b_8a_6b_6$	$a_8a_7a_6a_7b_7$	
$a_6a_8b_8a_7b_7$	$a_7b_7a_6a_8b_8$	$a_8a_7a_6a_8b_8$	
$a_6b_6a_7a_8b_8$	$a_7b_7a_8a_6b_6$	$a_8a_7a_8a_6b_6$	
$a_6b_6a_8a_7b_7$	$a_7a_6a_7b_7a_8b_8$	$a_8a_7b_7a_6b_6$	
$a_6b_6a_7b_7a_8b_8$	$a_7b_7a_6b_6a_8b_8$	$a_8b_8a_6a_7b_7$	
$a_6b_6a_8a_7a_8b_8$	$a_7b_7a_8a_6a_8b_8$	$a_8b_8a_7a_6b_6$	
$a_6b_6a_8b_8a_7b_7$	$a_7b_7a_8b_8a_6b_6$	$a_8a_6a_8a_7a_8b_8$	

Figure 7: Input signal $w_0 = a_0b_0$ at node 1 can produce 52 unique output signals at node 5.

states. The center box represents protocol divergence.

The dashed transmission arcs represent those transitions that are added when we use rate limiting with $k = 2$. An interesting interpretation of this figure results when we think of this as a graph of a Markov process — it states that there is a non-zero probability of divergence (falling into the middle box from which there is no escape) when no rate limiting is used. On the other hand, with probability 1 we can say that with rate limiting the system will eventually converge.

5 Remarks

The model presented in this paper suggests that the solving the BGP update interpretation problem will be much more difficult than previously imagined. A preliminary, and more complex, version of this model was presented at the Network Modelling and Simulation Summer Workshop at Dartmouth in the summer of 2002. This inspired the deployment of a set of "BGP beacons" [1] (prefixes that are announced and withdraw at well known times and rates from various points in the Internet) as a means of calibrating the transformation of signals for route changes with known input signals. These use of these beacons helped demonstrate the fact that route flap damping can punish "well behaved" routes [13].

References

[1] PSG BGP Beacons. http://www.psg.com/ zmao/BGPBeacon.html.

[2] RIPE Routing Information Service. http://www.ripe.net/ripencc/pub-services/np/ris-index.html.

[3] University of Oregon RouteViews project. `http://www.routeviews.org/`.

[4] Zebra routing software. http://www.zebra.org.

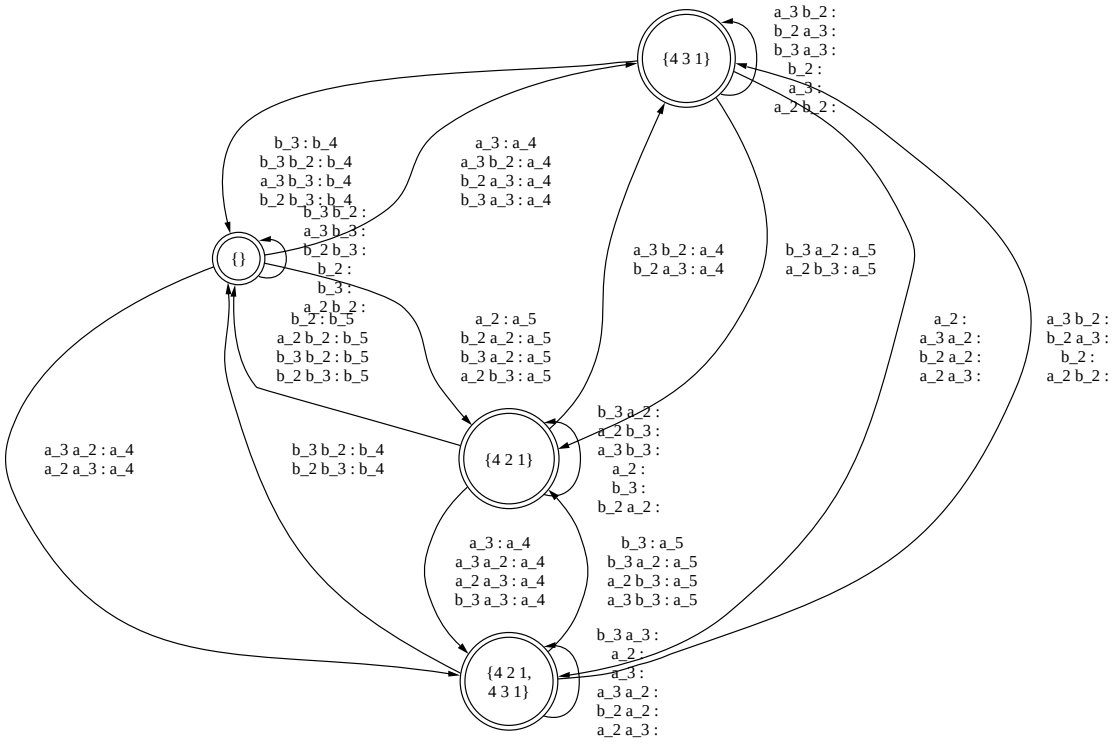


Figure 8: SIMPLE node 4 with $k = 2$

[5] Daniel Brand and Pitro Zafiropulo. On communicating finite-state machines. *J. of the ACM*, 30(2), April 1983.

[6] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(2):232–243, April 2002.

[7] Bassam Halabi. *Internet Routing Architectures*. Cisco Press, 1997.

[8] Geoff Huston. http://www.telstra.net/ops/bgp.

[9] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *Proc. ACM SIGCOMM*, August/September 2000.

[10] C. Labovitz, R. Wattenhofer, S. Venkatachary, and A. Ahuja. The impact of Internet policy and topology on delayed routing convergence. In *Proc. IEEE INFOCOM*, April 2001.

[11] Craig Labovitz, A. Ahuja, and Farnam Jahanian. Experimental study of Internet stability and wide-area network failures. In *Proc. Fault-Tolerant Computing Symposium*, June 1999.

[12] Craig Labovitz, Rob Malan, and Farnam Jahanian. Origins of pathological Internet routing instability. In *Proc. IEEE INFOCOM*, March 1999.

[13] Zhuoqing Morely Mao, Randy Bush, Timothy G. Griffin, and Matthew Roughan. BGP beacons. In *Internet Measurement Conference (IMC)*, 2003.

[14] Zhuoqing Morely Mao, Ramesh Govindan, George Varghese, and Randy H. Katz. Route flap damping exacerbates internet routing convergence. In *Proc. ACM SIGCOMM*, 2002.

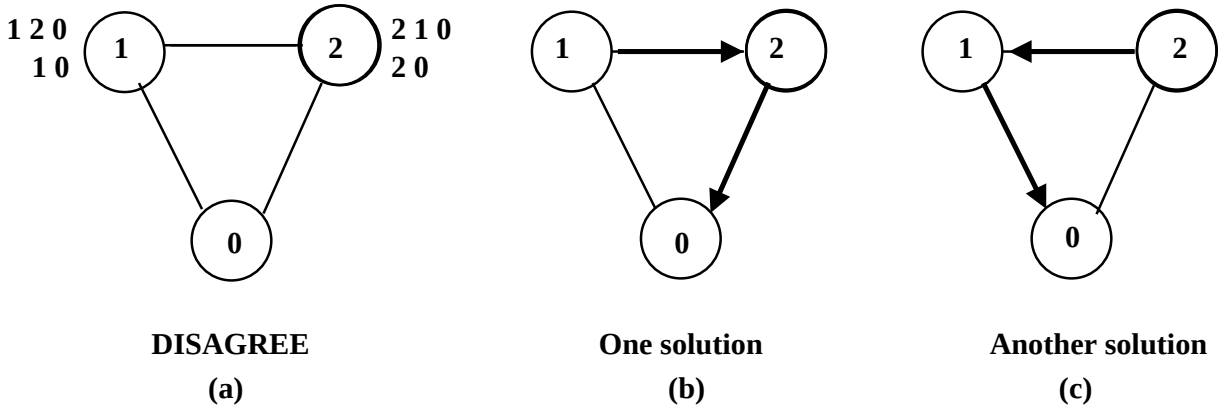


Figure 9: The SPP DISAGREE

[15] Y. Rekhter and T. Li. A Border Gateway Protocol. RFC 1771 (BGP version 4), March 1995.

[16] John W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, 1998.

[17] C. Villamizar, R. Chandra, and R. Govindan. BGP Route Flap Damping. RFC 2439, 1998.

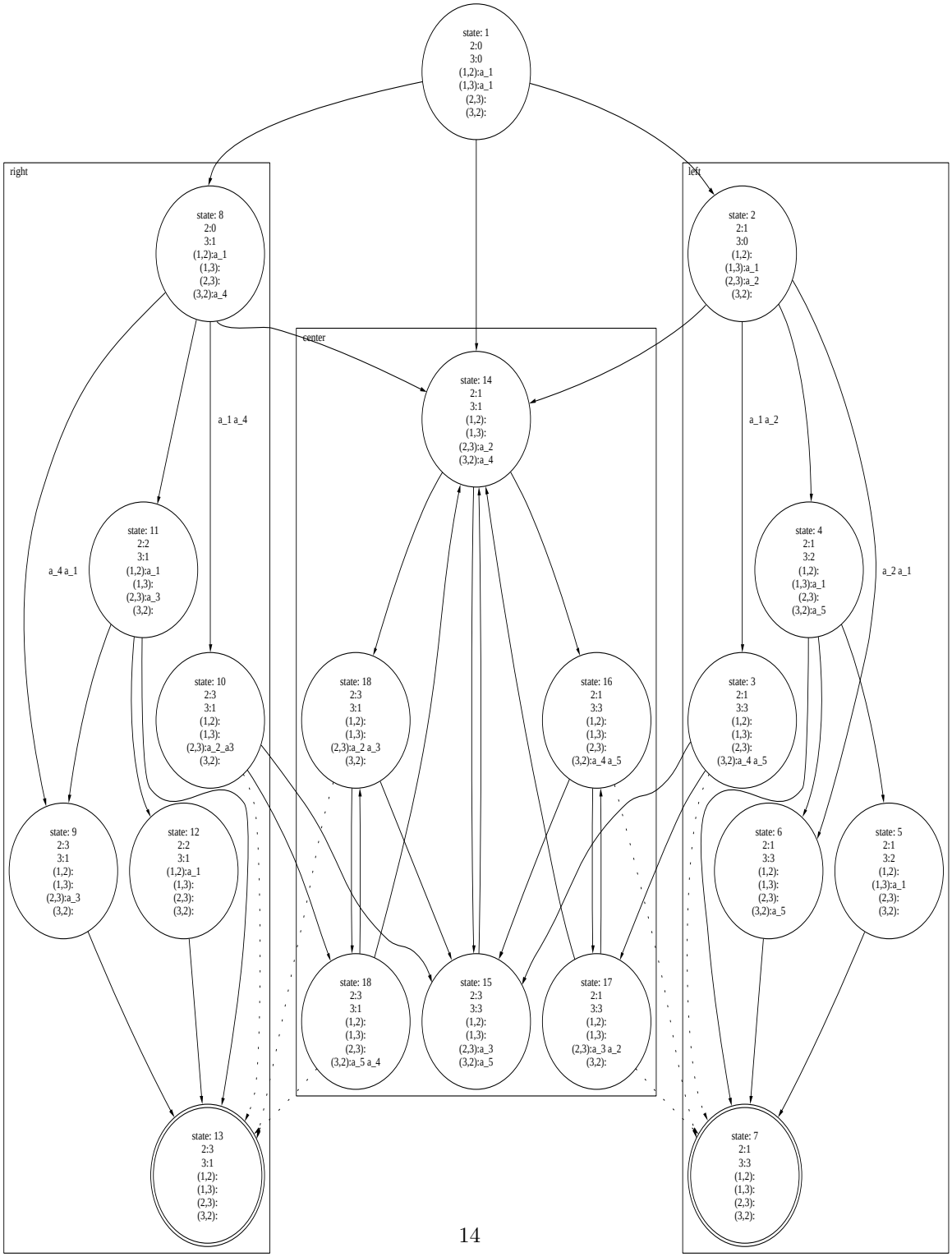


Figure 10: A global state transition diagram for the CFSMs of DISAGREE, with $k = 0$, and $k = 2$.