Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

PATH PRESERVING SCALE DOWN FOR VALIDATION OF INTERNET INTER-DOMAIN ROUTING PROTOCOLS

Glenn Carl
Shashi Phoha

EE Department and the Applied Research Laboratory
Pennsylvania State University
University Park, PA 16802, U.S.A.

George Kesidis

CSE and EE Departments
Pennsylvania State University
University Park, PA 16802, U.S.A.

Bharat Madan

The Applied Research Laboratory
Pennsylvania State University
University Park, PA 16802, U.S.A.

ABSTRACT

New solutions have been proposed to address problems with the Internet's interdomain routing protocol, BGP. Before their deployment, validation of incremental performance gains and backwards compatibility is necessary. For this task, the Internet's large size and complexity make all techniques but simulation infeasible. When performing large-scale network simulations, memory requirements for routing table storage can become a limiting factor. This work uses model reduction to mitigate this problem, with reduction defined in terms of the number of routers. Our framework uses path properties specific to interdomain routing to define the conditions of a path-preserving scale-down transformation. For implementation, vertex contraction and deletion were used to remove routers from a preliminary nominal network model. Vertex contraction was seen to violate the conditions of the transformation. A small subgraph from a measured topology is used for experimental validation. Routing tables are compared to show equivalence under the model reduction.

1 INTRODUCTION

Problems with the Internet's de facto interdomain routing protocol, BGP, have been well-documented (Feamster, Balakrishnan, and Rexford 2004a). Such issues have led to proposal of new or improved networking architectures (Feamster et al. 2004b, Li et al. 2005, Greenberg et al. 2005), but in today's conservative investment climate, even limited deployments of may incur excessive cost. Thus, a need exists to demostrate incremental performance gains and backwards compatibility before deployment of network technologies.

A routing protocol determines the paths (i.e., routes) followed by network traffic to reach their destinations. In the Internet, interdomain paths generally transverse one or more independently operated networks (i.e., autonomous systems). Each network's policy strongly influences these path definitions. An evaluation of BGP, or its replacement or improvement, needs to validate that resulting path calculations meet the intents of the individual networks' policies. These paths are stored in routing tables of Internet routers.

Interdomain route calculation is an operation distributed across all of the Internet's interconnected networks. It is possible that policy implementations at some distant network can affect local route calculations. Studying such activity is difficult, not only due to capturing the Internet's large size and heterogeneity, but also recreating the complex interactions of the routing protocol. Theoretical analysis is difficult, if not impossible, as complex routing protocols like BGP do not yet have analytical forms. Conversely, simulation approaches are more likely to succeed, with large-scale networks having been recently studied (Cowie, Nicol, and Ogielski 1999; Dimitropoulos and Riley 2004; Hao and Koppol 2003).

On the other hand, at an Internet scale, simulations can be limited by the large number of packet events associated with traffic flows, as well as the memory resources consumed by the routing tables (Nicol, Liljenstam, and Liu 2005). The latter is a significant concern as total memory estimates for the routing tables are on the order of 10Gb for a network of 10,000 BGP routers (10Mb memory per BGP router, 1000 bytes per route), which does not include the memory needed for the simulation code and its data structures (Nicol 2002).

1-4244-0501-7/06/$20.00 ©2006 IEEE

Assuming only one BGP router per autonomous system, simulation of today's Internet requires at least twice this memory estimate.

Current solutions to memory demands include distributed simulation and memory management techniques. The latter includes on-demand route calculations (Liljenstam and Nicol 2004; Riley, Fujimototo, and Ammar 2000) and memory sharing by the routes tables (Hao and Koppol 2003, Dimitropoulos and Riley 2004). Consideration of distributed simulation requires more computing hardware and tools (e.g., graph partitioners), specialized operator training, and has a smaller development and support community. The memory managements schemes are generally simulator specific. An alternative approach is to use a model with a lower number of routers, as long as such changes recreate the original large-scale path calculations. Model reduction requires no additional computing resource or operator training, and there are no simulator specific dependencies. Model reduction is complementary to distributed simulation and memory management techniques for lessening the memory demands of large-scale simulations.

Many network simulations use some form of model reduction, but none have the goal to preserve the calculations of interdomain routing. For example, it is common to use vertex-induced subgraphs from a larger Internet topology (Dimitropoulos and Riley 2004) as a reduced network model, but this approach may adversely affect the true, large-scale, interdomain route calculations due to deletion of neighboring policy interactions. In this paper, we use the path properties specific to interdomain routing and define conditions for a graph reducing transformation. We then demonstrate two ad hoc methods for its implementation and discuss the effects of each. Using a small subgraph from a measured topology, we simulated the original and reduced models. Comparison of simulated routing tables was performed and model equivalence is argued. We conclude with discussion of future work.

2 RELATED WORKS

In Kirshnamurthy et al. (2005), several graphical techniques were evaluated for reducing network topologies. These included deletion of vertices/edges, contraction of connected vertices, and inducing subgraphs based on breadth/depth first searches. Invariance of several graph theoretic measures (e.g., the power-law exponent of the graph's degree-frequency distribution) was used to quantify the fidelity of the model reduction. Similar work was performed in Lee, Kim, and Jeong (2005). Without knowing the effects of these graph-theoretic properties on interdomain routing, it is not clear such approaches would provide meaningful simulation results.

For the Modelnet testbed, a *distillator* tool was used to reduce network topologies to increase scalability (Vahdat et al. 2002). This approach partitioned the topology into multiple sets. The first set included all edge vertices, with subsequent sets including all unselected vertices that are one hop from the previous set. The amount of distillation defines the number of sets to be preserved, while the remaining sets are replaced with a full mesh of interconnections annotated with bandwidth, delay, and a loss rate attributes. Each mesh link represents an end-to-end path across the removed sets. Model reduction fidelity was determined by comparing bandwidth distributions generated by multiple TCP test flows.

Another model reduction approach was provided in Petit, Ammar, and Fujimoto (2005). Network experiments were first classified based on their composition of elastic (TCP) and non-elastic (UDP) traffic flows, which then determined the model reduction's fidelity metrics. Nodes (i.e., vertices) were then selected for removal if their outbound capacity exceeded their inbound capacity. After removal, surrounding links were rewired to preserve connectivity, and also annotated with capacity and propagation delay attributes such that the transmission delay per packet and traffic throughput were minimally affected. Measurements of (UDP) packet delay and/or (TCP) response time to web requests were used to quantify reduction fidelity.

Our work was similar in its approach to these latter two techniques. Selected vertices were removed and edges added to restore end-to-end connectivity. Differences arose as our context focused on interdomain routing protocols and not transport protocols. Our fidelity measures are based on the calculated paths within the network control plane, and not packet-based performance measures of data plane. Our work is complementary.

3 ABSTRACT MODEL OF BGP

A fundamental operation of computer networks is the transport of user data (i.e., traffic) between two systems. For packet networks, user data is segmented, tagged with a unique destination address, then moved along a path composed of various interconnected networking entities (e.g., routers, switches, transmission links). Such traffic-carrying paths are optimal in some sense, such as being the shortest or least used. For small networks, these paths can be manually defined, but for large networks, a routing protocol is used.

BGP is the de facto Internet interdomain routing protocol. Detailed and abstracted descriptions are available in (Rekhter and Li 1995, Stewart 1998) and (Griffin and Wilfong 1999) respectively. Following the latter, let $G = (V, E)$ be a simple, undirected, connected graph representing a static Internet interdomain topology. The vertices V are autonomous systems (AS) and the edges E represent their interdomain connections. Interdomain is a system-level abstraction, where paths begin and terminate at ASs. An AS is an independently operated network within the larger Internet, which is also assigned a unique 16-bit identifier (e.g., AS1239 references Sprint). When user traffic is transported, it follows a path composed of ASs and their interconnecting links.

Each vertex $v \in V$ receives (destination) reachability information contained in BGP route messages, r, from its connected neighbors. Let the route messages r be defined as the tuple

$$r = (\textbf{prefix}, \textbf{aspath})$$

where **prefix** and **aspath** are attributes. Let

- **prefix** represent a set of IP addresses, each a unique 32-bit number, which all have some number of leading bits in common. For example, 128.9.0.0/16 is a prefix representing all those 32-bit addresses that have the same leading 16 bits (i.e., 128.9).
- **aspath** be an ordered list of ASs to transverse in G to reach destination **prefix** from v.

The ordered list r.**aspath** $= (v_l \cdots v_1\ v_0)$ is path across l connected vertices in G, with $(v_{i+1}, v_i) \in E$ for integers $l > i \geq 0$. The number of vertices in r.**aspath** is the path's length. v_1 *originates* **prefix**, and does not need to visit any other vertex to reach **prefix**. v_1 has an implicit connection to **prefix**, a connection that is not captured in E. Only originating vertices can create route messages r. All others are copies made after receiving a route message r from a neighboring vertex.

Each neighbor of v can provide a single route message for some **prefix**. If v has m neighbors, then v can receive at most m messages, forming the set $R = (r_m, \cdots, r_2, r_1)$ of candidate paths to **prefix**. At each v, there is a set of candidate routes R for every **prefix**, and each r.**aspath** $\in R$ represents a unique path to that **prefix**.

BGP allows for the modification, addition, or deletion of route message attributes. These operations are defined on the edges $(u, v) \in E$. Let $policy(u \rightarrow v, r)$ represent a set of operations on the attributes of a route message sent from u to v. For example,

- when sending a route message to v, u adds (or *prepends*) itself to the beginning of the ordered list r.**aspath**, such that r.**aspath** becomes $(u\ v_l \ldots v_2\ v_1)$.
- v deletes all attributes of an incoming route message if $v \in r$.**aspath**. By deleting all attributes, the route message is effectively filtered and not added to v's set of candidate routes R.

As attribute modification can occur when a route message enters or exits a vertex, $policy(u \rightarrow v, r)$ can be decomposed into mutually exclusive sets $import(u \rightarrow v, r)$ and $export(u \rightarrow v, r)$. These two sets describe the operations on route messages when entering v or exiting u respectively.

After $policy(\cdot)$ operations, incoming route messages r are added to v's set of candidate routes R for r.**prefix**. Within each set R, all candidate routes are ranked according to their attributes. Higher ranking for shorter r.**aspath** length is common, but not absolute. $import(\cdot)$ can add attributes that assign higher preference. See Rekhter and Li (1995), Stewart (1998) for details on BGP's ranking function. The highest ranked route message r contains a path $p* = r$.**aspath** which defines v's *best* path (from v) to r.**prefix**. This particular route message r is then shared to all v's neighbors, subject to $policy(\cdot)$ operations. The set of all best paths determines how user traffic is engineered through the Internet. Clearly, policy defines a causality chain from modifying route message attributes, to influence on best path selection, to Internet traffic engineering.

3.1 Multiple Originating Autonomous Systems

To illustrate application of this abstract model, consider the following BGP issue. Any vertex w is allowed to originate a route message r for any **prefix**. Let w_1 and w_2 originate a route message r for the same **prefix**. Both w_1 and w_2 claim an implicit, direct connection to **prefix**, but assume that w_2's claim is not true.

Let w_1 and w_2 send their route messages r for **prefix** to their neighbors, which are then sent to their neighbors, and so on. It is then likely that some vertex v will receive different route messages r for the same **prefix**, but whose r.**aspath**s originated at different locations (i.e., $v_0 = w_1$ and w_2). Let both route messages r be added to the set of candidate routes R for **prefix** at v. After the ranking process, one of these two route messages will be chosen as best. An issue then arises if user traffic destined for **prefix** follows the path to w_2. This traffic may be lost (i.e., blackholed) or experienced increased delay since w_2 has no direct connection to **prefix**.

When multiple vertices claim to be the origin of the same **prefix**, a Multiple Originating Autonomous System (MOAS) conflict is said to exist (Zhao et al. 2001). In practice, this results not from a flaw in the BGP protocol, but from equipment misconfigurations, faults, or Internet users with malicious intents. Using our example above, if w_2 cannot reach **prefix**, then the network routing or traffic engineering to **prefix** has been partitioned, as some vertices use valid paths from w_1, while the rest use invalid paths from w_2.

The BGP protocol has no inherent mechanism to identify nor remedy this situation. We suggest that the use of large-scale interdomain routing simulations will enable better study of MOAS conflicts and provide a suitable testing platform for their potential solutions. Toward this goal, counting the number of vertices whose best path originated from the different vertices is a simple metric to quantify the MOAS conflict (Carl et al. 2006).

4 MODEL REDUCTION USING EQUILAVENCE PRINCIPLES

Model reduction can be developed from linear circuit theory's classical principle of Thevenin (or Norton) equivalence: Under certain conditions, one can replace a subset of a complicated network, by a simpler form, while preserving certain properties from the point of view of the rest of the network. In linear circuit theory, a subnetwork of linear components (e.g., resistor, capacitors, voltage and current sources) can be replaced by single voltage source and impedance while preserving the voltages and currents at all remaining components. Queuing theory has a similar theorem (by Chandy-Herzong-Woo) for replacing a subset of interconnected queues.

Using these concepts, we state that if a given Internet interdomain graph G has non-pathological policy conditions (i.e., BGP $policy(\cdot)$ operations that result in stable, non-oscillatory, sets of candidate routes (Varadhan, Govindan, and Estrin 2000)), and if we can remove vertices from G in such a way that important path properties (i.e., the number, length, and composition of candidate paths) for each **prefix** are preserved, then we can claim model equivalence. We note that dynamics are generally not preserved under equivalence theorems, e.g., Norton's theory for queues does not guarantee equivalence in sojourn times (Walrand 1988), and are outside the scope of this work.

4.1 Path Preservation

Our model reduction is defined as a path preserving scale-down transformation $PPSD : G = (V,E,P) \to G' = (V',E',P')$ where $\mid V' \mid < \mid V \mid$. P is the set of $policy(u \to v)$ for all $(u,v \in V \mid (u,v) \in E)$. Let X be the set of vertices to be removed from G, i.e., $V' = V \setminus X$. For any two vertices s,t in G, let $P_{t,s}$ be the set of candidate paths from t to s, for a **prefix** originated by s. Similarity, let $P'_{t,s}$ be the set of candidate paths from t to s for the same **prefix** originated by s in G'. The graph transformation PPSD is path preserving if the following conditions hold:

- *Number of Paths*: Given any two vertices s,t, with $s \neq t$ and $deg(t)$ being the number of edges incident on t, the set $P_{t,s}$ contains at most $deg(t)$ candidate paths from t to s, since only one route message r can arrive per edge into t. In G', the number of candidate paths between t and s, $\mid P'_{t,s} \mid$, should not be larger than $\mid P_{t,s} \mid$.
- *Mapping of Candidate Paths*: For any path $p \in P_{t,s} = (v_l \cdots v_1 \; v_0)$, with $v_0 = s, v_i \neq v_j, l \geq j, i \geq 0$, there is zero or one corresponding path $p' \in P'_{t,s}$ with $p \cup p' = p$, as well as the following two length conditions:
  1. $\mid p' \cap p \mid = \mid p \mid - \mid X \cap p \mid$
  2. $\mid p' \mid = \mid p \mid$.

  For best paths ($p* \in P_{t,s}$), $p* \to p'$ must always exist, and meet these conditions.
- *Vertex Ordering*: Define an order relation ($\prec$) on $v_i \in p = (v_l \cdots v_1 \; v_0)$ and $v'_j \in p' = (v'_{l'} \cdots v'_1 \; v_0)$, such that $v_i \prec v'_j, i \geq j$, if v_i is reached before v'_j when simultaneously following both paths p, p' after starting from v_0. For example, $v_i \preceq v'_j$ when $p = (3 \; 2 \; 1)$, $p' = (3 \; 1 \; 1)$, and $X = (2)$. Under mapping $PPSD : p \to p'$, $v_i \preceq v'_j$ and $v_i = v'_i$ when $v_i \notin X$.

5 SCALE-DOWN

To implement the PPSD transformation, ad hoc vertex aggregation and deletion techniques were considered. Vertex aggregation chooses a set of neighboring nodes, then contracts this set into a single vertex while maintaining all exterior edge connections. Figure 1(b) shows the contraction of vertices x and o into o''. Vertex deletion removes a set of vertices while adding edges to preserve connectivity. Figure 1(c) shows the removal of vertex x and the addition of edges $(x_2,o),(x_2,t_x),\cdots,(t_x,o)$.

Both approaches have different effects. Consider the original path between t_o and s, denoted as $p_{t_o,s} = (o \; o_1 \; o_2 \; s)$, and between t_x and s, denoted as $p_{t_x,s} = (x \; x_2 \; s)$. Associate these two paths with the same **prefix** originated at s. Furthermore, assume $p_{t_o,s}$ and $p_{t_x,s}$ were the best paths to s from t_o and t_x respectively. Under contraction of vertices x and o, the paths $p'_{t_o,s}, p'_{t_x,s}$ merge at the union point o'' in Figure 1(b). Therefore, the set of candidate paths from o'' to s (for **prefix**) will include $(o_1 \; o_2 \; s)$ and $(x_2 \; s)$. Assume the best path from o'' to s, $p_{o'',s}*$, will be selected from this candidate set, and then sent to both t_o and t_x. At t_o and t_x, their candidate paths will include $p_{o'',s}*$ as a subpath. Explicitly $p'_{t_o,s} = (o'' \; p_{o'',s}*)$ and $p'_{t_x,s} = (o'' \; p_{o'',s}*)$. The best path mapping conditions of $p_{t_o,s} \cup p'_{t_o,s} = p_{t_o,s}$ and $p_{t_x,s} \cup p'_{t_x,s} = p_{t_x,s}$ cannot be simultaneously meet. Now consider the vertex deletion approach in Figure 1(c). $p'_{t_o,s}$ and $p'_{t_x,s}$ no longer are forced to merge. Alternative connectively exists which can preserve the path mapping conditions of Table 1.

The path length conditions also need to be met after vertex deletion. Consider the three vertice network in Figure 2(a). The path from t to s can be either direct or through x. Suppose under stable conditions, s and t determine that their direct path is best. Given this static setting, x can be removed without further modification, as it is not required for path connectivity between s or t. All conditions of Table 1 have been trivially met as $p_{t,s} = p'_{t,s}$.

Table 1: Path Properties Under Transformation $PPSD : p \rightarrow p'$

Path Property	G	G'
Number of Paths	$0 <\mid P_{t,s} \mid \leq deg(t)$	$0 <\mid P'_{t,s} \mid \leq deg(t)$
Path Mapping	For each $p \in P_{t,s}$	zero or one $p' \in P'_{t,s}$ such that $p \cup p' = p$.
Path Lengths	For each $p \in P_{t,s}$	zero or one $p' \in P'_{t,s}$ such that $\mid p' \cap p \mid = \mid p \mid - \mid X \cap p \mid$ and $\mid p' \mid = \mid p \mid$.
Best Path	For each $p{*} \in P_{t,s}$	there is one and only one $p'{*} \in P'_{t,s}$ such that $\mid p'{*} \cap p{*} \mid = \mid p{*} \mid - \mid X \cap p{*} \mid, \mid p'{*} \mid = \mid p{*} \mid$
Vertex Ordering	For each $p = (v_l \cdots v_1 v_0) \in P_{t,s}$	and its corresponding $p' = (v'_{l'} \cdots v'_1 v'_0)$, $v_i \preceq v'_j, i \geq j$, and $v_i = v'_i$ when $v_i \notin X$.

Now suppose that t and s both have their best paths include x. First, upon removal of x, connectivity between t and s needs to restored. In Figure 2(b), this is satisfied by adding another direct edge between t and s denoted by $(s,x)(x,t)$. Now consider the path properties along this added edge. Relative to original $p_{t,s} = (x\ s)$, the vertex ordering condition is satified, but the path's length is a one less due to the deletion of x. Compensation is necessary to 'inflate' the path length, which can be accomplish through an *export* operation. Here, $(prepend\ s)$ is added to $export(s \rightarrow t)$ at s. With this $policy(\cdot)$ addition, $p_{t,s} = (x\ s) \rightarrow p'_{t,s} = (s\ s)$. The path length conditions of Table 1 are now equal. A final step then removes any multiple edges giving Figure 2(c).

Inherent to our approach, we needed to a priori know the best path between two vertices neighboring a vertex x selected for deletion. To accomplish this task, we used existing path algebra techniques, but developed a simple path ranking function. First we extracted a subgraph containing all neighboring vertices of x. Then all elementary paths were computed using techniques described in Carré (1979). Invalid paths based on a BGP path algebra (Sobrinho 2005) were then eliminated. The resulting paths form the sets of candidate paths between all vertex pairs. Finally, within each set, all candidate paths were evaluated by our ranking function for determination of the best path.

6 MODELING SIDE EFFECTS

Our transformation produces other effects while meeting our path preserving conditions. First, vertex degrees can increase. In Figure 1, after deletion of x, vertices o, x_2, and t_x acquired more edge connections. The graph's degree distribution tends toward uniform, since the edge connectivity around the deleted vertex becomes fully meshed.

A secondary effect is seen in user traffic redistribution over the edge set E'. In Figure 1(a), assume user traffic follows the paths $p_{t_o,x_2} = (o\ x\ x_2)$ and $p_{t_x,x_2} = (x\ x_2)$, which shares the same subpath $p_{x,x_2} = (x_2)$. After deletion of x, user traffic from t_o and t_x to x_2 will follow $p'_{t_o,x_2} = (o\ x_2\ x_2)$ and $p'_{t_x,x_2} = (x_2\ x_2)$ respectively. The previous aggregated user traffic over subpath p_{x,x_2} has been de-aggregated over disjoint edges (o,x_2) and (t_x,x_2). Note, the total traffic arriving or leaving any vertex is unchanged.

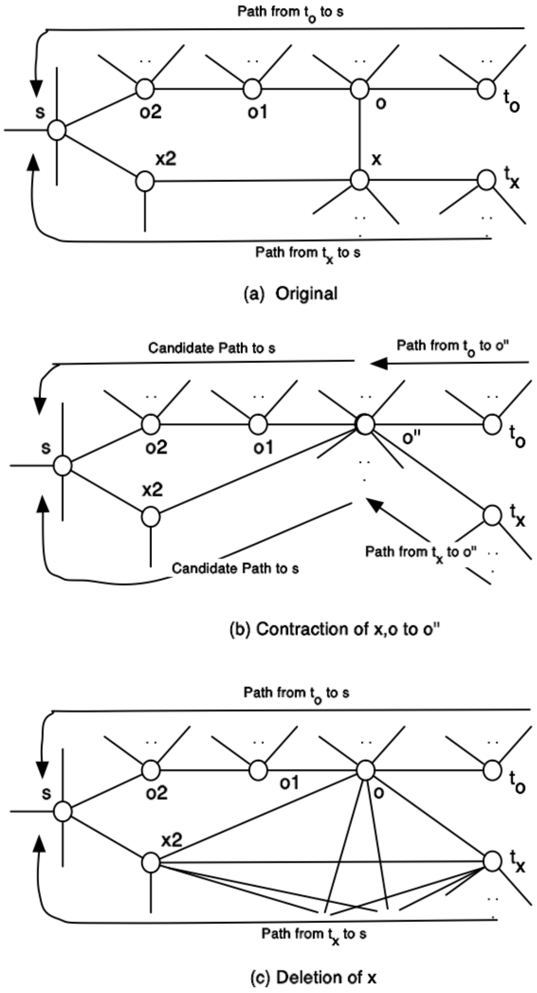
7 EXAMPLE

Experiments studying Internet interdomain routing protocols requires extensive specification, including an interdomain topology, routing policy information, accurate modeling of the routing protocol, and performance measures. Realistic topologies should be used since network topology has been found to influence BGP performance (Griffin and Premore 2001, Labovitz et al. 2001). A common source of interdomain topology information is the University of Oregon's Routeviews Project (<http://www.routeviews.org>).

Routing policy also needs to represented, since policy can influence traffic engineering. Unfortunately, network interdomain policies are proprietary to the privately operated ASs. The current state of the art is to assume a set of stable routing policies based on the business relationships between neighboring ASs. Inter-AS business relationships can be inferred from measured interdomain topologies using a heuristic algorithm (Gao 2001).

As an example of our model reduction, Figure 3(a) shows a subgraph, $G_s(V,E,P)$, extracted from the Routeviews snapshot of January 1, 2003. Each vertex $v \in V$ is labeled by its 16-bit AS identifier and represents a single AS policy domain executing the BGP routing protocol. No other routing protocols (e.g., iBGP, OSPF) are present. Routing policy at each vertex, $policy(u \rightarrow v)$ for all $(u,v) \in E$, follows from the stable recommendations of (Gao and Rexford 2000) based on the subgraph's inferred AS business inter-relationships.

Vertices AS701 and AS11422 were selected for removal. Figure 3(b) shows the reduced model after vertex deletion. Several new edges, such as (AS4,AS6461), (AS13788,AS6461) and (AS23165,AS6461), have been added. Edges which required *prepend* operations to artificially inflate the path lengths are designated by "+". For example, upon removal of AS11422 and AS701, the path lengths from AS6461 to AS4, AS13788, and AS23165 required inflation.



(a) Original

(b) Contraction of x,o to o''

(c) Deletion of x

Figure 1: Contraction vs. Deletion

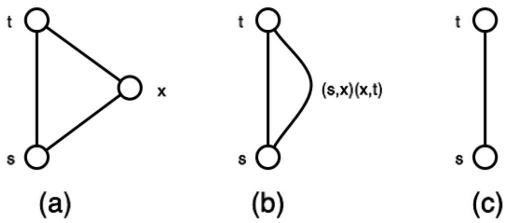


Figure 2: Scale-Down using Node Deletion

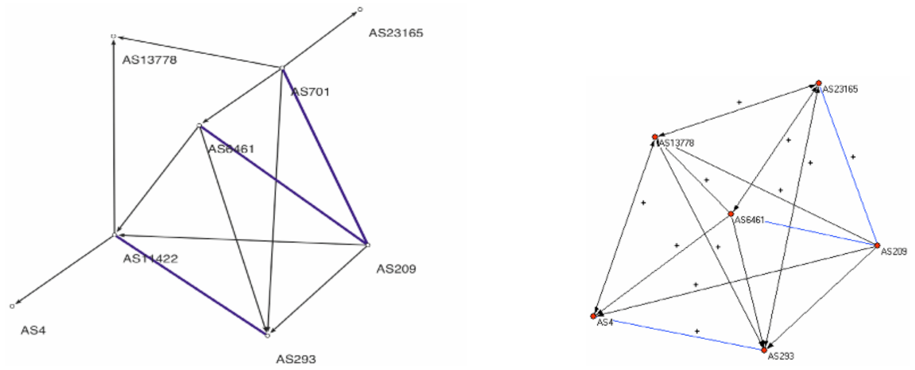
We then constructed a simulation experiment where AS4 originated (prefix) 128.9.0.0 and AS23165 originated 65.218.250.0. BGP++ (Dimitropoulos and Riley 2004) was used to simulate both original and reduced models. After 3000 simulated seconds, the candidate paths (i.e., BGP routing tables) for each vertex were logged. This output containee all received route messages *r*, and their attributes *r*.**prefix** and *r*.**aspath**. Table 2 illustrates the routing tables for both the original and reduced model. The corresponding best path is denoted with an "*".

Evaluation of Table 2 showed that each path condition of Table 1 has been satisfied. The number of candidate paths per prefix in the reduced model does not increase. For each path in the original model, there is at most one path with the same vertex ordering and path length in the reduced model. There was always one best path (denoted by *) per prefix. In those cases were a path in the original model does not have a corresponding path in the reduced model, it is noted that the original path was not designated as best. This loss of model fidelity is acceptable, as user traffic is minimally influenced by such paths, assuming the experiment's topology and policy operations remain static. With the removal of two vertices, the number of paths in the reduced model drops from 28 to 19. These routing tables will require less simulation memory. There will also be no data structures and packet events dedicated to the BGP simulation threads for non-existent vertices AS701 and AS11422.

8 FUTURE WORK

In our examples, the vertices selected for removal were for illustrative purposes only. In practice, the set of vertices selected for removal requires systematic definition. For example, in Weaver et al. (2004) scaled-down experiments removed IP address space that proportionally preserved key attack parameters such as number of systems infected and worm scans per second. For MOAS experiments, vertices for removal should be selected such that the resulting network partitions remain proportionally sized. After we formulate a approach for this task, we will extend the scale of our MOAS conflict simulations (Carl et al. 2006).

Only static experimental conditions were assumed. More realistic experiments will require support for dynamic events, such as topology and policy changes. Furthermore, since the number of edges traversed by route messages can be lessened by the model reduction (which we compensated with path inflation), route messages can arrive at neighboring vertices earlier. Link delays and protocol timers need to be appropriately modified. This is important for studying BGP problems dependent on the relative arrival of route messages, such as convergence time (Griffin and Premore 2001) and non-deterministic routings (Griffin and Huston



(a) Original **(b) Scaled-Down**

Figure 3: Example

Table 2: Simulated BGP Route Tables

AS	Prefix	Original Paths	Scaled-Down Paths
6461	65.218.250.0/24	* 701 23165	* 23165 23165
6461	128.9.0.0/16	209 11422 4	209 4 4
6461	128.9.0.0/16	* 11422 4	* 4 4
4	65.218.250.0/24	* 11422 209 701 23165	* 209 209 23165 23165
4	128.9.0.0/16	* (originated)	* (originated)
293	65.218.250.0/24	209 701 23165	209 23165 23165
293	65.218.250.0/24	6461 701 23165	6461 23165 23615
293	65.218.250.0/24	* 701 23165	* 23165 23165
293	128.9.0.0/16	701 6461 11422 4	
293	128.9.0.0/16	6461 11422 4	6461 4 4
293	128.9.0.0/16	209 11422 4	209 4 4
293	128.9.0.0/16	* 11422 4	* 4 4
23165	65.218.250.0/24	* (originated)	* (originated)
23165	128.9.0.0/16	* 701 6461 11422 4	* 6461 6461 4 4
209	65.218.250.0/24	* 701 23165	* 23165 23165
209	128.9.0.0/16	701 6461 11422 4	
209	128.9.0.0/16	6461 11422 4	6461 4 4
209	128.9.0.0/16	* 11422 4	* 4 4
13778	65.218.250.0/24	11422 209 701 23165	209 209 23165 23165
13778	65.218.250.0/24	* 701 23165	* 23165 23165
13778	128.9.0.0/16	701 6461 11422 4	
13778	128.9.0.0/16	* 11422 4	* 4 4
11422	65.218.250.0/24	6461 701 23165	-
11422	65.218.250.0/24	* 209 701 23165	-
11422	128.9.0.0/16	* 4	-
701	65.218.250.0/24	* 23165	-
701	128.9.0.0/16	* 6461 11422 4	-
701	128.9.0.0/16	209 11422 4	-

2005). These advanced aspects of simulation experiments are subjects of future investigation.

ACKNOWLEDGMENTS

This work was supported by the Applied Research Laboratory's Exploratory and Foundational Graduate Student Research Program and the DHS/NSF EMIST/DETER project.

REFERENCES

Carl, G., G. Kesidis, B. Madan, and S. Phoha. 2006, June 15-16. Preliminary BGP multiple-origin autonomous systems (MOAS) experiments on the DETER testbed. In *Deter Community Workshop*.

Carré, B. 1979. *Graphs and networks*. Clarendon Press.

Cowie, J. H., D. M. Nicol, and A. T. Ogielski. 1999. Modeling the global Internet. *Computing in Science and Engineering* 1 (1): 42–50.

Dimitropoulos, X. A., and G. F. Riley. 2004. Large-scale simulation models of BGP. In *MASCOTS*, 287–294.

Feamster, N., H. Balakrishnan, and J. Rexford. 2004a, November. Some foundational problems in interdomain routing. In *ACM SIGCOMM Workshop on Hot Topics in Networking (HOTNETS-III)*.

Feamster, N., H. Balakrishnan, J. Rexford, A. Shaikh, and K. van der Merwe. 2004b, September. The case for separating routing from routers. In *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*.

Gao, L. 2001. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking* 9 (6): 733–745.

Gao, L., and J. Rexford. 2000, June. Stable Internet routing without global coordination. In *Measurement and Modeling of Computer Systems*, 307–317.

Greenberg, A., G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. 2005. A clean slate 4d approach to network control and management. *SIGCOMM Comput. Commun. Rev.* 35 (5): 41–54.

Griffin, T., and G. Huston. 2005, October 16. BGP wedgies. RFC 4264 available at `<http://www.ietf.org/rfc/rfc4264.txt>`.

Griffin, T. G., and B. J. Premore. 2001, November. An experimental analysis of BGP convergence time. In *Proceedings of ICNP 2001*, pages 53–61.

Griffin, T. G., and G. T. Wilfong. 1999, August. An analysis of BGP convergence properties. In *Proceedings of SIGCOMM*, 277–288.

Hao, F., and P. Koppol. 2003. An internet scale simulation setup for BGP. *SIGCOMM Comput. Commun. Rev.* 33 (3): 43–57.

Kirshnamurthy, V., M. Faloutsos, M. Chrobak, L. Lao, J. Cui, and A. Percus. 2005, May 2-6. Reducing large internet topologies for faster simulations. In *Networking 2005*.

Labovitz, C., A. Ahuja, R. Wattenhofer, and V. Srinivasan. 2001, April. The impact of internet policy and topology on delayed routing convergence. In *INFOCOM*, 537–546.

Lee, S. H., P.-J. Kim, and H. Jeong. 2005, May 10. Statistical properties of sampled networks. Technical report, arXiv:cond-mat/0505232.

Li, Z., P. Mohapatra, and C.-N. Chuah. 2005, May. Virtual multi-homing: On the feasibility of combining overlay routing with BGP routing. In *NETWORKING*, 1348–1352.

Liljenstam, M., and D. Nicol. 2004, December. On-demand computation of policy based routes for large-scale network simulation. In *Proceedings of the 2004 Winter simulation Conference*.

Nicol, D. 2002, January. Challenges in using simulation to explain global routing instabilites. In *2002 Conference on Grand Challenges in Simulation*.

Nicol, D. M., M. Liljenstam, and J. Liu. 2005, December. Advanced concepts in large-scale network simulations. In *Proceeding of the Winter Simulation Conference*.

Petit, B., M. Ammar, and R. Fujimoto. 2005, July. Scenario-specific topology reduction in network simulations. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECT)*.

Rekhter, Y., and T. Li. 1995. A border gateway protocol. RFC 1771 (BGP version 4) available at `<http://www.ietf.org/rfc/rfc1771.txt>`.

Riley, G., R. Fujimototo, and M. Ammar. 2000, August-September. Stateless routing in network simulations. In *Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*.

Sobrinho, J. L. 2005. An algebraic theory of dynamic network routing. *IEEE/ACM Transactions on Networking* 13 (5): 1160–1173.

Stewart, J. 1998. *BGP4: Interdomain Routing in the Internet*. Addison-Welsey.

Vahdat, A., K. Yocum, K. Walsh, P. Mahadevan, D. Kostić, J. Chase, and D. Becker. 2002. Scalability and accuracy in a large-scale network emulator. *SIGOPS Oper. Syst. Rev.* 36 (SI): 271–284.

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

Walrand, J. 1988. *Introduction to queuing networks*. Prentice Hall.

Weaver, N., I. Hamadeh, G. Kesidis, and V. Paxson. 2004. Preliminary results using scale-down to explore worm dynamics. In *WORM '04: Proceedings of the 2004 ACM Workshop on Rapid Malcode (WORM)*, 65–72.

Zhao, X., D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. 2001, November. An analysis of BGP multiple origin AS (MOAS) conflicts. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 31–35.

AUTHOR BIOGRAPHIES

GLENN CARL is a Phd student in EE at the Pennsylvania State University. He is currently a graduate research assistant for the Applied Research Laboratory at Penn State. From 1995 to 2003, he held engineering positions in the semiconductor and telecommunications industries. His research interests include large-scale network simulation and security. He can be reached at <gmc102@psu.edu>.

GEORGE KESIDIS received his M.S. and Ph.D. in EECS from U.C. Berkeley in 1990 and 1992 respectively. He was a professor in the E&CE Department of the University of Waterloo, Canada, from 1992 to 2000. Since April 2000, he has taught in both the CS&E and EE Depts of the Pennsylvania State University. His research experience spansseveral areas of computer/communication networking including security, incentive engineering, efficient simulation, and traffic engineering. Currently, he is a senior member of the IEEE and the TPC co-chair of IEEE INFOCOM 2007. He can be reached at <gik2@psu.edu>.

SHASHI PHOHA has research interests in computational sciences that enable dependable distributed automation of multiple interacting devices over ad hoc and long haul networks. Since 2004, she has been the Director of a premiere national laboratory, the Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST). Since 1991, she has been a Professor of EE at the Pennsylvania State University and the Director of the Division of Information Sciences and Technology at the Applied Research Laboratory. Prior to that, she was the Director of Information Systems Analysis Division of the Computer Sciences Corporation where she led the development of the Global Transportation Network Architecture for DoD. She worked in Command, Control, Communications and Intelligence Systems at ITT and at the MITRE Corporation. She was awarded the 2004 Technical Achievement Award by the IEEE Computer Society. She was Guest Editor of Special Issues of IEEE Transactions (TMC), an associate editor of the IEEE Transactions on Systems, Man, and Cybernetics for four years and is editor of the International Journal of Distributed Sensor Networks. She received her M.S. in Operations Research from Cornell University (1973) and Ph.D. from Michigan State University (1976). She can be reached at <sxp26@psu.edu>.

BHARAT MADAN is currently with the Applied Research Lab of the Penn State University as Head of the Distributed Systems Department. His research interests include sensor, mobile, and wireless networks, intrusion tolerant secure systems, distributed systems, high performance computing, data structures and signal processing. Until 1996, he was with IIT Delhi as full Professor in the CS&E department He was also concurrently the Head of the Computer Center, IIT Delhi from 1992-1995. He has also held visiting positions at Loughborough Univ. of Tech., UK (1976-77), Naval Postgraduate (1984-1985), Univ. of Delaware (1988-89) and was a Visiting Scholar with the E&CE Department, Duke Univ (2002-2004). He is also an Adjunct Professor with the CS Department of the Old Dominion Univ., Norfolk since 1992. He also spent 6 years (1996-2002) at IBM and Ericsson conducting R&D. He can be reached at <bbm2@psu.edu>.