Expected Convergence Properties of BGP

Ramesh Viswanathan
Krishan K. Sabnani
Bell Laboratories
Lucent Technologies
Holmdel, NJ 07733
{rv,kks}@research.bell-labs.com

Robert J. Holt
Department of Mathematics
and Computer Science
City University of New York
Bayside, NY 11364
rholt@qcc.cuny.edu

Arun N. Netravali
arunnetravali@att.net

Abstract

Border Gateway Protocol (BGP) is the de facto *standard used for interdomain routing. Since packet forwarding may not be possible until stable routes are learned, it is not only critical for BGP to converge but it is important that the convergence be rapid. The distributed and asynchronous nature of BGP in conjunction with local policies makes it difficult to analyze with respect to convergence behavior. We present a novel model which, to our knowledge, is the first one to permit analysis of convergence in the aggregate (*i.e.*, over all message exchange orders between routers regarding route advertisements), rather than worst case behavior. We introduce the notion of probabilistic safety as requiring the probability of convergence to be* 1*. We provide a necessary and sufficient condition characterizing probabilistic safety that shows that probabilistic safety accommodates BGP configurations whose potential divergence stems solely from pathological message sequences. More generally, we show how to compute for any BGP configuration its probability of convergence. For probabilistically safe configurations, we present procedures for computing their expected time to converge as well as the probability distribution on their convergence times. The ability to compute these quantitative characteristics makes our work "constructive" and provides the basis for further understanding and deriving procedures for optimizing network characteristics. Finally, we simulate several network examples and verify the consistency between our analysis and the simulations.*

1 Introduction

The current *de facto* standard for interdomain routing is the BGP4 protocol [11, 12, 5]. To provide global connectivity among autonomous, financially competing domains, an important feature of the protocol is its ability, for each autonomous domain, to use locally independent policies in their route selection criteria and in deciding which routes they choose to export to their neighbors in the network. While BGP's policy-based nature allows enhanced flexibility in routing decisions, it is well known that it also leads to unexpected or undesirable convergence behavior [13, 3]. For example, it can give rise to configurations that may never converge, and path instability may result in delayed convergence [7, 8] which is often slowed further by unintended side effects of other performance tweaks such as route flap dampening [9]. Moreover, the ability of the network to transfer packets is reduced since forwarding paths may be invalid during route convergence.

Because of the distributed and asynchronous nature of the execution of BGP, its precise behavior, even for a fixed network topology and policy configuration, cannot be deterministically predicted since it is dependent on the particular sequence in which different routers send or receive BGP route advertisements. An infinite number of such sequences are consistent with each router faithfully following the protocol. Previous works on divergence or delayed convergence have shown the existence of route advertisement sequences in which the undesired convergence behavior (divergence or long time to converge) could occur. While such work is extremely valuable, it only illustrates a particular sequence leading to divergence without regard to whether such a case is common or pathological. As such, this can be viewed as the *worst case* behavior of BGP. In practice, the behavior of BGP would be governed by route advertisement sequences that are "likelier" to occur and it is *a priori* unclear whether a particular sequence exhibiting bad behavior corresponds to a pathological one unlikely to occur or one that is representative of its commonly observed behavior. Consequently, the main interest of this paper is in developing a better understanding of the convergence properties of BGP in the aggregate (or *expected*) when averaged over all possible route advertisement sequences (that are consistent with the protocol).

We develop a probabilistic model of BGP execution that enables the analysis of its expected convergence behavior. The key insight enabling the formal development of the model is to discretize the execution into time intervals such that: (1) any BGP speaker in the network sends a route advertisement no more than once during any single time interval and, (2) the occurrence of the event of any BGP speaker sending a route advertisement in any time interval is independent of the occurrence of route advertisements at any other speaker or during any other time time interval. These two properties ensure that the evolution of the *probability distribution* of the possible current best paths (representing the network state) over these time intervals is a Markov process, which leads to our formulation of the *dynamical behavior of BGP as a linear system*. In particular, we show that the vector representing the probability distribution of the possible network state at any instant is a linear function of its value at the previous instant. The linear function itself can be computed from the network topology and policy configuration together with probabilities representing the frequencies of route advertisements at different nodes. Significantly, the formulation of BGP's probabilistic dynamics as a linear system allows leveraging the body of work on stability analysis for linear systems (*c.f.* [14]).

We demonstrate the utility of this model for several problems related to the analysis of the expected convergence behavior of BGP. The first problem we consider is that of determining whether a BGP configuration is unsafe, *i.e.*, can exhibit oscillating or divergent behavior. In the non-probabilistic setting, a BGP configuration is considered to be safe [3, 2], if every (fair) route advertisement sequence results in convergence. In this paper, we define a BGP configuration to be *probabilistically safe* if its probability of convergence (over the space of advertisement sequences) is 1. Trivially, any safe BGP configuration is probabilistically safe, but we show that *the class of probabilistically safe BGP configurations includes configurations that would not be deemed safe by previous definitions*. We provide a necessary and sufficient condition for probabilistic safety. The intuitive reading of this characterization is as follows. An oscillating behavior of a BGP execution can be considered to be "transient" or escapable if there is some advertisement sequence that would cause it to reach a converged or stable state, and an oscillating behavior is "permanent" or inescapable if no advertisement sequence from that point onwards can result in convergence. Our characterization shows that a BGP configuration is probabilistically safe if and only if it has no inescapable oscillations. To our knowledge, this is the first formal proof showing that escapable oscillations therefore correspond to pathological route advertisement sequences that are unlikely to persist. More generally, *we present a procedure that, given a BGP configuration and probability distribution on the possible initial network state, computes its probability of convergence.* This procedure can be used to determine whether a BGP configuration satisfies the more relaxed requirement of converging with a probability that is at least some desired threshold, *e.g.*, that it has a 99% probability of convergence.

Our second set of results covers bounds on convergence time. *We provide a general method to compute the expected time to converge for any given BGP configuration.* The method is based on showing that, for any arbitrary BGP configuration, the expected time to converge from any network state can be expressed as a system of linear equations — expected convergence time can therefore be obtained by solving this system. We illustrate this general computation on two families of examples. The first example we consider is that of a full mesh network with shortest path routing. The convergence behavior when the originating destination withdraws its prefix announcement is one of the classical examples where slow convergence behavior has been observed in practice, and where this behavior had been previously argued [7] by demonstrating a long route advertisement sequence required for convergence. We show for this example that the expected time to converge has the same variation with the size of the network as the worst case time to converge, thus corresponding to the observed behavior. On the other hand, *we exhibit a second class of networks for which the expected time to converge is less than the worst case time to converge by a difference of $O(n^2)$, where n is the size of the network.* Together, these two examples show that there is little correlation, in general, between the worst case number of updates and the expected number of updates before convergence. Our second result on expected time to converge is a *method for computing the probability distribution on the convergence times for a BGP configuration.* Besides providing a finer resolution on the expected convergence time, this information could be useful, for example, in determining how long one should wait to ensure that the likelihood of convergence meets some required threshold.

Finally, we present simulations suggesting that the assumptions reflected in our proposed probabilistic model provide a faithful approximation of the behavior of BGP. In our simulations of probabilistically safe BGP configurations, we never observe divergence even if they are unsafe in the non-probabilistic sense. With regard to expected convergence time, the simulation results are consistent with the model-based analysis when queueing effects are not significant.

The rest of the paper is organized as follows. In Section 2, we show how the behavior of BGP can be modeled as a linear stochastic system. Section 3 details the results on convergence probability and Sections 4 and 5 develop the methods for computing expected waiting time and probability distribution on convergence times. Section 6 reports our simulation results and we conclude with directions for further work in Section 7.

2 BGP Execution as a Linear Stochastic System

We present a model of BGP execution with respect to which we can obtain quantitative measures of expected behavior and convergence speed. Our formal development is based on the Stable Paths Problem (SPP), proposed in [3] where it was shown to serve as an abstract model of EBGP (Exterior BGP) configurations where the MED (Multi Exit Discriminator) attribute does not influence path selection. The Stable Paths Problem is, however, expressive enough to encompass import/export policies and policies based on setting the local preference attribute. We review the SPP model of BGP configurations in Section 2.1. Section 2.2 presents an (automata-theoretic) model for the (non-probabilistic) execution behavior of a network whose BGP configuration is specified as an SPP instance. This serves as the basis for a model of the probabilistic behavior of BGP, developed in Section 2.3.

2.1 BGP Configuration Model

In the SPP model of BGP configurations [3], the network is modeled as an undirected graph $G = (V, E)$ with vertices $v \in V$ representing Autonomous Systems (AS) (with an associated BGP speaker). Each edge in (element of) E is of the form $\{u, v\}$ where $u, v \in V$ and represents a peering relationship between BGP speakers at u and v. Route selection is considered with respect to one destination prefix (nlri); as such we take $V = \{0, 1, \ldots, n\}$ with the node 0 having special status as the originating AS to which all other nodes are trying to establish routes. As notational convenience, we define the neighbor set $N(u)$, for any node $u \in V$, as the set $\{v \mid \{u, v\} \in E\}$. Route advertisements consist of AS-level paths in which no AS appears twice;[1] as such, an advertisement is modeled as a path in G that is a sequence of nodes $(v_k v_{k-1} \ldots v_0)$ such that $v_k \neq v_{k-1} \neq \cdots \neq v_0$ and $\{v_i, v_{i-1}\} \in E$ for $1 \leq i \leq k$. The empty path is denoted ϵ and is used to indicate the lack of any route to the destination 0. Nonempty paths $P = (v_1 v_2 \ldots v_k)$ and $Q = (w_1 w_2 \ldots w_m)$ can be concatenated if $v_k = w_1$, in which case their concatenation is denoted PQ and defined to be the path $(v_1 v_2 \ldots v_k w_2 \ldots w_m)$. For any path P, its concatenation with the empty path is defined to be the empty path, *i.e.*, $P\epsilon = \epsilon P = \epsilon$. We use vP as abbreviation for the path concatenation $(v, u)P$, where u is the first node appearing in P. Thus, when a router v receives a route P from one of its neighbors, it can infer vP to be a candidate route from itself to the destination. In the E-BGP setting, when there is a choice between two different routes P_1, P_2 at a router, the selection among the two is determined as: (a) picking whichever of the two paths has a higher local preference attribute set, (b) otherwise, picking the shorter of the two paths, and (c) otherwise, picking the route with the lowest next hop. This is abstracted in the SPP model by taking a ranking function among paths that enjoys certain properties based on this path selection procedure. Finally, export and import policies are abstracted as a set of permissible routes at each router which are the only ones that it could ever select.

Definition 2.1 (BGP Configuration Model) *An instance of a Stable Paths Problem (SPP) is $S = (G, \mathcal{P}, \Lambda)$, where $G = (V, E)$ is an undirected graph, the set of permitted paths $\mathcal{P} = \{\mathcal{P}^v \mid v \in V - \{0\}\}$ with each $\mathcal{P}^v$ a set of paths such that each non-empty path in $\mathcal{P}^v$ is a simple path from v to the node 0, and $\Lambda = \{\lambda^v \mid v \in V - \{0\}\}$ with each $\lambda^v : \mathcal{P}^v \rightarrow \mathbf{N}$ a ranking function that assigns each permitted path a natural number. Additionally, the following conditions need to be satisfied:*

(SP1) *The empty path is permitted: $\epsilon \in \mathcal{P}^v$ for every $v \in V - \{0\}$.*

(SP2) *The empty path is lowest ranked: $\lambda^v(\epsilon) = 0$ for every $v \in V - \{0\}$.*

(SP3) *Strictness: If $\lambda^v(P_1) = \lambda^v(P_2)$ then either $P_1 = P_2$ or $P_1 = (vu)P_1'$ and $P_2 = (vu)P_2'$ for some node u and paths P_1', P_2'.*

Let $S = ((V, E), \mathcal{P}, \Lambda)$ be an instance of SPP. For a node $u \in V$ and a set of paths $W \subseteq \mathcal{P}^v$ with distinct next hops, we define the best path in W with respect to the policy at u, denoted $max(u, W)$, to be the path $P \in W$ such that $\lambda^u(P) \geq \lambda^u(P')$ for all $P' \in W$ if $W \neq \emptyset$ and to be the empty path ϵ if $W = \emptyset$. Note that because of the strictness condition (SP3), there is a unique path $P \in W$ satisfying the conditions of this definition if $W \neq \emptyset$.

2.2 BGP Execution Model

Given an instance of a BGP Configuration S, we now define a model representing the execution of BGP in S, following either a prefix announcement or prefix withdrawal (by the AS 0). The execution model is presented as an automaton with (labeled) transitions, based on the following intuitions. BGP execution proceeds by each node (router) in the network asynchronously advertising routes (to the destination AS represented by node 0) to its neighbors. The sending of an advertisement by a router u is abstracted as an event $advertise_u$. The discrete steps of the execution of the automaton are taken to be time-intervals such that when

[1] The effect of AS-padding could be equivalently achieved by suitable setting of the local preference attributes.

a node sends a route advertisement, it has already processed route advertisements sent (by other nodes) in any previous time interval but not those sent in the same interval. The states of the automaton representing the execution can then be taken to be the latest route advertisements sent by each node, with transitions labeled by the set of nodes that have sent route advertisements in that time-interval.

Let $S = ((V, E), \mathcal{P}, \Lambda)$ be an instance of SPP. Define a path assignment π to be a function that maps each node $u \in V - \{0\}$ to a permitted path $\pi(u) \in \mathcal{P}^u$; intuitively, the path assignment represents the latest routes advertised by each node. We denote the set of all path assignments for S by $Q(S)$. Depending on whether we are considering prefix announcement or withdrawal, any path assignment π is extended to the node 0 by defining $\pi(0) = 0$ for prefix announcement and $\pi(0) = \epsilon$ for prefix withdrawal. Given a path assignment π, the route that would be advertised by a node u, $Next_u(\pi) = max(u, choices(u, \pi))$, where $choices(u, \pi)$ is defined as the set $\{(uv)\pi(v) \in \mathcal{P}^u \mid \{u, v\} \in E\}$, with $\pi(0)$ taken according to whether prefix announcement or withdrawal is being considered.

We now describe the automaton representing the execution following a prefix announcement or a prefix withdrawal.

(States) The state set is $Q(S)$, *i.e.*, states of the global automata are path assignments. For modeling a new prefix announcement, the initial state is taken to be the path assignment π such that $\pi(u) = \epsilon$ for every node $u \in V - \{0\}$. For prefix withdrawal, the initial state can be any path assignment (representing the last route advertisements before the prefix withdrawal message is sent by 0).

(Transition Labels) Transitions are labeled by sets of events of the form $advertise_u$ for each $u \in V - \{0\}$. In other words, each transition label is a set $\{advertise_u \mid u \in U\}$ for some $U \subseteq V - \{0\}$.

(Transitions) In state π, on a label $e = \{advertise_u \, |u \in U\}$ for some set U, the transition relation $\pi \xrightarrow{e} \pi'$ is defined by $\pi'(v) = \pi(v)$ for any $v \notin U$ and $\pi'(u) = Next_u(\pi)(u)$ for any $u \in U$.

A state π is *stable* if for any set $U \subseteq V - \{0\}$, we have that $\pi \xrightarrow{e} \pi$ for $e = \{advertise_u \mid u \in U\}$.

Our execution automaton model differs from the notion of evaluation graph defined in [3] primarily in that our transition labels consist of route advertisement events while evaluation graphs are based on route recomputation events. This alternate interpretation of labels allows us to naturally model MRAI (Minimum Route Advertisement Interval) timers in BGP configurations (further described in Section 6).

2.3 Probabilistic Execution Model for BGP

Given an SPP instance S, we fix some enumeration $\pi_1, \ldots, \pi_{|Q(S)|}$ of all the path assignments (*i.e.*, elements of $Q(S)$). When clear from the context, we will often say that the network state is i to mean that the automaton state is π_i. The automaton capturing the global behavior described in Section 2.2 can be represented by the following transition matrix, where for any set T, we use 2^T to denote its power set consisting of all subsets of T.

Definition 2.2 (Transition Matrix) *For an SPP instance $S = ((V, E), \mathcal{P}, \Lambda)$, the* transition matrix $\mathbf{A}(S)$ *of dimension $|Q(S)| \times |Q(S)|$ with entries $\mathbf{A}(S)_{ij} \subseteq 2^{V-\{0\}}$ is defined as:*

$$\mathbf{A}(S)_{ij} = \{U \mid \pi_i \overset{\{advertise_u \ | \ u \in U\}}{\rightarrow} \pi_j\} \ .$$

Each entry of $\mathbf{A}(S)_{ij}$ of the transition matrix given by Definition 2.2, therefore, indicates the possible sets of nodes that must advertise for the automaton state to change from the path assignment π_i to π_j.

To describe the probabilistic evolution of the path assignments under BGP execution, assume some probabilities on the occurrence of the events $advertise_u$ for nodes u other than the destination 0. This is given by a tuple $\mathbf{p} = (p_1, \ldots, p_n)$, which we call an activation probability vector, with p_i representing the probability of the event $advertise_i$, *i.e.*, node i sending a route advertisement. We assume that the occurrence of each $advertise_i$ is an independent event — this assumption is reasonable because the decision by each node to advertise its best path is a local one without any global coordination among the times that different nodes send their advertisements. Under this assumption, an activation probability vector $\mathbf{p}$ yields probabilities for the occurrence of an event of the form $\{advertise_u \, |u \in U\}$ for some $U \subseteq V - \{0\}$ corresponding to the nodes exactly in the set U advertising their best paths, and probabilities for the occurrence of a set of such events which can be calculated as follows. For a set $U \in 2^{V-\{0\}}$, we define

$$\mathbf{p}(U) = \left(\prod_{i \in U} p_i \right) \prod_{j \notin U} (1 - p_j) \qquad (1)$$

corresponding to the product of the probabilities of the nodes in U advertising and the probabilities of the nodes not in U not advertising, since each of these events is mutually independent. For a set $\mathcal{E} \subseteq 2^{V-\{0\}}$, we define

$$\mathbf{p}(\mathcal{E}) = \sum_{U \in \mathcal{E}} \mathbf{p}(U) \qquad (2)$$

since each of these events is mutually exclusive. It can be easily seen that for the space of all possible events corresponding to the set $2^{V-\{0\}}$, we have that $\mathbf{p}(2^{V-\{0\}}) = 1$ for any probability vector $\mathbf{p}$.

An activation probability vector induces a transition matrix that can be used to represent the probabilistic evolution of the states under BGP execution.

Definition 2.3 (Probabilistic Transition Matrix) *Let $S = ((V, E), \mathcal{P}, \Lambda)$ be an SPP instance. For an activation probability vector $\mathbf{p}$, the induced* probabilistic transition matrix $\mathbf{A}(S)_\mathbf{p}$ *of dimension $|Q(S)| \times |Q(S)|$ is defined by $(\mathbf{A}(S)_\mathbf{p})_{ij} = \mathbf{p}(\mathbf{A}(S)_{ij})$, with $\mathbf{p}(\mathbf{A}(S)_{ij})$ given by Equation (2).*

Intuitively, each entry $(\mathbf{A}(S)_\mathbf{p})_{ij}$ in the probabilistic transition matrix represents the probability that the state would be the path assignment π_j given that the path assignment at the end of the previous time interval was π_i. Let $\mathbf{v}$ be a $|Q(S)| \times 1$ column vector representing the probability distribution on states, with $[\mathbf{v}]_i$ defined as the ith element of $\mathbf{v}$, giving the probability that the last route advertisements correspond to path assignment π_i. The probability distribution of states at the next time interval is then given by $(\mathbf{A}(S)_\mathbf{p})^T\mathbf{v}$ where we use $\mathbf{A}^T$ to denote the transpose of a matrix $\mathbf{A}$. Under the assumption that the probability of a node sending a route advertisement is the same in each time interval, we thus get that the probability distribution of network states $\mathbf{v}_n$ after n time intervals is given by $\mathbf{v}_n = ((\mathbf{A}(S)_\mathbf{p})^T)^n\mathbf{v}_0$, if $\mathbf{v}_0$ is the initial probability distribution on states, and $(\mathbf{A})^n$ denotes the n-fold matrix product of a matrix $\mathbf{A}$. Finally, we recall that a matrix $\mathbf{A}$ of order $N \times N$ is said to be *stochastic* if $0 \leq \mathbf{A}_{ij} \leq 1,\ i, j = 1, \ldots, n$ and $\sum_{j=1}^{N} \mathbf{A}_{ij} = 1,\ i = 1, \ldots, N$. A stochastic matrix guarantees that if $\mathbf{v}$ is a probability distribution, *i.e.*, $0 \leq [\mathbf{v}]_j \leq 1$ for $j = 1, \ldots, n$ and $\sum_{j=1}^{N}[\mathbf{v}]_j = 1$, then the vector $\mathbf{A}^T\mathbf{v}$ is also a probability distribution. It is easy to show that for any activation probability vector $\mathbf{p}$, the induced matrix $\mathbf{A}(S)_\mathbf{p}$ is stochastic. Summarizing, the probabilistic behavior of a BGP configuration can thus be cast as a linear stochastic system via its probabilistic transition matrix.

Proposition 2.4 (Probabilistic BGP Execution) *Let S be an SPP instance, and $\mathbf{p}$ an activation probability vector. We then have the following:*

1. *The matrix $\mathbf{A}(S)_\mathbf{p}$ is stochastic.*
2. *If $\mathbf{v}_n$ is the probability distribution of path assignments after n time intervals, then the probability distribution of path assignments after $n + 1$ time intervals, $\mathbf{v}_{n+1}$, is given by $\mathbf{v}_{n+1} = \mathbf{A}(S)_\mathbf{p}^T\mathbf{v}_n$.*
3. *If $\mathbf{v}_0$ is the initial probability distribution of path assignments then the probability distribution of path assignments after n time intervals is $\mathbf{v}_n = (\mathbf{A}(S)_\mathbf{p}^T)^n\mathbf{v}_0$.*

3 Probability of Convergence

In the deterministic setting, safety of a BGP configuration (*c.f.* [3]) has been taken to mean that there exist no activation sequence which would result in divergence or instability, or conversely that every activation sequence leads to a stable path assignment. In this section, we consider a probabilistic version of this notion, where we require that the probability (over all activation sequences) of the network eventually arriving at a stable path assignment is 1, or conversely that for any path assignment that is not stable the probability of the network eventually arriving at such a path assignment is 0. In Section 3.1, we formalize this intuition. In Section 3.2, we develop a characterization of probabilistic safety. Using this characterization we can then show that probabilistic safety is a more liberal condition in that there can be BGP configurations which would be considered deterministically unsafe but are nevertheless probabilistically safe. In Section 3.3, we present the more general solution for computing the probability of convergence for any BGP configuration.

3.1 Probabilistic Safety Definition

For the rest of this section, we fix some SPP instance S, and use $\mathbf{A}$ to denote its transition matrix $A(S)$. By Proposition 2.4, we have that after n time intervals, the probability distribution on path assignments is given by the column vector $v_n = (\mathbf{A}_p^T)^n\mathbf{v}_0$ where $\mathbf{v}_0$ is the initial probability distribution. The long-term or eventual probability distribution is the limit vector obtained as $n \to \infty$. The probability that the network eventually arrives at some path assignment π_i is therefore given by the i'th entry of this column vector, *i.e.*, $\lim_{n\to\infty}\left[(\mathbf{A}_\mathbf{p}^T)^n\mathbf{v}_0\right]_i$. We consider a configuration to be probabilistically safe if for any path assignment that is not stable (as defined in Section 2.2) we have that this probability is equal to 0. We define three variants of this safety condition; one corresponding to whether both the initial network state and activation probabilities are known, the second corresponding to when the initial network state is known but the activation probabilities can be arbitrary, and finally when both the initial network state and activation probabilities can be arbitrary. The last requirement is applicable to the typical scenario where the network has stabilized previously to some unknown path assignment (or a probability distribution on them), and a change to the network topology or policy configuration results in routes being recomputed starting from this potentially arbitrary prior path assignment.

Definition 3.1 (Probabilistic Safety) *Let S be an SPP instance and $\mathbf{A}$ its transition matrix.*

1. *S is* safe *with respect to an admissible activation probability vector $\mathbf{p}$ and initial probability distribution on*

path assignments $\mathbf{v}_0$ if for any state π_i that is not stable we have that

$$\lim_{n\to\infty} \left[(\mathbf{A}_{\mathbf{p}}^T)^n \mathbf{v}_0\right]_i \;=\; 0 \;.$$

2. *S is* activation independent safe *with respect to an initial probability distribution on path assignments $\mathbf{v}_0$ if for any admissible activation probability vector $\mathbf{p}$, S is safe with respect to $\mathbf{p}$ and $\mathbf{v}_0$.*

3. *S is* initial-state independent safe *if for any admissible activation probability vector $\mathbf{p}$ and any initial probability distribution $\mathbf{v}_0$, S is safe with respect to $\mathbf{p}$ and $\mathbf{v}_0$.*

We next develop necessary and sufficient conditions for each of these three safety requirements.

3.2 Safety Characterization

Our characterization of safety is obtained by classifying the path assignments as being stable, transient, or cyclic. These in turn are defined based on whether they have a non-zero probability of reaching a stable state. Let S be an SPP instance, and $\mathbf{A}$ its transition matrix. For a given activation probability vector $\mathbf{p}$, we define the binary relation $R_{\mathbf{p}}$ on path assignments in $Q(S)$ by $\pi_i R_{\mathbf{p}} \pi_j$ iff ${\mathbf{A}_{\mathbf{p}}}_{ij} > 0$. Additionally, we define the binary relation R on path assignments in $Q(S)$ (that is independent of an activation probability vector) as $\pi_i R \pi_j$ iff $\mathbf{A}_{ij} \neq \emptyset$, *i.e.*, the transition matrix entry for going from state π_i to π_j is non-empty. Using these relations, we can define stable, transient and cyclic states with and without respect to a fixed activation probability vector. In the following definition, we use R^+ to denote the transitive closure of a relation R and R^* to denote the reflexive transitive closure.

Definition 3.2 *Let $S = ((V, E), \mathcal{P}, V)$ be an SPP instance, and $\mathbf{A}$ its transition matrix.*

1. *A path assignment π_i is stable if $\mathbf{A}_{ii} = 2^{V-\{0\}}$. A path assignment π_i is stable with respect to a probability activation vector $\mathbf{p}$ if ${\mathbf{A}_{\mathbf{p}}}_{ii} = 1$.*

2. *A path assignment π_i is transient if there exists a stable path assignment π_j with $\pi_i R^+ \pi_j$. A path assignment π_i is transient with respect to a vector $\mathbf{p}$ if there exists a stable path assignment π_j with $\pi_i R_{\mathbf{p}}^+ \pi_j$.*

3. *A path assignment π_i is cyclic if there is no stable path assignment π_j with $\pi_i R^* \pi_j$. A path assignment π_i is cyclic with respect to a vector $\mathbf{p}$ if there is no stable path assignment π_j with $\pi_i R_{\mathbf{p}}^* \pi_j$.*

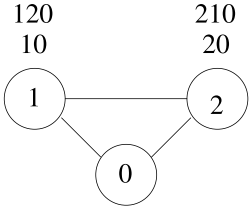

Figure 1. Probabilistically Safe but Deterministically Unsafe Configuration

Intuitively, a transient state is one that has a non-zero probability of reaching a stable state in one or more steps and a cyclic state is one that cannot reach a stable state. Since the transitive closure can be computed inductively, Definition 3.2 yields a straightforward algorithm for determining stable, cyclic, and transient states. Namely, by starting with the set of stable states and inductively including states that can reach them (which can be determined by examing the column entries that are non-zero) we can obtain the set of all transient states. Any states that are not included in this set are then cyclic.

The following theorem states the necessary and sufficient conditions for the three forms of safety.

Theorem 3.3 (Safety Characterization) *For any SPP instance S, we have the following:*

1. *S is safe with respect to a probability vector $\mathbf{p}$ and initial distribution $\mathbf{v}_0$ iff there is no state π_j such that π_j is cyclic with respect to $\mathbf{p}$ and $\pi_i R_{\mathbf{p}}^* \pi_j$ for some π_i with $[\mathbf{v}_0]_i > 0$.*

2. *S is activation independent safe with respect to initial distribution $\mathbf{v}_0$ iff there is no state π_j such that π_j is cyclic and $\pi_i R^* \pi_j$ for some π_i with $[\mathbf{v}_0]_i > 0$.*

3. *S is initial-state independent safe if there is no cyclic state.*

The essence of Theorem 3.3 is that a configuration is safe if every reachable path assignment from the initial state of the network has a non-zero probability of reaching a stable state. Using this characterization, it is easy to show that any probabilistically unsafe configuration is also deterministically unsafe. The converse, however, does not hold. Consider, for example, the "Disagree" network configuration of [3] shown in Figure 1 (in which the permissible paths are listed above each node in order of preference for that node). Although this network can exhibit a divergent sequence (corresponding to the sequence of advertisements $\{1, 2\}, \{1, 2\}, \ldots$), since every path assignment has a non-zero probability of stepping to a stable path assignment, it is in fact probabilistically safe. The particular divergent advertisement sequence has zero measure relative to the set of all advertisement sequences.

Using the algorithm for determining cyclic states and by computing the transitive closure, Theorem 3.3 yields an algorithm for determining the three forms of safety that is polynomial in the size of $Q(S)$; the size of $Q(S)$ can, however, be exponential in the size of S. As the following theorem shows, it is unlikely that one could obtain more efficient algorithms.

Theorem 3.4 *The decision problems of determining safety, activation independent safety, and initial-state independent safety, are NP-hard.*

In [3], the notion of a "trap" in an evaluation graph is defined and it is shown that determining the existence of a trap is NP-hard. The proof of Theorem 3.4 is based on showing reductions from this problem to the characterizing conditions of Theorem 3.3. Interestingly, the existence of a "trap" was identified in [3] on intuitive grounds as a distinction between "weak divergence" and "strong divergence." Theorem 3.3 together with the reductions can therefore be seen as a precise delineation and validation of this distinction.

3.3 Computing Probability of Convergence

As described in Section 3.1, the probability that a BGP configuration will eventually be in a stable state is given by the expression

$$\lim_{n\to\infty} \sum_{\pi_i \text{ stable}} \left[\mathbf{A}^{Tn}\mathbf{v}_0\right]_i$$

where $\mathbf{v}_0$ is an initial probability vector, so that its components are all nonnegative and sum to 1.

The computation of $\mathbf{A}^{Tn}$ can be performed by expressing $\mathbf{A}^T$ in the form $\mathbf{PJP}^{-1}$, where $\mathbf{J}$ is the Jordan form of $\mathbf{A}^T$ and the matrix $\mathbf{P}$ contains the eigenvectors of $\mathbf{A}^T$. The diagonal entries of $\mathbf{J}$ consist of the eigenvalues of $\mathbf{A}^T$ (or $\mathbf{A}$) with appropriate multiplicity. Then $\mathbf{A}^{Tn}$ is computed as $\mathbf{PJ}^n\mathbf{P}^{-1}$, where $\mathbf{J}^n$ is simple to compute due to its special structure. As $n \to \infty$, $\mathbf{J}^n$ converges to a matrix with some 1's on the main diagonal, and 0's elsewhere. The number of 1's is equal to the sum of the number of stable states and the number of distinct cycles. If we call this limiting matrix $\mathbf{J}^\infty$, then the probability of the network ending up in a stable state is the sum of the row elements corresponding to the stable states of the vector $\mathbf{A}^{Tn}\mathbf{v}_0 = \mathbf{PJ}^\infty\mathbf{P}^{-1}\mathbf{v}_0$.

4 Expected Convergence Time

In this section, we consider how the transition matrices corresponding to BGP configurations can be used for computing the expected time to converge. Section 4.1 presents the general procedure for computing the expected number of time intervals to converge based on solving a system of linear equations obtained from the transition matrix. In Sections 4.2 and 4.3, we consider two families illustrating the lack of relationship between expected convergence time and worst case convergence time. Section 4.2 considers a full-mesh network in which one of the routes is withdrawn for which we show that expected and worst case convergence times are similar. In Section 4.3, we present an example family of network configurations for which the expected time to converge is better than the worst case convergence time by $O(n^2)$ where n is the size of the network.

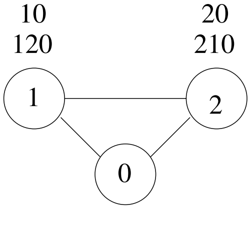

Figure 2. Three Node Full Mesh with Prefix Withdrawal

4.1 Computing Expected Convergence Time

In this section, we show how the expected waiting time (number of time intervals) until a BGP configuration reaches a stable state can be determined exactly by solving a system of linear equations. The key idea behind the general method is to consider for every network state, the expected waiting time to converge after reaching that state. As the base case, for a stable state i, this quantity is trivially 0. From any other state i, if we move to a state j, then the time taken to converge is 1 time interval plus the time taken to converge from state j. The expected waiting time from state i is then the average of this quantity over all next states j weighted with the probability of reaching j. In symbols, letting W_i denote the waiting time until a stable state is reached starting at state i, and recalling that a_{ij} is the probability that one moves to state j in one time interval given that one is starting in state i, this is

$$E(W_i) = \sum_j a_{ij}\left[1 + E(W_j)\right] = 1 + \sum_j a_{ij}E(W_j) \quad (3)$$

for all states i that are not stable. Note that the summation over j may include i, and does so when there is a positive probability of the system remaining in state i after one time interval. If i is a stable state, then naturally $E(W_i) = 0$. It can be shown that the solution to $E(W_j)$ is convergent iff the system is safe with respect to starting in the state

	<10,20>	<120,20>	<ε,20>	<10,210>	<120,210>	<ε,210>	<10,ε>	<120,ε>	<ε,ε>
<10,20>	$\overline{12}$	$1\overline{2}$		$\overline{1}2$	12				
<120,20>		$\overline{2}$						2	
<ε,20>			$\overline{2}$						2
<10,210>				$\overline{1}$		1			
<120,210>					$\overline{12}$	$1\overline{2}$		$\overline{1}2$	12
<ε,210>						$\overline{2}$			2
<10,ε>							$\overline{1}$		1
<120,ε>								$\overline{1}$	1
<ε,ε>									$1+\overline{1}$

Figure 3. Probabilistic transition matrix for Fig. 2

π_j (*i.e.*, a probability distribution on path assignments that maps state j to 1 and all other states to 0).

4.2 Configuration with Long Expected Convergence Time

We consider the example of a full mesh or clique network where the originating destination withdraws its prefix, a classical example exhibiting delayed convergence. By using the general method presented in Section 4.1, we show that the difference between expected and worst-case convergence time is $O(1)$ in the size of the network. We first illustrate the application of the procedure on the case where the network has three nodes and then present the argument for the general case.

The configuration consisting of a full-mesh network topology and shortest path routing for the 3-node case is given in Figure 2. As described in Section 2.2, to model the withdrawal by the destination node 0, we take the path assignment at node 0 to be the empty path ϵ; the initial state is taken to consist of the path assignment where each node has selected the direct path to the destination node 0. The probabilistic transition matrix for the network with two nonzero nodes, Figure 2, is given in Figure 3, where we use the symbols i and $\bar{i}$ to denote p_i and $1-p_i$, respectively, for $i = 1, 2$. Thus $1\overline{2}$ is $p_1(1 - p_2)$. A blank entry indicates probability 0. Solving the system of equations given by the general method of Section 4.1 for this matrix yields

$$\begin{aligned} E(W_1) &= [(p_1 + p_2 - 2p_1p_2)(p_1^2 + 3p_1p_2 + p_2^2) \\ &\quad + p_1^2p_2^2(p_1 + p_2)]/[p_1p_2(p_1 + p_2 - p_1p_2)^2] \end{aligned}$$

with $E(W_1)$ denoting the expected waiting time, from the initial state of $\langle 10, 20\rangle$. In the special case where $p_1 = p_2 = p$, say, this reduces to $E(W_1) = 2(5 - 5p + p^2) /[p(2 - p)^2]$. For small values of p, $E(W_1)$ has the expansion

$$E(W_1) = \frac{5}{2p} - \frac{p}{8} - \frac{p^2}{8} - \cdots \ , \qquad (4)$$

which converges for all $p \in (0, 1]$.

The general n-node case when the probabilities p are small can be analyzed as follows. If the transition probability from one state to another is p, then the expected waiting time for that transition to occur is $1/p$, with no other states involved. Consequently, if the longest path from the initial state to any stable or cyclic state is L, then the expected waiting time is bounded by L/p, or $L/(\max_i p_i)$ if the p_i are different. Second order effects occur when two different nodes are activated with the same time interval, but these occur only on the order $O(1/p^2)$, and all three nodes are activated within one time interval a fraction on the order of $O(1/p^3)$. Consequently they only affect the waiting time to terms of higher order than $O(1/p)$, that is, constant terms, terms of order $p, p^2, \ldots$. For an n-node clique, we can show (by induction on n) that the longest path length is $O(n^2)$ while the shortest path length consisting of single-node activations is $\Omega(n^2)$. From this, it follows that the leading term coefficient of $1/p$ in an expansion similar to Equation (4) for the n-node case would be $\Theta(n^2)$ from which it follows that the difference between the worst and average case is $O(1)$.

We now consider the general case when the probability of activations p is large; more specifically, when the probability of each node not advertising in a given time interval is small in comparison to the reciprocal of the number of nodes. We now show that in this case the expected waiting time from the initial state to the final state is linear in the number of nodes. Under these assumptions, using q to denote $1 - p$, the probability that all n nonzero nodes will advertise in a given time interval is $p^n = (1 - q)^n = 1 - nq + O(q^2)$. The expected waiting time is the reciprocal of this quantity times the length between advertising intervals. If the latter is T, then the expected waiting time for an interval in which all the nodes have advertised is

$$\frac{T}{(1 - q)^n} = T[1 + nq + O(q^2)] \ .$$

This follows from the property of a geometric distribution that the mean is the reciprocal of the parameter (probability of success). When all the nodes advertise, their path advertisements have lengths that are either 1 greater than previously, or the empty path. The initial state is $\langle 10, 20, \cdots, n0\rangle$, and thus after n steps in which all nodes have advertised, all the nodes advertise the empty path. The expected waiting time to converge is bounded by the sum of the expected waiting times to go from one interval where all the nodes advertise to the next such interval. Thus the total expected waiting time is at most

$$nT[1 + nq + O(q^2)] = T[n + n^2q + O(q^2)] \qquad (5)$$

provided that q is small compared to $1/n$.

4.3 Large Variance between Expected and Worst Case Convergence Time

In this section, we present a family of examples where the difference between the expected waiting time and worst case waiting time is $O(n^2)$ where n is the network size. The configuration for a network with n nonzero nodes is defined as follows. For node 1, the only valid path is 10. For node 2, the order of path preference is the order of path preference is 20, 210, 2310, $\ldots$, $234\cdots n10$, and for node 3 it is 30, 310, 3410, $\ldots$, $34\cdots n10$. For node k with $k \geq 4$, the order is $k0$, $k10$, $k(k+1)10$, $k(k+1)(k+2)10$, $\ldots$, $k(k+1)(k+2)\cdots n10$, $k(k+1)(k+2)\cdots n2310$, $k(k+1)(k+2)\cdots n23410$, $k(k+1)(k+2)\cdots n234(k-1)10$. We consider the scenario of prefix withdrawal, so that the initial state is $\langle 10, 20, \cdots, n0\rangle$ and the stable state is $\langle \epsilon, \cdots, \epsilon\rangle$.

The intuition behind the construction is to allow execution sequences that are exactly one of the longest ones obtained in the clique example of Section 4.2, *e.g.*, by having the nodes activate in the order $2, \ldots, n, 1, 2, \ldots, n, 2, 3, \ldots, n, 2$. The path following the sequence above has n^2-2n+3 steps, which leads to a worst case time of $O(n^2)$. All states not obtainable from this sequence are excluded. Thus if the nodes are activated in other orders, particularly if node 1 is activated before all the others are, then the network will converge to the stable state much more quickly.

The instance of this general family of configurations for $n = 4$ is given in Figure 4. If nodes 2, 3, and 4 are activated before node 1 is, then the network may take 11 steps to attain the stable state $\langle \epsilon, \epsilon, \epsilon, \epsilon\rangle$, such as in the sequence 2, 3, 4, 1, 2, 3, 4, 2, 3, 4, 2. If node 1 is activated before the other three are, then the network converges more quickly, and usually much more quickly. If node 1 is activated first, then convergence is reached in three more steps no matter in which order the other three nodes are activated. This ensures that the average case requires significantly fewer steps than the worst case. We now detail the more precise computation of the expected convergence time.

For $n = 3$, in the case where all of the p_i are equal to p, we can solve the expected waiting time from the initial state $\langle 10, 20, 30\rangle$ to the stable state $\langle \epsilon, \epsilon, \epsilon\rangle$ to be

$$E(W_1) = \frac{10 - 10p + 3p^2}{p(2-p)^2} = \frac{5}{2p} + \frac{p}{8} + \frac{p^2}{8} + \cdots \quad .$$

The longest path, of length 6, occurs when the nodes are activated in the order 2, 3, 1, 2, 3, 2 or 3, 2, 1, 2, 3, 2. In general the average waiting time for a specific node to be activated is $1/p$, but when there is a choice of two nodes that may be activated, such as the choice between nodes 2 and 3 at the first step of the longest path, the average waiting time is $1/(2p)$. Thus the waiting time for the longest path is $1/(2p) + 5/p = 11/(2p) = 5.5/p$. Consequently the average waiting time, with leading term $2.5/p$, is approximately 5/11 = 0.455 of the longest waiting time. associated with the paths of length 6.

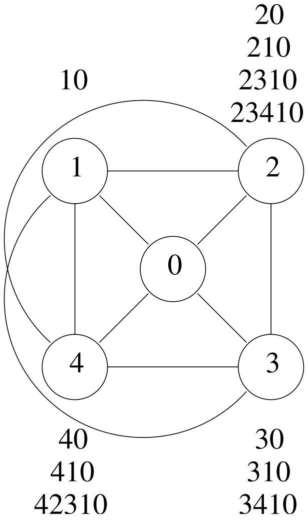

Figure 4. Four-nonzero-node example with large difference between average and worst cases.

For four nonzero nodes ($n = 4$), the expected waiting time from the initial state $\langle 10, 20, 30, 40\rangle$ to the stable state $\langle \epsilon, \epsilon, \epsilon, \epsilon\rangle$ is

$$E(W_1) \quad = \quad \frac{2.628}{p} + 0.235 + 0.190p + 0.143p^2 + \cdots \; .$$

The longest paths, of length 11, are obtained when nodes 2, 3, and 4 are activated in any order, then node 1, and then nodes 2, 3, 4, 2, 3, 4, 2. The waiting time for the worst case is thus $1/(3p)+1/(2p)+9/p = 9.833/p$. Thus the expected waiting time is approximately 0.267 times the longest time. This smaller ratio (compared to 0.455 for $n = 3$) indicates that the ratio of the average to longest times is decreasing markedly with n.

More generally, it can be shown that the difference between expected convergence time and the longest time is quadratic in n, the number of nodes. Even if we assume the worst case longest path length whenever node 1 is not activated first, which occurs with probability $1 - 1/p$ when all the node activation probabilities are equal, that would give an expected waiting time of $(1/p)(n+1) + (1-1/p)(n^2-2n+3)$, and the difference between this and the longest path length is $(n^2 - 3n + 2)/p$.

5 Probability Distribution of Convergence Times

In this section, we provide a general method for obtaining the probability distribution on the length of time that any given BGP configuration would take to converge. More specifically, given the probabilities of activation of different BGP speakers within a fixed time interval, we show how to compute the probability distribution on the number of time intervals that will be taken to converge. This computation takes into consideration time intervals during which no nodes or multiple nodes are activated.

The general procedure is obtained by considering the probability of the network being in a stable state after a certain number of time intervals. Specifically, let $P_n(i)$ be the probability that the network is in state i after the n'th time interval. Then the probability that the network will converge within L time intervals, denoted $P_{\leq L}$, is given by summing up $P_L(i)$ for all stable states i. The probability that the network will take exactly L time intervals to converge, denoted $P_{=L}$, can then be computed using $P_{\leq L}$ by the difference $P_{\leq L} - P_{\leq L-1}$. Using the values for $P_n(i)$ given by Proposition 2.4, we therefore obtain the following theorem, where we use $\mathbf{I}$ to denote the identity matrix whose diagonal entries are 1 and all other entries 0.

Theorem 5.1 *Let S be an SPP instance, $\mathbf{p}$ be an activation probability vector, and $\mathbf{A}$ be its induced probabilistic transition matrix. If $\mathbf{v}_0$ is the initial probability distribution of path assignments, the probability distribution of converging within n time intervals and probability distribution of converging after exactly n time intervals are given by the following equations:*

$$\begin{array}{rcl} P_{\leq L} & = & \sum_{\pi_i \textit{ stable}} \left[\mathbf{A}^{TL}\mathbf{v}_0\right]_i \\ P_{=L} & = & \sum_{\pi_i \textit{ stable}} \left[(\mathbf{A}^{T} - \mathbf{I})\mathbf{A}^{T(L-1)}\mathbf{v}_0\right]_i \end{array} .$$

6 Simulation Results

The simulation studies were conducted using SSFNet [1]. SSFNet is a discrete-event based simulation package in which simulation is performed at the IP packet level. Network models can be specified using a configuration language DML, and SSFNet supports the emulation of multiple network protocols at each router including BGP. The BGP implementation is compliant with the BGP-4 specification in RFC 1771 [11], while allowing configuration of certain network operation parameters. In our experiments, each SPP instance $((V, E), \mathcal{P}, \Lambda)$ considered was mapped to a network model (in DML) by using an AS-level topology corresponding to the graph (V, E) and taking each AS to consist of a single router. This eliminates the effects of internal routing and I-BGP that have not been considered in our model. The AS corresponding to node 0 of the SPP instance was taken to announce or withdraw a single destination prefix which is the only destination prefix processed in the simulations. Router workload is emulated in the experiments by imposing a synthetic CPU processing delay after the processing of each BGP message — for the results described below, this delay was set to be randomly selected between 0.01 and 1 second. Advertisements are rate-limited by an MRAI timer which is jittered as follows. For an MRAI setting of t seconds, after each expiration of the timer, the next value of the timer is set to expire after a value chosen randomly in the interval $[0.75, 1]$ times t. The randomness in the CPU delay together with the randomness due to the MRAI jitter allow for reorderings of BGP message exchanges in the simulations. Our simulation framework is largely inspired by the methodology of [4]; additionally, we have included MRAI jitter. In the experiments described below, withdrawal advertisements were not rate limited and sender side loop detection was not used, though these choices do not seem to qualitatively affect the results.

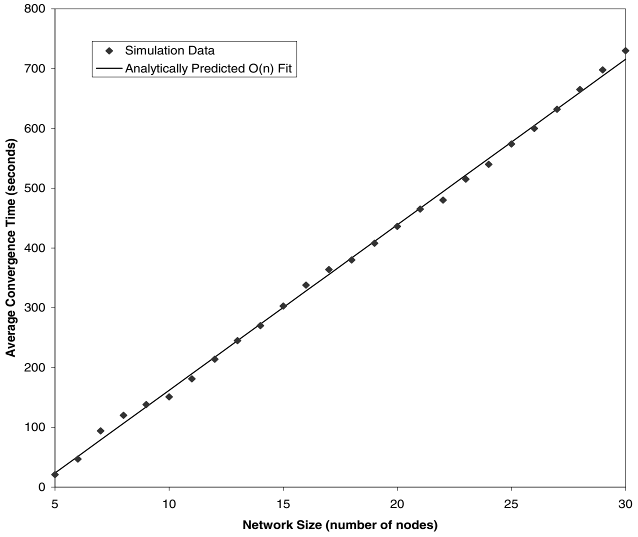

Figure 5. Variation of Average Convergence Time with Network Size

6.1 Probabilistic Safety

The aim of this class of experiments was to examine how well the analytical notion of "probabilistic safety" corresponds to lack of divergence in practice. To this end, we define a family $\mathcal{D}_n$ of BGP configurations (for any natural number n) with the property that each configuration $\mathcal{D}_n$ is probabilistically safe (i.e., has probability of convergence equal to 1) but is deterministically unsafe (i.e, there exists a

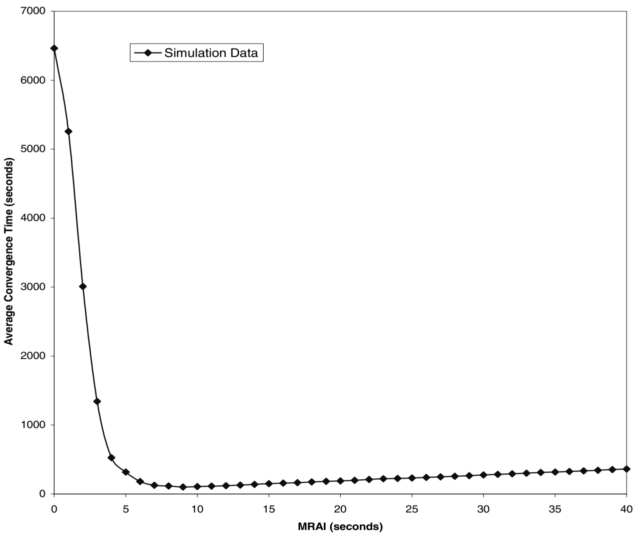

Figure 6. Variation of Average Convergence Time with MRAI for Clique of Size 15

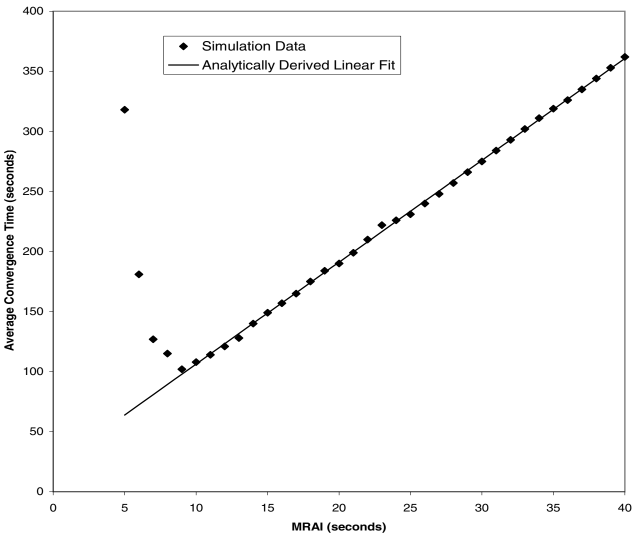

Figure 7. Comparison of Simulation and Model Analysis of Variation of Average Convergence Time with MRAI for Clique of Size 15

message exchange sequence that would lead to divergence). The BGP configuration $\mathcal{D}_n$ consists of $2n{+}1$ network nodes labeled $\{0, 1, \ldots, 2n\}$, with links between nodes i and $i{+}1$ for every i, links between nodes i and $i + 2$ for every i, and links between nodes i and $i + 3$ for every odd i. The policies at each node are such that: (1) at every node that is an odd i, a path including the node $i + 1$ is preferred over a path that does not include it, (2) for any even i, a path including the node $i - 1$ is preferred over a path that does not include it, and (3) at every node i, any path that includes both j and $j + 1$, for some odd j with $j, j + 1 < i$, is not permitted. Each of these configurations was simulated for MRAI values ranging from 0 to 30 seconds, in increments of 5 seconds. Random seeds were used to generate the MRAI jitter and CPU delay values, and for each fixed value of MRAI timer or CPU delay, the experiment was run with 30 random seeds to generate the alternate value. This resulted in a total of 6300 runs (900 runs for each MRAI value and 7 MRAI values) for each network $\mathcal{D}_n$ (for a fixed n). The value of n itself was varied from 5 to 20 (in increments of 1) resulting in a total of $100,800$ runs. Each of these runs resulted in a stable state. Thus they match exactly with the model-based prediction of the probability of convergence being 1 and seem to confirm that divergent message sequences in probabilistically safe configurations are pathological and never occur in practice.

6.2 Expected Convergence Time

The aim of this class of experiments was to compare the analytical results on expected convergence time obtained on the basis of our model with simulation results — the points of similarity or difference would then identify the network operating conditions under which our mathematical model provides a good approximation. For a particular BGP configuration, we have shown that we can analytically derive exact expressions for its expected convergence time. However, unlike the results on probability safety, these expressions are functions of the activation probability parameters which cannot be explicitly set in the simulation. Quantitatively comparing these would therefore require obtaining exact estimates of the activation probability values that correspond to the particular operating conditions. Therefore, the comparison made in these experiments was the qualitative prediction of the variation of convergence time with certain parameters, such as network size and the MRAI timer value. These results are therefore less sensitive to the exact estimates of the activation probabilities.

We consider the example of a clique with prefix withdrawal. With advertisement rates limited by the MRAI timer, the model-based analytical prediction for expected convergence time is given by $T(n + n^2q + O(q^2))$ (Equation (5), proved in Section 4.2), where T is the unit time-interval taken for advertisements, q is the probability that a node would not advertise in the interval, and n is the number of nodes other than the origin node from which the destination prefix is withdrawn. If the MRAI timer value is m and jitter is in the interval $[0.75, 1]$, then the time interval to take for T in which one is assured that a node cannot advertise more than once is $T = 0.75m$. Since q is relatively small, the model-based prediction of variation of convergence time is $O(nm)$ where n is the number of nodes and m is the MRAI value, *i.e.*, linear in the number of nodes and MRAI timer value and we examine this variation through simulation.

Regarding the linear growth with network size, we ran the simulation for clique sizes ranging from 5 to 30 (this is the number of nodes including the destination AS). For each clique size, average convergence time was computed over 900 runs obtained from a combination of 30 random seeds for CPU delay and 30 random seeds for MRAI jitter, with MRAI value set to 30 seconds. Figure 5 shows the average convergence time obtained in this manner. The data values fit well with a linear variation and the best linear fit obtained through a least squares method is shown in Figure 5 as well. A measure of this fit is the R squared correlation coefficient which in general would be between 0 and 1, with a value closer to 1 indicating a closer correlation. This correlation coefficient for the linear fit given in Figure 5 is 0.998604.

With respect to the linear growth with MRAI value, we ran the simulation for a fixed size clique and MRAI values ranging from 0 to 30 seconds. For each MRAI value, average convergence time was computed over 900 runs obtained from a combination of 30 seeds for CPU delay and 30 seeds for MRAI jitter. Figure 6 shows these results for a clique size of 15 nodes which indicates that convergence time initially decreases with increasing MRAI values until an MRAI value of about 9 seconds after which it increases. The fact that the growth beyond 9 seconds is linear is shown more clearly in Figure 7 where the plot primarily focuses on this latter increasing section of Figure 6. The best linear fit that is also plotted in Figure 7 has an R squared correlation coefficient of 0.999426 which indicates that for MRAI values of more than 9 seconds, the simulation results match the analytical linear prediction very well. Similar linear growth beyond a certain MRAI value is obtained for other clique sizes as well, and is also consistent with simulation experiments previously reported in [4, 10] (where MRAI timers were not jittered). The lack of consistency for lower MRAI values, on the other hand, can be traced to the fact that when MRAI values are low, at each router, the queue of route advertisements received from its peers is not fully processed when the route advertisement at the expiration of the MRAI timer is sent. This results in advertisements of routes that are based on earlier advertisements and that therefore do not correspond to the best path based on the latest peer advertisements. Since our execution model does not explicitly include queues, any route advertisements in our model are assumed to be based on the latest peer advertisements. When queues are not fully processed at the instants of route advertisements, this may therefore result in state transitions that would not be present in our model, which therefore results in a mismatch. Queuing therefore seems to have effects that are not captured within our model; however, it should be noted that the default MRAI value in commercial implementations is 30 seconds and the simulations would therefore seem to suggest that in network operating conditions with this default MRAI value, queuing effects are not significant.

7 Conclusion

In this paper, we have presented a mathematical model of BGP execution that permits analysis of its probabilistic or aggregate behavior. Using the model, we are able to identify the notion of probabilistic safety as having probability of convergence be 1 which is a more permissive notion than requiring that all message exchange sequences lead to convergence. Probabilistic safety allows us to distinguish pathological oscillating message exchange sequences from those that can actually occur in practice. The model is simple enough to permit quantitative analysis — as examples, we have given general procedures for computing probability of convergence, expected time to converge, and probability distribution on times to converge. At the same time, results from simulation seem to suggest that the model provides a useful approximation of the complexities of BGP behavior, especially when the RIB-IN buffer queues have been typically completely processed when route advertisements are sent.

As opposed to a model from which we can only obtain decision procedures (*i.e.*, a yes or no answer), the quantitative nature of the analysis supported by the model would make it well-suited to designing procedures for optimizing BGP configurations with respect to requirements on probability of convergence or time to converge. At the same time, we consider the work presented here an initial basis for exploring further questions not addressed in this paper. Firstly, the algorithmic procedures presented in this paper are polynomial in the size of the matrices corresponding to BGP configurations which could be exponential in the size of the networks. By Theorem 3.4 and the connection between expected convergence time and probabilistic safety (stated at the end of Section 4.1), it follows that the existence of more efficient procedures for exact solutions is unlikely. However, one could consider obtaining more efficient algorithms for approximation versions of these problems. Secondly, we have not addressed the question of obtaining exact estimates for the probability parameters in our model. Many of our results such as the characterization of probabilistic safety and qualitative understanding of the variation of convergence time do not depend on the exact values of these probability parameters. However, quantitative estimates of probability of convergence or expected convergence time for a specific network configuration are inherently dependent on these probabilities. As such, general principles for obtaining the probability parameters on the basis of measured network delays and MRAI timer values would be valuable. Finally, extending this work to Interior-BGP and quantifying the impact of queuing effects on convergence are further directions to explore.

8 Acknowledgments

The research of Robert Holt and Arun Netravali was supported in part by Bell Laboratories of Lucent Technologies.

References

[1] SSFNet: Scalable Simulation Framework. http://www.ssfnet.org/.

[2] T. Griffin, F. B. Shepherd, and G. T. Wilfong. Policy disputes in path-vector protocols. In *ICNP*, pages 21–30, 1999.

[3] T. Griffin and G. T. Wilfong. An analysis of BGP convergence properties. In *SIGCOMM*, pages 277–288, 1999.

[4] T. G. Griffin and B. J. Premore. An Experimental Analysis of BGP Convergence Time. In *9th International Conference on Network Protocols (ICNP)*, November 2001.

[5] B. Halabi. *Internet Routing Architectures*. Cisco Press, 1997.

[6] H. J. Karloff. On the convergence time of a path-vector protocol. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004*, pages 11–14, New Orleans, January 2004.

[7] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *Proceedings of SIGCOMM*, pages 175–187, 2000.

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

[9] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz. Route flap damping exacerbates internet routing convergence. In *Proceedings of the 2002 SIGCOMM conference on Applications, technologies, architectures, and protocols for computer communications*, pages 221–233. ACM Press, 2002.

[10] J. Nykvist and L. Carr-Motyckova. Simulating convergence properties of BGP. In *Proceedings of the 11th International Conference on Computer Communications and Networks (ICCCN 2002)*, pages 124–129, Oct. 14-16 2002.

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

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

[13] K. Varadhan, R. Govindan, and D. Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks (Amsterdam, Netherlands: 1999)*, 32(1):1–16, 2000.

[14] L. Zadeh and C. Desoer. *Linear System Theory*. McGraw-Hill, New York, 1963.