Proof of an Interdomain Policy

A Load-Balancing Multi-Homed Network

Andreas Voellmy
Yale University
New Haven, CT, U.S.A.
andreas.voellmy@yale.edu

ABSTRACT

Configuration of interdomain routing policies is notoriously difficult, despite their relatively simple structure. We believe that this difficulty arises in part due to the gap between a network's interdomain traffic goals and the interdomain routing policies which implement them. The gap arises because BGP policy accomplishes network goals indirectly, through interaction with the forwarding plane, and because these goals can often only be met when neighboring networks agree to implement certain policies.

We present a case study of the formal verification of a real world BGP configuration, a load-balancing multi-homed network discussed in a book on BGP configuration. We invent a formal specification from the informal description of the example and prove that the BGP policy meets this specification. The goal of this study is to make explicit and precise, the principles used in reasoning about BGP policy. We aim to show not that verification is easy, but rather that this reasoning is far from trivial and that the difficulty in configuring BGP routers may lie not in writing policy, but in understanding the effects of this policy in a complex environment. The study also illustrates some of the benefits of formal specification and verification for BGP policy, namely that such efforts help in discovering ambiguities, inconsistencies and implicit assumptions in specifications and policies.

The case study and proofs have been formalized in the Isabelle/HOL theorem prover and are available on the web. [1]

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*Correctness proofs, Formal methods*; C.2.2 [**Computer-Communication Networks**]: Routing protocols

General Terms

Verification, Languages

[1] http://www.haskell.org/YaleHaskellGroupWiki/Networks

SafeConfig'09, November 9, 2009, Chicago, Illinois, USA.

1. INTRODUCTION

Configuration of interdomain routing policies is notoriously difficult, despite the relatively simple structure of routing policies. Indeed, it has been estimated that almost half of all network outages are a result of network misconfigurations [1].

We believe that this difficulty arises in part due to the gap between a network's interdomain traffic goals and the interdomain routing policies which implement them. The gap arises because BGP policy accomplishes network goals indirectly, through interaction with the forwarding plane, and because these goals can often only be met when neighboring networks agree to implement certain policies.

In this paper, we analyze a simplified version of a multi-homed load-balancing policy presented in Zhang et al [3], a Cisco book on BGP configuration. We choose this example because it illustrates that the goal of BGP policy is often to achieve a certain effect on traffic entering and leaving a network, whereas the BGP policy directing route choices is simply a mechanism to achieve these goals. In addition, we hope that this case represents a typical and realistic scenario for a multi-homed autonomous system. By formally proving that the policy achieves its goals, we demonstrate the size of the gap between mechanism and goal in an arguably real-world example.

In order to accomplish our aims, we develop a highly simplified configuration verification method consisting of:

- a specification language to describe the behavior of various aspects of the network, including forwarding and neighbor behavior;
- a policy language for describing the behavior of routers;
- first-order axioms describing the relationships between the various forwarding, routing, and policy components, thereby allowing formal proofs of policy correctness to be given and verified.

Following this, we present the details of the load-balancing multi-homed network example. We specify and implement the example network and bridge the gap with a formal proof of correctness, which has been implemented and verified in the Isabelle/HOL theorem prover. In the process, we identify an assumption underlying the correctness proof which is not mentioned in Zhang et al [3], the source of the case

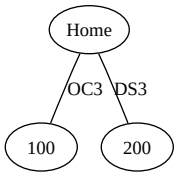

Figure 1: The BGP connections for the example specification.

study. That this assumption was uncovered during the process of formal verification is an indication of the benefits formal methods and tools may bring to the practice of interdomain policy configuration.

2. THE EXAMPLE - INFORMALLY

In this section we informally introduce a simplified version of the scenario presented in Zhang et al [3], a Cisco book on BGP configuration. The example is for the configuration of a network which is *multi-homed* to two providers, i.e. it has two connections to external providers. The network is connected to AS 100 by a high-bandwidth OC-3 (Optical Carrier, 155 Mbit/s) link and to AS 200 by a lower bandwidth DS-3 (Digital signal, 45 Mbit/s) link, as shown in Figure 1.

The intended behavior is to forward most traffic over the high-bandwidth OC-3 link to AS 100, while still utilizing the lower-bandwidth DS-3 link to take advantage of the available bandwidth. To achieve this, the network aims to send all traffic that is destined to a customer of AS 200, i.e. a network that pays AS 200 for interdomain connectivity, over the DS-3 link, while all other traffic, including traffic to customers of 100, should traverse the OC-3 link. This presumably will result in most traffic using the OC-3 link through AS 100, since most destinations will not be customers of AS 200. On the other hand, the traffic that will be sent over the DS-3 link will be destined to customers of AS 200, and thus this path will be shorter than a path through AS 100. This policy, therefore attempts to achieve some load balancing while also favoring short routes.

In order to implement this goal, the network will need to differentiate between routes that are for customers of AS 100 or AS 200 and those that are not customers. Rather than fixing a static set of networks that are customers of each provider AS 100 and 200, the network requests what is known as *partial and default routes* from both AS 100 and AS 200, meaning that AS 100 and AS 200 will both advertise a route with a default prefix and in addition will announce routes with prefixes belonging to all and only customers of AS 100 and AS 200, respectively. In other words all routes with non-default prefixes announced by 100 and 200 will be customers of 100 and 200 respectively, and conversely, all customers of 100 and 200 will have a route with non-default prefix announced. This scheme allows the home network to determine when a route is for a customer of 100 or 200, namely when the route carries a non-default prefix and is advertised by 100 or 200, respectively, without statically specifying the customers of 100 and 200.

The policy given in Zhang et al [3] is to rank routes learned over the OC-3 at a high preference level, $high = 120$, while routes learned over the DS-3 are given preference $low = 100$. Zhang et al justify this policy by explaining that this will have the effect of preferring the OC-3 link for the default route and routes to customers of AS 100 (since routes for customers of AS 100 will be announced over the OC-3 link). On the other hand, routes over the DS-3 link will be preferred for networks that are customers of 200 and not customers of 100.

Unfortunately, the example as stated above has several problems. First of all, the stated goal is impossible to realize if there exists a single network that is a customer of both 100 and 200. This is because, with the network given, traffic for any particular address will leave the network over only one of the two links. However, the goal states that traffic for an address that belongs to a customer of both 100 and 200 should traverse both the OC-3 link and the DS-3 link, which is impossible. Therefore, the specification needs to be revised, and there are several possible ways to do so. One possibility is to add an extra assumption regarding the impossibility of a network being a customer of both 100 and 200. Another possibility is to weaken the specification so that it only require traffic for customers of 100 that are not customers of 200 to traverse the OC-3 link, and similarly for traffic for customers of 200 that are not customers of 100 to traverse the DS-3 link. With this revision, the specification places no requirement on traffic for customers of both 100 and 200. A third possibility is to require all traffic to customers of 100 to traverse the OC-3 link, while only traffic for customers of 200 that are not customers of 100 to traverse the DS-3 link.

We pursue the second and third options in the verification of the example in Section 4. Unfortunately, we find that we are unable to verify the specification as stated in the third option, because it may in fact not hold. We can show this by the following counterexample to the specification. Suppose there is a network *asn* that is a customer of both 100 and 200. Hence, by our assumptions about AS 100 and AS 200, AS 100 will announce a route r with prefix p to network *asn* and AS 200 will announce a route r' with prefix p' to network *asn*. Suppose further that prefix p' is more specific (i.e. a longer prefix) than p and that r and r' are the best routes for prefixes p and p' respectively. Since BGP installs the best route for each prefix in the forwarding table, BGP will install both r and r', even though r is ranked at 120 and r' is ranked at 100. Now consider an address a contained in network *asn*, and suppose it is in prefix p'. Traffic to address a will then be forwarded according to route r', since p' is more specific than p, and hence will traverse link DS-3. This violates the specification as stated in the third option. We present the details of this example in Section 4.1 in terms of the language we present below.

In order to achieve the third specification, then, we need to revise it by adding an assumption asserting that, if a network is a customer of both AS 100 and AS 200 and if it is announced by both AS 100 and AS 200, then the routes announced carry the same prefix. With this assumption we are able to verify the third specification.

3. LANGUAGE

In this section, we develop a basic formal language in which statements describe forwarding behavior, BGP state and BGP policy. With this language, we can specify desired network behaviors as well as policy implementations. The language uses set theory and natural numbers in a basic way, and we take as given the theories describing the behavior of these domains.

The model we present here is highly simplified and does not model BGP accurately. Some of the simplifications we make include a simple decision process, the assumption that every router has the same policy and that this combined with the simple decision process results in every router having the same ordering over routes, and that every router learns of all routes announced to the network. These simplifications only make reasoning about BGP policies easier, which enhances our claim that reasoning about the effects of BGP policy can be very difficult.

To begin with, our language includes a set of addresses, *Addresses*, and a finite set of address prefixes, *Prefixes*. Although Internet addresses are 32-bit integers, we model them simply as natural numbers, i.e. $Addresses = \mathbb{N}$. An Internet prefix p is a binary sequences of length at most 32 and denotes the set of addresses a whose binary expansion has p as a prefix. Prefixes have an important property: given an address a and a collection of prefixes P containing a, there exists a $\subseteq$-least, i.e. most specific, prefix containing address a. This property holds because each $p \in P$ is a prefix of a and therefore if $p, p' \in P$ then either p is a prefix of p' or vice versa. Since they are linearly ordered by prefix and the collection of prefixes is finite, there is a minimum prefix. This property is the only prefix-specific property we will need, so we model prefixes abstractly as a set *Prefixes*, whose elements are subsets of addresses, which satisfies the abovementioned property. This allows us to use the set-theoretic predicates $\in$ to assert that an address is contained in a prefix and $\subseteq$ to assert containment of prefixes, and we preserve the above property by asserting it as an axiom:

Axiom 1. For any set of prefixes $P \subseteq Prefixes$ and address $a \in Addresses$,

$$(P \neq \emptyset \wedge (\forall p \in P \rightarrow a \in p)) \rightarrow \exists p \in P.[\forall p' \in P.p \subseteq p']$$

We define a default prefix, 0.0.0.0/0, as the set of all addresses, i.e. $0.0.0.0/0 = Addresses$.

The language includes a finite set of neighboring networks *Neighbors*, and a finite set of links *Links*, and a function *linkTo* such that $linkTo(l)$ is the neighbor which is connected to link l.

BGP routes are modelled as the set $Route = Prefixes \times Seq(AS) \times Links$, where $Seq(AS)$ is the set of finite sequences of AS numbers. We use a function *length* which gives the length of a sequence.

The language includes a finite set $Announce \subseteq Neighbors \times Route$ that models the routes announced by neighbors, and we assert the finiteness of *Announce* as an axiom:

Axiom 2. $Announce \subseteq Neighbors \times Route$ and is finite.

We define the set *Knows*, the set of BGP routes known to the network (from all neighbors), in terms of *Announce* as:

Definition 1. $Knows = \{rte \mid \exists nbr\ .\ Announce(nbr, rte)\}$

We require that any route announced by a neighbor traverse a link to that neighbor, and we assert this with an axiom:

Axiom 3.

$$(nbr, (s, p, l)) \in Announce \rightarrow linkTo(l) = nbr$$

Although BGP policy and route choice are complex in practice, we provide a highly simplified model which has just enough detail to specify and verify our case study. For our purposes, we need a function $rank : Route \rightarrow \mathbb{N}$ which gives the numeric preference value for each route, and two tie-breaking orderings - a strict ordering $<_{paths}$ on paths that is also linear for any set of paths having some fixed length (e.g. the ordering could compare the AS number of the first differing AS in the path), and a strict linear ordering $<_{Links}$ on links that is used for breaking ties. An order $<$ is *strict* if it is asymmetrical, i.e. $x < y$ implies $y \not< x$ for any x and y, and an order $<$ is *linear* if exactly one of $x < y$, $x = y$, $y < x$ holds for any x, y. We assert strictness and linearity of $<_{Links}$ and $<_{paths}$ with the following axioms:

Axiom 4. $<_{Links}$ is a strict linear order on *Links*.

Axiom 5. $<_{paths}$ is a strict order on $Seq(AS)$, i.e. AS paths, and, for any natural number, n, its restriction to paths of length n is a linear order.

Given these, we define a prefix-wise linear order, $<_{Route}$ on routes as:

Definition 2. For all s, p, l, s', p', l', $(s, p, l) <_{Route} (s', p', l')$ holds if and only if $s = s'$ and

$$\begin{aligned}
&(rank(s,p,l) > rank(s',p',l')) \vee \\
&(rank(s,p,l) = rank(s',p',l') \wedge \\
&\quad (length(p) < length(p')) \vee \\
&\quad (length(p) = length(p') \wedge p <_{paths} p') \vee \\
&\quad (p = p' \wedge l <_{Links} l') \\
&)
\end{aligned}$$

We claim, without proof, that $<_{Route}$ is a prefix-wise strict linear order:

Definition 3. For any prefix s, define $Route_s = \{(s', p', l') \in Route \mid s = s'\}$.

LEMMA 1. *For any prefix s (i.e. $s \in Prefixes$), the set $Route_s$ is strictly and linearly ordered by $<_{Route}$.*

We can then define a set *Best*, where $rte \in Best$ asserts that rte is the best known route among those with the same prefix. We define *Best* as:

Definition 4.

$$\begin{aligned}
Best = &\{(s,p,l) \in Knows \mid \\
&\forall p' l' \, [Knows(s,p',l') \rightarrow (s,p,l) \leq_{Route} (s,p',l')]\}
\end{aligned}$$

Given these definitions, we can prove properties that will be important in correctness proofs later:

LEMMA 2. $Best \subseteq Knows$

PROOF. Obvious. □

LEMMA 3. *If* $(s,p,l) \in Knows$, *then* $\exists p'l' \,.\, (s,p',l') \in Best$.

PROOF. Since $(s,p,l) \in Knows$, the set $\{(s',p',l') \in Knows \mid s = s'\} \neq \emptyset$ and is finite. Since the routes in the set all have the same prefix, the set is linearly ordered by $\leq_{Route}$. Since any nonempty, finite subset of a linearly ordered set has a least element, there exists a least element (s,p',l') in the set and this is in *Best*. □

LEMMA 4. *If* $(rank(s,p,l) > rank(s,p',l')$ *and* $(s,p,l) \in Knows$, *then* $(s,p',l') \notin Best$.

PROOF. Even if $(s,p',l') \in Knows$, since $(s,p,l) \in Knows$ and $(s,p,l) <_{Route} (s,p',l')$, $(s,p',l') \notin Best$. □

Routers forward traffic to an address to the most specific route, and so we define the set *MostSpecific*, as follows, with the help of a definition of *KnownPrefix*:

Definition 5.

$$KnownPrefix(a) = \{s \mid \exists p,l \,.\, (s,p,l) \in Knows \wedge a \in s\}$$

Definition 6.

$$MostSpecific = \{(a,s) \mid s \in KnownPrefix(a) \wedge \forall s' \,.\, s' \in KnownPrefix(a) \rightarrow s \subseteq s'\}$$

That is, $(a,s) \in MostSpecific$ if and only if the most specific known route to address a has prefix s.

Given these definitions we know have useful lemma, stating that if a route to an address is known, then there is a best and most specific one:

LEMMA 5. *If* $s \in KnownPrefix(a)$, *then* $\exists s',p',l' \,.\, (s',p',l') \in Best \wedge (a,s') \in MostSpecific \wedge s' \subseteq s$.

PROOF. By the assumption, the set $\{s' \mid \exists p',l'[(s',p',l') \in Knows \wedge a \in s'\}$ is nonempty. Therefore, by Axiom 1, there is a $\subseteq$-minimal prefix s' of $KnownPrefix(a)$ and hence $s' \subseteq s$. Therefore, $(a,s') \in MostSpecific$, by definition 6. Then by Lemma 3, we have $(s',p'',l'') \in Best$ for some p'',l'', and the conclusion is established. □

Finally, we define the predicate $Egress : Addresses \times Links$, which gives the egress link, i.e. the link over which packets will flow, for an address (if it exists):

Definition 7.

$$Egress(a,l) = \exists s,p \,.\, (a,s) \in MostSpecific \wedge (s,p,l) \in Best$$

The *Egress* predicate thus reflects the forwarding behavior of the network with regard to external addresses. The specification we develop later will be given in terms of assertions about the *Egress* predicate.

Figure 2 summarizes the language.

4. MULTI-HOMED LOAD BALANCING NETWORK

In this section we formally model the example presented informally in Section 2 and verify the second and third versions of the specification mentioned in that section. We begin by formalizing the topology as follows:

Sets:	*Prefixes*, *Neighbors*, *Links*, *Route*, *Announce*
Functions:	*linkTo*, *rank*
Predicates:	$<_{Links}$, $<_{paths}$
Axioms:	1-5
Defined terms:	*Knows*, *Best*, *MostSpecific*, *Egress*.

Figure 2: Summary of the language.

Definition 8. $Neighbors = \{100, 200\}$

Definition 9. $Links = \{oc3, ds3\}$

Definition 10.

$$linkTo(oc3) = 100$$
$$linkTo(ds3) = 200$$

If we momentarily take for granted the existence of a predicate *Cust*, where $Cust(a, nbr)$ asserts that address a is in a network that is a customer of nbr, then we can formalize the second and third specifications in Section 2 as $\gamma_2 \wedge \gamma_3$ and $\gamma_1 \wedge \gamma_2 \wedge \gamma_3$, respectively, where:

Definition 11.

$$\gamma_1 = \forall a \,[\, (Cust(a,100) \wedge Cust(a,200)) \rightarrow Egress(a, oc3)]$$
$$\gamma_2 = \forall a \,[\, (\neg Cust(a,100) \wedge Cust(a,200)) \rightarrow Egress(a, ds3)]$$
$$\gamma_3 = \forall a \,[\, \neg Cust(a,200) \rightarrow Egress(a, oc3)]$$

Note that γ_3 describes the case of traffic to destinations that are customers of neither 100 nor 200, since for such traffic $\neg Cust(a, 200)$ will hold.

We can describe the behavior of 100 and 200 as $PartialAndDefault(100) \wedge PartialAndDefault(200)$, where

Definition 12.

$$PartialAndDefault(asn) = PartialAnnounced(asn) \wedge DefaultAnnounced(asn)$$

Definition 13.

$$DefaultAnnounced(asn) = \exists p,l \,.\, (asn, (0.0.0.0/0, p, l)) \in Announce$$

Definition 14.

$$PartialAnnounced(asn) = [\forall a \,.\, Cust(a, asn) \leftrightarrow (\exists s,p,l \,.\, (asn, (s,p,l)) \in Announce \wedge a \in s \wedge s \subset 0.0.0.0/0)]$$

Note that this description of the behavior of AS 100 and AS 200 completely determines the *Cust* predicate in the case of this example without statically determining precisely which networks are customers.

The policy given in Zhang et al [3] is to rank routes learned over the OC-3 at a high preference level, $high = 120$, while routes learned over the DS-3 are given preference $low = 100$, and we can formalize this policy as:

Definition 15.

$$rank(s, p, oc3) = 120$$
$$rank(s, p, ds3) = 100$$

This has the effect of preferring the OC-3 link for the default route and routes to non-customers of AS 200 only, while preferring routes over the DS-3 link for networks that are customers of 200 and not customers of 100. In fact, we prove these assertions in the following lemmas. We start with a lemma that will be used later.

LEMMA 6.

$$PartialAndDefault(asn) \vdash \forall a.Cust(a, asn) \rightarrow \exists s, p, l[(s, p, l) \in Best \wedge (a, s) \in MostSpecific \wedge s \subset 0.0.0.0/0]$$

PROOF. Since $PartialAndDefault(asn)$ and $Cust(a, asn)$, we have $Announce(asn, (s, p, l)) \wedge a \in s \wedge s \subset 0.0.0.0/0)$ for some s, p, l. By definition of *Knows* we have $(s, p, l) \in Knows$ and therefore by Lemma 5, we have $(s', p', l') \in Best \wedge (a, s') \in MostSpecific \wedge s' \subseteq s$ for some s', p', l'. By transitivity of $\subset$, we have $(s', p', l') \in Best \wedge (a, s') \in MostSpecific \wedge s' \subset 0.0.0.0/0$. □

LEMMA 7.

$$PartialAndDefault(asn) \vdash Cust(a, asn) \rightarrow \exists l\,.\,(Egress(a, l) \wedge Cust(a, linkTo(l)))$$

PROOF. By Lemma 6 and $Cust(a, asn)$, we have $(s, p, l) \in Best \wedge (a, s) \in MostSpecific \wedge s \subset 0.0.0.0/0$, and hence $Egress(a, l)$.

Hence $(s, p, l) \in Best \subseteq Knows \wedge (a, s) \in MostSpecific$ and so $(asn', (s, p, l)) \in Announce$ for some *asn'*, $a \in s$, $s \subset 0.0.0.0/0$ and $linkTo(l) = asn'$. Thus $Cust(a, linkTo(l))$ by Definition 14. □

Now we demonstrate that γ_2 and γ_3 hold in the following two lemmas.

LEMMA 8. *Traffic for addresses that belong to customers of AS 200 only, use the ds3 link (i.e. γ_2 holds):*

$$\{PartialAndDefault(100), PartialAndDefault(200)\} \vdash \forall a[(Cust(a, 200) \wedge \neg Cust(a, 100)) \rightarrow Egress(a, ds3)]$$

PROOF. Assume $PartialAndDefault(100)$, $PartialAndDefault(200)$, $Cust(a, 200)$ and $\neg Cust(a, 100)$. By Lemma 7 we have $Egress(a, l)$ and $Cust(a, linkTo(l))$. Either $l = ds3$ or $l = oc3$.

If $l = oc3$, then $linkTo(l) = 100$ and $Cust(a, 100)$, a contradiction. Therefore $l = ds3$ and $Egress(a, ds3)$. □

LEMMA 9. *Use oc3 link for addresses that do not belong to customers of 200 (i.e. γ_3 holds):*

$$\{PartialAndDefault(100), PartialAndDefault(200)\} \vdash \forall a[\neg Cust(a, 200) \rightarrow Egress(a, oc3)]$$

PROOF. Assume $PartialAndDefault(100)$, $PartialAndDefault(200)$, $\neg Cust(a, 200)$, and let a be an arbitrary address.

Since $DefaultAnnounced(100)$, we have

$$(100, (0.0.0.0/0, p, oc3)) \in Announce$$

for some path p. By Lemma 5 we get $(s', p', l') \in Best$, $(a, s') \in MostSpecific$, and $s' \subseteq 0.0.0.0/0$ for some s', p', l', and hence $(asn, (s', p', l')) \in Announce$ for some *asn*.

Suppose $l' = ds3$. By Axiom 3, we get $asn = 200$. Either $s \subset 0.0.0.0/0$ or $s = 0.0.0.0/0$.

If $s \subset 0.0.0.0/0$, then $(200, (s', p', ds3)) \in Announce$, $a \in s'$, $s' \subset 0.0.0.0/0$ and hence $Cust(a, 200)$ by assumption of $PartialAnnounced(200)$, contradicting our assumption. Hence $s = 0.0.0.0/0$.

By definition of *rank*, we have

$$rank(s, p', oc3) > rank(s, p, ds3)$$

and in particular we have

$$rank(0.0.0.0/0, p', oc3) > rank(0.0.0.0/0, p, ds3)$$

By assumption of $PartialAnnounced(100)$, we have

$$(100, (0.0.0.0/0, p', oc3)) \in Announce$$

and hence $(0.0.0.0/0, p', oc3) \in Knows$. Then by Lemma 4, we have $(s, p, ds3) \notin Best$, a contradiction. Hence $l \neq ds3$.

Therefore $l = oc3$ and $Egress(a, oc3)$. □

At this point, we have verified the second specification, namely $\gamma_2 \wedge \gamma_3$. As mentioned in Section 2, we cannot verify the third specification, $\gamma_1 \wedge \gamma_2 \wedge \gamma_3$ because there is in fact a counterexample, which we present more precisely below.

4.1 Counterexample

The third specification, $\gamma_1 \wedge \gamma_2 \wedge \gamma_3$, fails, because without further assumptions, γ_1 fails to hold in every situation. For example, consider the following set of announcements:

$$\begin{aligned} Announce = \{&(100, (0.0.0.0/0, [100], oc3)),\\ &(100, (1.0.0.0/8, [100, 300], oc3)),\\ &(200, (0.0.0.0/0, [200], ds3)),\\ &(200, (1.0.0.0/16, [200, 300], ds3))\} \end{aligned}$$

It is easy to verify that Axioms 1-5 are satisfied, and that $PartialAndDefault(100)$ and $PartialAndDefault(200)$ are satisfied, since both 100 and 200 have announced default routes.

Now, we can compute *Knows* and *Best* given this set of announcements, where we are assuming *rank* is defined as in Definition 15 when computing *Best*:

$$\begin{aligned} Knows = \{&(0.0.0.0/0, [100], oc3),\\ &(1.0.0.0/8, [100, 300], oc3),\\ &(0.0.0.0/0, [200], ds3),\\ &(1.0.0.0/16, [200, 300], ds3)\} \end{aligned}$$

$$\begin{aligned} Best = \{&(0.0.0.0/0, [100], oc3),\\ &(1.0.0.0/8, [100, 300], oc3),\\ &(1.0.0.0/16, [200, 300], ds3)\} \end{aligned}$$

Now let address $a = 1.0.0.0$. Since $(a, 1.0.0.0/16) \in MostSpecific$, we get $Egress(a, ds3)$. On the other hand, by definition of *Cust*, we have $Cust(a, 100)$ and $Cust(a, 200)$ and so γ_1 requires that $Egress(a, oc3)$. Therefore, we fail to satisfy γ_1 under this set of announcements.

4.2 Adding an Assumption

We therefore need to formulate an extra assumption in order to verify the third specification, $\gamma_1 \wedge \gamma_2 \wedge \gamma_3$. We do this using the following definition, which asserts that any routes with non-default and overlapping prefixes announced by both networks are in fact equal.

Definition 16.

$$\begin{aligned}&\mathit{CustomersAnnouncedIdentically} = \forall a, s, p, l, s', p', l' \\ &[((100, (s, p, l)) \in \mathit{Announce} \wedge (200, (s', p', l')) \in \mathit{Announce} \\ &\wedge\ a \in s \wedge a \in s' \\ &\wedge\ s \neq 0.0.0.0/0 \wedge s' \neq 0.0.0.0/0) \\ &\rightarrow s = s']\end{aligned}$$

We now add this assumption and prove the remaining part of the specification:

LEMMA 10.

$$\begin{aligned}\{&\mathit{PartialAndDefault}(100), \\ &\mathit{PartialAndDefault}(200), \\ &\mathit{CustomersAnnouncedIdentically}\} \\ &\vdash \mathit{Cust}(a, 100) \rightarrow \mathit{Egress}(a, \mathit{oc3})\end{aligned}$$

PROOF. Assume $\mathit{Cust}(a, 100)$. Then we have, for some s, p, l, $(100, (s, p, l)) \in \mathit{Announce}$, $a \in s$, and $s \subset 0.0.0.0/0$. By Lemma 7 we have $\mathit{Egress}(a, l')$ and $\mathit{Cust}(a, \mathit{linkTo}(l'))$. Either $l' = \mathit{ds3}$ or $l' = \mathit{oc3}$.

Suppose $l' = \mathit{ds3}$. Then $\mathit{Cust}(a, 200)$ and so, for some s', p', $(200, (s', p', l')) \in \mathit{Announce}$. We therefore have the premise of the *CustomersAnnouncedIdentically* assumption, and hence we have $s = s'$. Since, (by definition of *rank*), $\mathit{rank}(s, p', \mathit{ds3}) < \mathit{rank}(s, p, \mathit{oc3})$ and $(s, p, \mathit{oc3}) \in \mathit{Knows}$, we have $(s, p', \mathit{ds3}) \notin \mathit{Best}$ (by Lemma 4), contradicting $(s', p', l') = (s, p', \mathit{ds3}) \in \mathit{Best}$. Hence $l' \neq \mathit{ds3}$.

Therefore $l' = \mathit{oc3}$ and $\mathit{Egress}(a, \mathit{oc3})$. □

Given these three lemmas, we can now verify our specification as originally given:

THEOREM 1.

$$\begin{aligned}\{&\mathit{PartialAndDefault}(100), \\ &\mathit{PartialAndDefault}(200), \\ &\mathit{CustomersAnnouncedIdentically}\} \\ &\vdash \sigma_1 \wedge \sigma_2\end{aligned}$$

PROOF. σ_1 holds by Lemma 8. σ_2 holds since, if $\mathit{Cust}(a, 100)$, then $\mathit{Egress}(a, \mathit{oc3})$ by Lemma 10, while if $\neg\mathit{Cust}(a, 200)$, then $\mathit{Egress}(a, \mathit{oc3})$ by Lemma 9. □

5. FORMALIZATION

We have formalized the above language and proofs in the Isabelle/HOL theorem prover [2], and our source is located on our web page [3]. The language and proofs in the Isabelle/HOL formalization are essentially the same as those presented here, but are adapted to Isabelle's typed higher-order logic. The formalization in Isabelle/HOL gives us high confidence that we have expressed the relevant features and axioms of the language and that we have correctly verified the specification of the case study.

[2]http://isabelle.in.tum.de

[3]http://www.haskell.org/YaleHaskellGroupWiki/Networks

6. CONCLUSION AND FUTURE WORK

The previous sections illustrate a case study in which the desired behavior of the network with regard to interdomain traffic is more naturally expressed in terms of forwarding behavior than in terms of route preferences. We describe interdomain forwarding behavior with assertions regarding the *Egress* predicate, describe our policy with assertions regarding the *rank* function, and prove that the implementation achieved the specification, subject to certain assumptions, using only logical arguments supplemented with our definitions and axioms.

This case study illustrates where the difficulty in configuration lies, namely in the gap between specification and implementation. In this case, both the network specification and the policy implementation are simple, yet the justification for the correctness of the policy is not at all trivial. The justification is not deep or complex, but it does require detailed and careful considerations which may be overlooked by operators, especially given the apparently trivial specification and implementation.

The complications in the correctness arguments arise primarily due to the nontrivial behavior of forwarding and the fact that routing policy is applied to each prefix independently. Traffic is forwarded along the most specific route and this requires showing that the most specific *and* best route uses a particular link. On the other hand, the prefix-wise application of policy makes the specification impossible to achieve without the *CustomersAnnouncedIdentically* assumption (or something like it), which may be an assumption of questionable validity. If interdomain policy included the ability to state something to the effect of "only use a route for non-default prefix p from 200 if there is no known non-default route containing p", then we could have implemented the policy without any unreasonable assumptions on neighbor behavior.

We take the fact that certain ambiguities in the specification and certain possibilities of network behavior were unmentioned in the original case study to be evidence that understanding network behavior is non-trivial, and that providing an explicit network model and reasoning principles will help in safely configuring networks with regard to interdomain traffic.

An interesting direction for future work would be in combining the verification techniques with policy languages supporting the expression of policy functions as a means for expressing policy templates, such as Nettle [2], i.e. policy fragments which when applied to arguments yield policies. Just as introducing functions or procedures in a programming language increases the need for specifying the abstract behavior of such components, the introduction of policy functions increases the need to specify the behavior of networks under the application of such policies. In particular, it would be interesting to express the abstract principles used in the load balancing case study presented here, to express the pattern as a policy function, and to describe the resulting function with assertions in a specification language similar to the one we have provided here. Such assertions could be used either in verification, or perhaps more likely, they could be checked on random or systematically generated sets of announcements satisfying the above axioms. While such testing would of course not prove the absence of errors, it would provide a low-effort technique for increasing confidence in the correctness of policies.

7. ACKNOWLEDGEMENTS

Thanks to Paul Hudak, Richard Yang, and Vijay Ramachandran for encouraging this work and to Antonis Stampoulis for helpful advice on formalizing the work in Isabelle/HOL. This research was supported in part by NSF grant CCF-0728443 and DARPA grant STTR ST061-002 (a subcontract from Galois, Inc.).

8. REFERENCES

[1] R. Mahajan, D. Wetherall, and T. Anderson. Understanding bgp misconfiguration. *SIGCOMM Comput. Commun. Rev.*, 32(4):3–16, 2002.

[2] A. Voellmy and P. Hudak. Nettle: a language for configuring routing networks. In *Domain-Specific Languages, IFIP TC 2 Working Conference*, volume 5658, pages 211–235. Springer Berlin / Heidelberg, July 2009.

[3] R. Zhang and M. Bartell. *BGP Design and Implementation*. Cisco Press, 2003.