Atom-Based Routing

Patrick Verkaik, kc claffy, Andre Broido, Evi Nemeth

June 22, 2002

1 Introduction

Global routing in today's Internet is negotiated among individually operated sets of networks known as Autonomous Systems (AS). An AS is an entity that connects one or more networks to the Internet, and applies its own policies to the exchange of traffic.

AS policy is used to control routing of traffic from and to certain networks via specific connections. These policies are articulated in router configuration languages and implemented by the Border Gateway Protocol (BGP) [8].

A basic BGP exchange consists of a message announcing (advertising) reachability of a single network via a certain router. The reachability information includes an AS path, which is a sequence of ASes. BGP assumes that:

- This path is taken by the reachability message.
- The advertised network can be reached via this path.

A BGP table associates a network prefix identifier (prefix) with the AS path through which the network is considered reachable. This table is an important ingredient of the packet forwarding process by a BGP router.

Reduction of the number of entries in BGP tables is typically seen as beneficial to infrastructural integrity. The number of entries in a table has bearing on both router memory and CPU cycles. The number and size of routing update messages (announcements and withdrawals of networks) tends to increase with the number of prefixes in the table. Not only are communication costs affected by this, but also the CPU resources needed to process the updates [4].

The project proposed in this document aims to significantly reduce growth of BGP table size and updates, in particular in the Internet backbone[1], through the use of BGP policy atoms [1]. The intent is to devise a routing protocol[2] (or

[1]The problem of BGP table size is not as severe outside the backbone. Routers that are not part of the backbone can rely on 'default routes' which direct traffic for unrecognised destinations towards the backbone. Therefore they do not need to have a complete picture of the Internet. Routers in the backbone on the other hand cannot make use of such default routes, and need to have a picture of the Internet which covers every globally reachable IP address.

[2]We consider the routing protocol to be both the messages exchanged by the protocol and the basic algorithms of the routers that exchange these messages.

adapt a routing protocol such as BGP) which makes use of atoms to achieve a protocol of lower compexity.

Atoms could reduce current backbone BGP table sizes and growth by a factor of two, using the methodology of [2]. However as mentioned later, there is a potential of a far greater return than a factor of two: around 22%.

2 Background — CIDR

CIDR (Classless Inter-Domain Routing) has considerably reduced routing table size growth through (a) aggregation of IP addresses into IP address prefixes, and (b) an address allocation policy that creates opportunities for aggregation [7]. In this section the concepts of prefixes and aggregation are explained. Then we will examine the benefits CIDR has to offer, as well as its limitations.

2.1 Prefixes

An *(IP address) prefix* represents a range of IP addresses, and consists of an IP address part *addr* and a prefix length part *p*. A prefix is usually written as *addr/p*. The *p* indicates the leftmost contiguous significant bits within *addr* [7]. The IP address range denoted by a prefix *addr/p* are those IP addresses whose leftmost *p* bits equal the corresponding bits in *addr*. For example, the prefix 192.24.0.0/13 (IP address 192.24.0.0 and prefix length 13 bits) corresponds to the IP address range from 192.24.0.0 to 192.31.255.255.

When two prefixes have the same IP address part, but differ in their prefix length part, the prefix with the longer prefix length is said to be *more specific*, and the prefix with the shorter prefix length is said to be *less specific*. For example, prefix 192.24.0.0/21 is more specific than prefix 192.24.0.0/13. The IP address range of a prefix is a subset of the IP address range of any less specific prefix.

2.2 Aggregation and CIDR Benefits

CIDR allows a router to aggregate (summarise) a number of IP addresses and IP prefixes into a single IP prefix, and to announce to other routers only the resulting less specific prefix (aggregated prefix) instead of the more specific IP addresses and prefixes that it covers. To significantly benefit from the introduction of aggregation, CIDR also specifies an address allocation policy which creates the conditions in which aggregation can be performed.

An example of aggregation is shown in Figure 1. A provider AS (AS 2) has been allocated the IP address block 192.24.0.0/13. The provider has two customer ASes (AS 1 and AS 3). In accordance with CIDR address allocation policy, the IP address blocks assigned to these customer ASes have been allocated out of the provider's address space: AS 1 is allocated 192.24.0.0/21 and AS 3 is allocated 192.24.8.0/21. AS 1 and 3 announce prefixes for these address blocks to AS 2. The announcements are indicated by the arrows. AS 2 is attached to the backbone, and must make reachability announcements covering the address blocks of all three ASes. Due to the allocation of AS 1 and AS 3's address blocks out AS 2's address space, AS 2 is now able to aggregate the prefixes of AS 1 and AS 3 into its own prefix 192.24.0.0/13, and therefore only needs to announce this single prefix into the backbone.

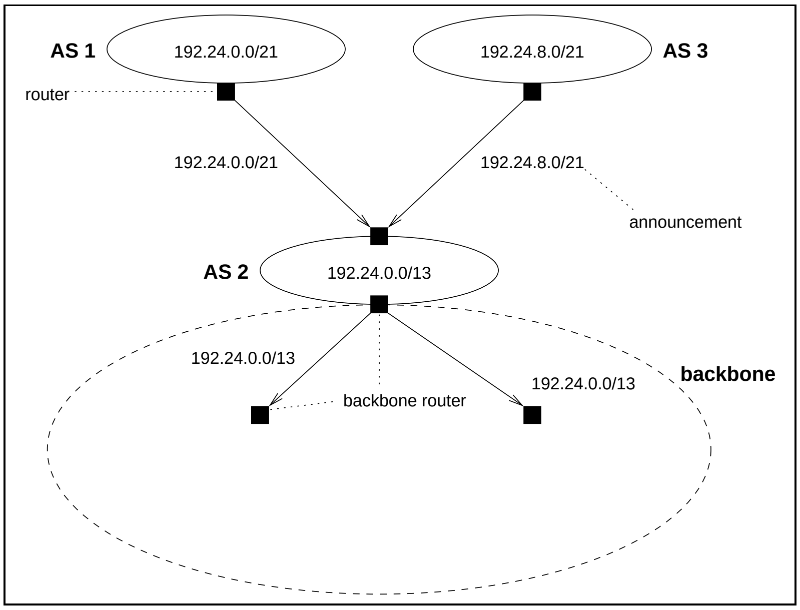

Figure 1: Aggregation of prefixes.

CIDR offers the following benefits through aggregation:

1. The aggregating router is able to announce an aggregated prefix instead of the more specific IP addresses and prefixes that the aggregated prefix covers. This reduces the table size of any routers that learn of this announcement.

2. The number and size of update messages (announcing or withdrawing IP addresses) are reduced. Not only do update messages carry just the aggregated prefixes instead of the more specific information, but CIDR can also prevent instability at the edge of the network to immediately propagate to the backbone [4]. The instability is 'absorbed' at the point where the affected address space is aggregated.

2.3 More Specifics and the Limitations of CIDR

CIDR allows a more specific prefix of some other prefix to be advertised. More specific prefixes override the routing policies associated with its less specific prefixes, as follows. A BGP router that receives advertisements of a more and a less specific prefix, will forward traffic along the AS path of the more specific prefix.

A BGP router that has received advertisements of a more and a less specific prefix may aggregate the more specific prefix into the less specific prefix in its advertisements to other routers. However, there is an important reason for choosing not to do so: by advertising only the aggregate, the overriding quality that the more specific prefix has (by virtue of its longer prefix length) is not passed on to other routers, resulting in a loss of policy information. (An example of this appears below.) As a result, BGP routers will often not aggregate more and less specific prefixes, and instead advertise both prefixes to other routers.

More specific prefixes are often used for traffic engineering purposes. An example of this is shown in Figure 2. AS 1 has two provider ASes, AS 2 and 3, both of which are attached to the backbone. Having several providers, AS 1 is said to be *multihomed*. One reason for multihoming an AS is to improve the connectivity of the AS. AS 2 and 3 have been allocated the address blocks 192.24.0.0/13 and 192.32.0.0/13, respectively, which they announce into the backbone. AS 1 has been allocated an IP address block 192.24.0.0/21 out of the address space of AS 2. In this scenario, AS 1 wishes to balance the load of its incoming traffic over the two links. To do so, it advertises half of its address space (192.24.0.0/22) to AS 2 and the other half (192.24.4.0/22) to AS 3. To ensure that the whole of AS 1's address remains reachable, should either of its provider links go down, AS 1 additionally advertises its entire address block (192.24.0.0/21) to both providers. Due to the fact that the two more specific advertisements take precedence, load balancing will still be achieved.

The prefixes advertised to AS 3 cannot be aggregated into AS 3's own prefix advertisement, and must therefore be advertised separately into the backbone. The prefixes advertised to AS 2 could be aggregated into AS 2's own prefix advertisement. However, that would cause all traffic destined for AS 1 to be attracted towards the more specific prefixes advertised by AS 3, defeating the load balancing objective. Therefore AS 1 convinces AS 2 to announce both the whole and the half block into the backbone.

Figure 2 illustrates two important limitations of CIDR, both of which adversely affect the size of BGP router tables:

- Networks (AS 1) that are not allocated out of the address space of their provider (AS 3) cannot be aggregated into their provider's prefixes. The prefixes of such networks may therefore end up as separate entries in BGP router tables.

- By allowing more specific prefixes of existing prefixes to be advertised, additional prefixes are introduced that may end up in BGP router tables.

Other reasons behind the occurrence of these phenomena are discussed in [1], [3] and [4].

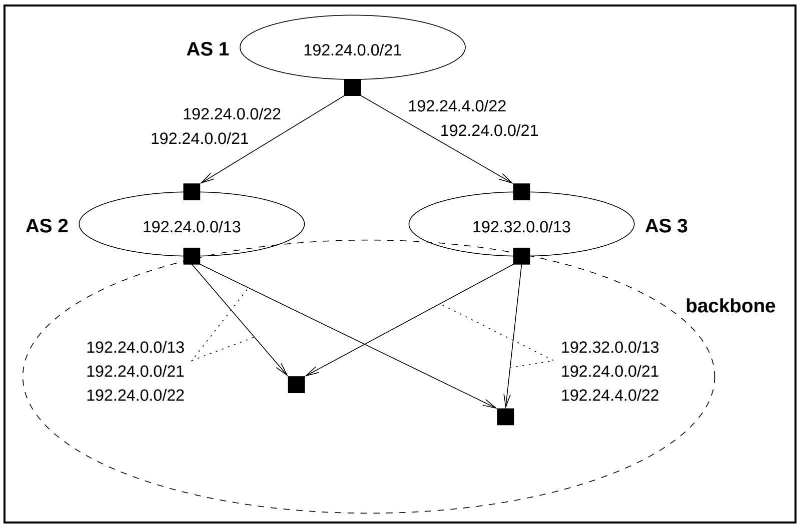

Figure 2: A multihomed AS.

3 Policy Atoms

In [1] the notion of 'policy atoms' was introduced as a means to analyse the complexity of routing tables. By analysing a number of backbone routers the authors found that the use of policy atoms could potentially reduce the size of a complete backbone BGP table by a factor of two, while preservering all globally visible routing policies [2]. In addition, they found that the number of atoms properly scales with the Internet's growth.

3.1 Definition of Policy Atoms

An atom is defined relative to a system of BGP routers. For example, it may be defined relative to the Internet backbone routers or (in the extreme) to all BGP routers in the Internet. Each atom consists of prefixes that are treated equivalently by the chosen system of BGP routers. For example in Figure 2 the prefixes 192.24.0.0/13 and 192.24.0.0/22 are treated equivalently by the backbone routers and are therefore part of the same atom (relative to the backbone).

In [1], an atom is defined as follows. Two prefixes are said to be *path equivalent* if no BGP router can be found among the considered BGP routers that sees them with different AS paths. An equivalence class of this relation is called a *(BGP policy) atom*. It follows from this definition that prefixes in the same atom share a set of AS paths.

In this proposal we use a slightly modify definition, as follows[3]. In determin-

[3] This definition corresponds more closely to the definition of *crown atoms* in [1] than to regular (non-crown) atoms.

ing path equivalence of two prefixes, we ignore the part of a prefix's AS path that falls outside the system of BGP routers. From this definition, it follows that prefixes in the same atom share a set of truncated AS paths, where each AS path is truncated to exclude the part that falls outside the system. Note that under this modified definition a smaller number of atoms will result.

We call the system of BGP routers relative to which an atom is defined the *scope* of the atom.

Algorithmically, atoms may be constructed as follows:

1. Select a system of BGP routers to be considered (as mentioned above). This becomes the scope of the atoms.

2. Select the set of prefixes to be considered. For example, the prefixes common to all BGP routers in the chosen system might be selected [1]. Another possible selection is the set of prefixes each of which is known by at least one BGP router in the chosen system.

3. With each prefix associate a set of AS paths between the BGP routers. Truncate each AS path in the set by removing the part that falls outside the chosen system.

4. For each set of truncated AS paths, find all prefixes that share this set of paths.

The following example derives atoms from the prefixes shown in Figure 2 using the algorithm. The system of BGP routers considered (and therefore the scope of these atoms) consists of the backbone routers that appear in Figure 2; the prefixes considered are all those shown in Figure 2. The atoms that can then be derived are shown in Figure 3. Note that although the full AS paths of e.g. 192.24.0.0/13 and 192.24.0.0/22 are different (since the former is originated by AS 2 and the latter by AS 1), the parts of the AS paths which fall within the backbone, i.e. the truncated AS paths, are the same. Therefore they are in the same atom.

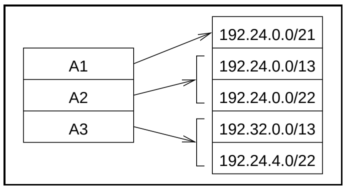

Figure 3: Atoms derived from Figure 2.

Note that the size of the atoms which result from the above definition depends in part on the system of routers considered: the fewer routers are considered the larger the atom size will tend to be. In addition the atom size also depends on the selection of prefixes considered.

4 Atom-Based Routing

Routing protocols such as BGP operate on individual prefixes. Each update, table entry, and computation is based on a single prefix as the basic element. Although several prefixes may be stored or transmitted at a time by BGP, the prefix remains the basic element of the protocol. For example, an update message may carry a route containing several prefixes, but the receiving BGP router will still need to consider each prefix in the message separately in its computations.

A routing protocol based on atoms will treat a number of prefixes as equivalent and amortise overhead of the protocol over the equivalent prefixes. Such a routing protocol is the goal of this project.

The effects of atom-based routing are similar to CIDR in that both are able to summarise prefixes (as aggregates and atoms respectively) and treat the summary as a unit. An important difference is that CIDR aggregation can be performed independently by each router; however by definition the computation of an atom requires cooperation between many routers.

Figure 4 illustrates the general idea behind atom-based routing as applied to the example in Figure 2. The atoms shown are those derived in Figure 3. In Figure 4, advertisements of atoms replace advertisements of individual prefixes. As a result, the number of update messages in the backbone has decreased from six to four, and the number of entities advertised into the backbone has decreased from five to three. This is one example of how an atom-based routing protocol might take advantage of the equivalence of the prefixes in an atom. Note that this example assumes that the routers involved know what prefixes make up each atom, i.e. have some of means of knowing the mapping in Figure 3.

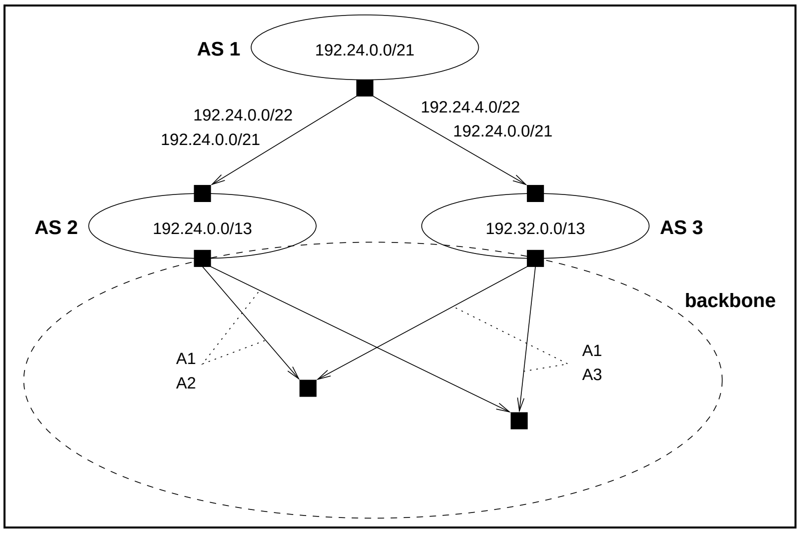

Figure 4: Example of atoms.

4.1 Dimensions of Atom-Based Routing Protocols

A number of atom-based routing protocols are possible, varying in a number of dimensions, some of which are:

Scope of atoms Computing atoms in a large system of BGP routers may run into scalability problems. Possible scalability problems are:

- Determining the precise set of equivalent prefixes that make up an atom may require taking a global snapshot of all BGP routers in the system. For systems consisting of many routers, this may be too expensive.
- If the messages of the atom-based protocol contain references to atoms, the routers involved in a message exchange need to agree on what atom is being referred to in the message. In other words, the routers need to have a common naming (identification) system of atoms. Establishing a common naming system among all the routers in a large system may be too expensive.

These problems may be diminished by limiting the scope of an atom. For example, an atom-based protocol might divide a large system of BGP routers into smaller 'areas', and compute atoms independently in each area. The scope of an atom then consists of the area it is computed in.

The more areas the system is divided into, the smaller each area will be, and the cheaper it will be to manage the atoms of that area. However, there is a trade-off, in that the more areas the system is divided into, the less useful each atom may be to the system as a whole.

Knowledge of prefixes An atom-based routing protocol may be able to do away with the knowledge of the individual prefixes that compose an atom (i.e. a mapping such as the one shown in Figure 3), in a subset of the routers. An example appears in Section 5.1.

Alternatively, even if knowledge of individual prefixes continues to be maintained by each router, an atom-based routing protocol allows large tables to be taken out of the critical path during packet forwarding. This capability is examined in Section 5.2.

4.2 Benefits of Atom-Based Routing

Benefits in one or more of the following areas are to be expected:

Table size If each entry in a router's table governs an atom rather than a single prefix, fewer entries need to be stored. Note that a straightforward compression algorithm in a BGP router may be able to reduce table size as well, but at the expense of CPU cycles. A more real advantage is obtained if (a subset of) routers do not need to be aware of each prefix in the system, or at least if individual prefixes can be eliminated from their packet forwarding tables and algorithms.

A factor of two reduction of backbone BGP table sizes was already established in [2]. However, there is a potential of a far greater return than a factor of two in a subset of routers. In replacing prefix entries by atom entries, a router's table size could be shrunk to 22.2% [2][4].

Note that even a factor of two can be substantial, considering that we should expect this factor to hold not only for today but for the ongoing growth of the Internet. In other words, this result halves the growth rate of backbone routing tables for all time.

While in terms of theoretical complexity this is a trivial result, from an engineering perspective, this can easily be the difference between a thriving Internet and considerable indigestion in the routing plane. Growth in the backbone tables is somewhere around (at least) half the rate dictated by Moore's law.

Update communication costs If each update message in the routing protocol governs an atom rather than one prefix, fewer update messages are needed. Note that BGP already allows an update message to govern more than one prefix simply by including a list of prefixes to which the message applies. Update messages in an atom-based routing protocol on the other hand may govern multiple prefixes without listing those prefixes explicitly (thereby reducing their size).

Another potential source of savings in communication costs is the effect of absorption of routing instability, which is similar to the absorption of routing instability resulting from CIDR aggregation (Section 2.2). An example of this appears in Section 5.1.

Note that there is a trade-off with the communication costs needed to compute and maintain an agreed set of atoms between routers.

Update computation costs The algorithms of the protocol allow the routing information of the prefixes of an atom to be updated at one time. Whereas BGP may carry several prefixes in one update message, prefixes still need to be considered individually by the BGP algorithms.

Furthermore, the effect of absorption of routing instability mentioned above would also reduce update computation costs.

Note that again there is a trade-off with the processing needed to compute and maintain an agreed set of atoms between routers.

[4]This figure is based on the definition of atoms given in [2]. Using the modified definition of Section 3.1, the number of entries should drop to the number of *crown atoms*. Recent figures (June 1 2002) indicate a factor of four for the original definition of atoms in [2], or five for our modified definition.

5 Practical Deployment of Atom-Based Routing

This section shows two ways of incrementally deploying atom-based routing in the current Internet. The approaches discussed share the following properties:

- They affect only cooperating routers in the backbone. They are transparent to other backbone routers, as well as to routers outside the backbone.
- The protocols used are based on existing routing and forwarding protocols, with only minor modifications.

Neither approach addresses atom computation. Instead they simply assume that the problem of scalable atom computation has been solved. In Section 5.3 we briefly discuss a number of issues pertaining to atom computation.

5.1 Atom-Based Routing Islands

Here we show how to embed an 'island' of atom-based routing in the backbone. Routers that are part of such an island reap the benefits of atom-based routing. On the other hand, outside the island the application of atom-based routing is completely transparent (apart from observed performance). By introducing an island of atom-based routing, and gradually growing it, it is possible to effect a gradual transition to an 'atomised' Internet backbone. The aproach described here extends BGP, and uses IP addresses (IPv4 or IPv6 addresses) to represent atom ids. The aproach has the following properties (see Section 4.1):

Scope of atoms The scope of each atom is the entire island of atom-based routing. Note that a more sophisticated version might divide an island into more than one scope for scalability.

Knowledge of prefixes This aproach does not require each router to be aware of all prefixes. Only a subset of the routers are; others are only aware of atoms without knowing what prefixes they consist of.

The aproach distinguishes the following routers:

- External routers, which are outside the island. These routers are not aware of the atom-based routing protocol.
- Internal routers, which are within the island. Internal routers are further divided into edge routers and transit routers.
- Edge routers which are internal routers that exchange (forward) packets with external and internal routers.
- Transit routers which are internal routers that merely exchange (forward) packets with other internal routers.

The internal routers implement routing and forwarding in three parts:

1. Atom computation: As mentioned earlier, the problem of atom computation is assumed to have been solved. We assume the outcome of the computation is a collectively agreed upon atom id for each atom, and a mapping between atom identifiers and the prefixes atoms consist of. Only edge routers need to be aware of this mapping; transit routers do not.

2. Updates to the routing of atoms: Routing update messages are exchanged between all internal routers (both edge and transit routers). This part of the approach uses BGP. But whereas normally BGP carries reachability information for individual prefixes, here we use BGP to announce reachability information for atom ids. Since an atom id is just an IP address, regular BGP can be used. Alternatively, the MP_REACH_NLRI attribute of BGP [10] can be used to carry this information if we view atom ids as an address family.

3. Forwarding of packets: A packet is accepted by an edge router from an external router. Based on the destination IP address of the packet, the edge router tags the packet with the correct atom id. Based on the atom id, the packet is then forwarded by the system of internal routers (transit and edge routers) until it reaches the edge router where it leaves the island. This edge router removes the atom id tag and forwards the packet to an external router. One way of implementing this is to 'source route' the packet:

   - At the edge router where the packet enters the island, an IPv4 loose source route option [6] is added to the packet. If the packet already contains a source route option, the source route option is modified instead. The packet is source-routed first to the atom id destination, and then to the original destination of the packet. For IPv6, a routing header [11] can be inserted to achieve source routing.
   - The packet is forwarded until it leaves the island. Given that the atom id is an IP address, existing forwarding implementations can be used.
   - At the edge router where the packet leaves the island, the changes that were made to the packet when it entered the island (IPv4 source route option or IPv6 routing header) are undone.

Another way of implementing forwarding is to use tunneling, e.g. IP over IP, MPLS or GRE.

In addition, edge routers are responsible for passing on topology changes from external routers (in the form of prefixes) to internal routers (in the form of updates to atom definition and updates to atom routing) and vice-versa.

Referring back to the advantages in Section 4.2:

Table size Transit routers are able to maintain tables with far fewer entries: one per atom instead of one per prefix. Only the edge routers need to maintain per-prefix tables. In particular during packet forwarding, smaller tables are accessed in the internal routers, i.e. in all but one edge router (where the packet enters the island).

Update costs Updates that merely change the contents of an atom (i.e. add or remove prefixes) do not affect transit routers at all, since the routing of the atoms remains the same. They are effectively absorbed (see Section 4.2) by the edge routers. Similarly, updates that affect the routing of an entire atom can be summarised in an update message for the atom, rather than an update message per prefix. Routers that process such update messages need to spend fewer CPU cycles on updating their tables.

5.2 Disjunct Atom-Based Routers

Note: the ideas in this section require further elaboration.

The previous section discussed islands of connected atom-based routers. Here we show how to embed disjunct (disconnected) atom-based routers in the backbone.

The atom-based routers in this section are optimised for forwarding packets that carry an atom id tag, but also handle regular packets. The presence of these routers is transparent to regular, 'prefix-based' routers. This means that atom-based routers do not need to be placed together in islands, as in Section 5.1. Instead, atom-based routers may be located in different parts of the backbone, separated by prefix-based routers, and still be able to take advantage of atom-based routing. By gradually increasing the number of atom-based routers, it is possible to effect a gradual transition towards a more 'atomised' Internet backbone.

Atom-based routers forward packets as follows. If an atom-based router encounters a packet that has not yet been tagged, it tags the packet with the correct atom id and forwards it. If an atom-based router encounters a packet that has already been tagged, it simply forwards it based on the atom id, using a small forwarding table.

Atom tagging occurs in such a way that (a) other atom-based routers can detect whether or not a packet has been tagged, and (b) unmodified prefix-based routers are able to forward tagged packets. There are two ways of achieving this:

1. Atom tagging is performed in a similar way to Section 5.1, e.g. using a loose source option. However, instead of using regular IP addresses to represent atom ids, an atom id is represented by an IP address taken from a special range. This allows atom-based routers to readily detect that a packet has been tagged. In the case of IPv4, atom ids might initially be taken from the class E ('experimental') range. If the approach is successful a separate range might be allocated at some point (by IANA).

   To allow prefix-based routers to forward tagged packets, all we need to do is to ensure that additional entries for IP addresses representing atom ids are made (e.g. by announcing them using regular BGP messages). This of course has the disadvantage of increasing the size of the prefix-based router tables.

2. Atom tagging is performed in some other (unspecified) way that does not alter the destination of the packet. Since the destination of the packet is unaltered, prefix-based routers do not need to have separate entries corresponding to atom ids.

We are also able to handle prefixes that are not (yet) part of any atom. If an atom-based router encounters a packet that has not been tagged, but does not know of an atom that contains the prefix either, it passes it on to the nearest prefix-based router that it knows[5], which forwards it using a regular, large forwarding table. Therefore, even if some ratio of Internet traffic never becomes atomised, we will still be able to handle atomised traffic in an expedited way, and fall back to a less efficient forwarding behaviour for the remaining traffic.

5.3 Atom Computation

Neither of the above approaches addresses atom computation. Atom computation consists of identifying atoms and determining what prefixes belong to each atom. The outcome of atom computation is a collectively agreed upon atom id for each atom, and a mapping between atom ids and the prefixes atoms consist of.

We know how to compute atoms in a centralised way. However, to perfom atom computation in a scalable, distributed way is an unsolved problem, which will be addressed in the course of the project. Several scalability issues were discussed in Section 4.1. Special attention will be paid to convergence time. An example of a convergence issue is how to update (a) atom contents (i.e. the prefixes atoms consist of) and (b) atom routing (i.e. the routing of atoms as a whole) independently. We must avoid updates to one causing unnecessary updates to the other while the system converges to a new routing state. Note that convergence time is highly recognised as problematic at this point, and is an example of why this kind of research is so important.

6 Ansers to Questions

6.1 What about IP Version 6?

At the moment, applicability of atom-based routing to IPv6 is an open-ended question. The routing architecture for IPv6 is not well-defined. If, as seems likely, momentum or inertia is the rule, then the IPv6 routing architecture will be fundamentally similar to the IPv4 architecture (including CIDR and its limitations) and the results of this project will transpose cleanly.

IPv6 does introduce a more structured allocation of addresses, apparently intended to allow better aggregation. However under CIDR, multihomed ASes

[5] Possibly located at the same site.

will still introduce prefixes that their providers cannot aggregate (Section 2.3). Also, current engineering tactics (such as load balancing across prefixes) will still be prevalent in an IPv6 world. In addition, the increased length of IPv6 addresses will have a negative effect on router table size, and on the complexity of (distributed) computations.

However, the combination of atoms and IPv6 produces the interesting idea of placing an atom id in IPv6 addresses (at the time of allocation of the address). Routers would not need to maintain an explicit mapping between atoms and address prefixes covering such addresses. The number of entries in the tables covering such addresses would be equal to the number of atoms rather than the number of prefixes (which is substantially less).

Although the address format of an IPv6 address has pretty much been defined, if an implementation of atoms were done and proved successful, there would be motivation for the IETF to look again at the address bits and the way that they are allocated, and perhaps carve out some for an atom id.

6.2 Who Benefits from Atom-Based Routing?

Keeping the Internet backbone routing tables small is beneficial to all users of the Internet. It will help reduce the cost of the infrastructure, simplify the hardware technology that must be deployed and generally lengthen the life of the Internet address space.

6.2.1 Current Growth

The one thing everyone agrees on is that growth continues. Had it been left unchecked (e.g. by CIDR, controls by registry allocation and provider annoucement policies), it would have already quickly outstripped Moore's law. Even now, with checks in place, growth is uncomfortably fast, since providers have to continually upgrade memory sizes and processors in Internet backbone routers. 'Coping' in this way is not a good strategy and bears considerable long term risk.

6.2.2 Future Growth

The most recent backbone routers are within a factor of four of the most current processor clock rates and will continue to close quickly. However, the microprocessor industry is less motivated to develop faster processors as the desktop PC demand curve has flattened. If microcomputer industry continues to slow, router industry may actually have to drive the computing industry or have to use multiprocessor computers just to process the control plane. Neither is an appealing prospect, either from a technology point of view or from a robustness perspective. Furthermore, it seems unlikely that the router industry will be able to drive the computing industry: not even the entire router market is large enough to sway the microprocessor industry to accelerate R&D sufficiently.

6.3 Aren't Routing Vendors Tackling these Problems?

The ability to advertise multiple prefixes with common attributes as part of a single BGP update message is a known capability and is not overly difficult. However, the further ability to take that set of prefixes and perform some form of proxy aggregation is not currently a well-understood capability. This functionality would have great returns as it would decrease the prefix loading behind the domain that is performing the atomicity/aggregation computations. A key point to remember is that today, many domains advertise unnecessarily specific prefixes to effect the optimal local routing policy. This research points to a way to recapture the hierarchical topology that the providers originally deployed. This would return a multiplier far greater than the factor of two that has previously been discussed. Further, while such techniques have been discussed in the hallways of the IETF, they have never been fully analyzed and put into practice. There are undoubtedly issues here that will need to be addressed and pushing the research in this direction would hold great promise.

Also note that vendors tend to pay attention to research work more if a prototype implementation and some measurement on it has been done. For example, when Van Jacobson came up with the RED idea (Random Early Detection) for congestion management, he implemented it in the NS simulator and had data for the router vendors to look at. They took his code and algorithms and implemented it. If it had just remained a technical report or a SIGCOMM paper it would not have made it into our Internet.

7 Related Ideas

For completeness we point to related ideas that we are aware of.

7.1 Geoff Huston's Atoms

Geoff Huston mentions 'atoms' as well in [5]. Huston's atoms are somewhat different from the atoms in this proposal. Huston's atoms are allocated to ASes; ours are computed by backbone routers. Either approach has its advantages: allocated atoms are conceptually more easily established (not requiring computation), whereas computed atoms are likely to be more effective, in that they are computed based on the similarity of prefixes as observed by the routers. For instance, computed atoms allow prefixes originated by different ASes to be part of the same atom (such as atom A3 in Figures 2 and 3). In addition, computed atoms allow for greater transparency, since non-backbone routers are not involved in any way (Section 5).

However Huston's definition of atoms provides an interesting alternative that may be of use to us. In particular we can use the techniques in Sections 5.1 and 5.2 and apply them to allocated rather than computed atoms. So it should be possible to use allocated atoms as a 'fallback' should it turn out that computed atoms are too expensive or unscalable.

To elaborate a little further on allocated atoms: rather than computing atoms in the backbone of the Internet, ASes throughout the Internet can be allocated atom ids (e.g. by IANA). ASes then announce and withdraw atom ids rather than prefixes in routing update messages. To propagate the mapping between atom ids and prefixes, BGP update messages that include a BGP community attribute [9] can be used. The BGP community attribute contains the atom id, which applies to the prefixes carried by the update message. Such an approach removes the burden of atom computation from backbone routers. On the other hand, it is more drastic in that it requires all ASes in the Internet to cooperate.

7.2 Frank Kastenholz's Aggregates

In [12] Frank Kastenholz introduces a new kind of aggregate. Kastenholz's aggregates are a significantly different concept from atoms. However, Frank Kastenholz is addressing similar issues. Also, like us, Kastenholz separates aggregate ids from aggregate contents, and performs routing computations on aggregate ids instead of prefixes. However, Kastenholz's approach does not remove knowledge of prefixes from any routers. In contrast, we remove prefixes from a subset of routers (the transit routers in Section 5.1), or at least from the forwarding tables (Section 5.2).

We also believe that our approach is somewhat less disruptive than Kastenholz's: we point to ways in which atoms can be applied effectively within the backbone (or part of the backbone) transparently (Sections 5.1 and 5.2).

8 Planning

First three months — A Basic Atom-Based Router In the first three months a basic atom-based router is implemented. This router is based on allocated atoms (Section 7.1). The work of the first period is carried out in the following steps:

1. *Refining and Releasing Atom Computation Scripts* To get Patrick Verkaik started, we begin with documenting Andre Broido's existing Perl scripts for atom computation. These scripts are subsequently released.
2. *Implement Basic Atom-Based Router* Next, we implement a basic atom-based router. The router is designed to work in the framework of Section 5.1, and can function as a transit or edge router. It uses existing router code (such as `gated`) wherever possible. Atoms are not yet computed; they are allocated (Section 7.1), since allocated atoms are more easily implemented.
3. *Release* We release the implementation.

Next three months — Advanced Atom-Based Router In the next three months an advanced atom-based router is implemented. In particular atom computation will be addressed. The following steps are taken:

1. *Refine Atom Computation* We review the above atom computation scripts with the aim of finding a scalable, realtime algorithm to compute atoms. One possible outcome is that atoms are initially computed as in the scripts (i.e. centralised and offline), but subsequently updated in a scalable, realtime way.
2. The improved algorithm is implemented, and incorporated into the atom-based router.
3. *Evaluation* We measure and evaluate the resulting atom-based router, comparing it to a standard BGP router. This could be done using an SSFnet simulation, with RouteViews [13] as a data source to feed the simulation.
4. *Release* We release the implementation.

Notes:

- We intend to have experts looking over our shoulders from the start. Note that Tony Li (Procket Networks), Dave Ward (Cisco Systems), Curtis Villamizar (Avici Systems) and Dennis Ferguson (Juniper Networks) have already expressed interest. In this way we hope to have the results put to practical use for the benefit of the Internet community.

- Unfortunately, the precise start date remains unknown at this time, and depends on issues such as VISA application. We aim to get started at the beginning of September. For Patrick Verkaik to attend the workshop in Leiden in October, we would like Patrick to work from The Netherlands until the workshop, and to fly to San Diego after the workshop. The remainder of his six months will be carried out in San Diego.

- The planning above does not take into account vacation time.

- Four months into the project we review whether there should be an extension of another six months (Section 8.1).

8.1 Extension

Provided the outcome of the first six months is agreeable to all parties, we hope to extend the project for another six months. Ideas for the next six months are:

- Applying the atom concept to other domains. We expect atoms and our implementations to be applicable to other domains such as overlay networks (e.g. Peer-to-Peer and Ad-Hoc networks).

- Publishing the results.

- Creating specifications of the protocols that were derived or modified during the first six months. This involves removing any hacks (use of inappropriate or reserved IP header fields for example). The specification can be the basis of an RFC.

9 Project Members

The proposed project consists of the following members:

Patrick Verkaik	100%
Andre Broido	50%
kc claffy	5-10%

Table 1: Project members.

Note: funding from NLnet is requested only for Patrick Verkaik. CAIDA commits 50% of Andre Broido's time, and 5-10% of kc claffy's time to this project. In addition, several other CAIDA members are expected to spend part of their time on this project as necessary.

References

[1] Andre Broido, kc claffy, 'Analysis of RouteViews BGP data: policy atoms', Proceedings of the Network-Related Data Management workshop, Santa Barbara, May 23, 2001

[2] Andre Broido, kc claffy, 'Complexity of global routing policies', http://www.caida.org/outreach/papers/2001/CGR/

[3] Andre Broido, Evi Nemeth, kc claffy, 'Internet Expansion, Refinement, and Churn', European Transactions on Telecommunications, January 2002, http://www.caida.org/outreach/papers/2001/IERC/

[4] Geoff Huston, 'Analyzing the Internet's BGP Routing Tables', The Internet Protocol Journal, Volume 4, Number 1, March 2001. http://www.telstra.net/gih/papers/ipj/4-1-bgp.pdf

[5] Geoff Huston, 'Scaling Inter-Domain Routing — A View Forward', The Internet Protocol Journal, Volume 4, Number 4, December 2001. http://www.cisco.com/warp/public/759/ipj_4-4/ipj_4-4_scaling.html

[6] J. Postel, 'Internet Protocol', RFC 791, September 1981.

[7] Y. Rekhter, T. Li, 'An Architecture for IP Address Allocation with CIDR', RFC 1518, September 1993.

[8] Y. Rekhter, T. Li, 'A Border Gateway Protocol 4 (BGP-4)', RFC 1771, March 1995.

[9] R. Chandra, P. Traina, T. Li, 'BGP Communities Attribute', RFC 1997, August 1996.

[10] T. Bates, R. Chandra, D. Katz, Y. Rekhter, 'Multiprotocol Extensions for BGP-4', RFC 2283, February 1998.

[11] S. Deering, R. Hinden, 'Internet Protocol, Version 6 (IPv6) Specification', RFC 2460, December 1998.

[12] Frank Kastenholz, 'ISLAY — A New Routing and Addressing Architecture', INTERNET DRAFT. http://partner.unispherenetworks.com/rrg/draft-irtf-routing-islay-00.txt

[13] University of Oregon's RouteViews project,
http://www.antc.uoregon.edu/route-views/