The Internet Protocol Journal

March 2006 Volume 9, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Autonomous Systems Numbers (ASNs) play an important role in the routing architecture of the Internet. An *Autonomous System* (AS) is, according to RFC 4271, "… a set of routers under a single technical administration, using an *interior gateway protocol* (IGP) and common metrics to determine how to route packets within the AS, and using an inter-AS routing protocol to determine how to route packets to other ASs." AS numbers are—like IP addresses—a finite resource, and predictions exist for when the AS number pool will be depleted. In our first article, Geoff Huston explains how ASNs work, and introduces us to the 4-byte ASN scheme that will allow for future growth beyond the currently predicted depletion date.

Our second article looks at another aspect of Internet routing and addressing—the IPv4 number space itself. Designers and operators of internets are often required to perform various address calculations in order to properly configure their networks. Russ White takes us through several exercises and introduces some "tricks of the trade" to make such calculations easier.

Our articles on spam in the last issue of IPJ prompted some feedback from our readers, and promises of more articles from other authors. This problem space clearly has more than a single solution. We look forward to bringing you more coverage of this topic in future editions.

The second issue of the *IETF Journal,* published by the Internet Society, is now available. Some people have asked me if I think of this new journal as a "competitor" to IPJ. I am happy to say that the *IETF Journal* is very much complementary to IPJ and covers important news from the IETF that we hope our readers will find interesting. You can access the *IETF Journal* by visiting: **http://ietfjournal.isoc.org**

The *IPJ Reader Survey* will soon close. We are grateful to the many readers who took the time to tell us about their reading habits, ideas for future articles, and other suggestions. Of course, we always welcome your feedback on any aspect of IPJ. Just drop us a line via e-mail to: **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Exploring Autonomous System Numbers

by Geoff Huston, APNIC

So what are *Autonomous System Numbers* (ASNs), and what role do they play in the technology of the Internet? This article explores the role of ASNs as a critical element of the Internet routing architecture. We will first explore how the AS number space is structured, examine how ASNs are used in the interdomain routing environment and then look at the consumption rate of these numbers, and finally examine our options when we get to the point of likely ASN pool exhaustion. However, in order to put this into context, a brief overview of Internet routing architecture follows.

Internet Routing Architecture

Internet routing architecture is structured as a two-level hierarchy. The environment is first partitioned into *domains* with each domain using an internal routing environment. These network domains use an interior routing protocol (commonly referred to as an *Interior Gateway Protocol* [IGP]), which maintains a complete mapping set for the current internal topology of the domain, together with the set of "best paths" between any two points within the network domain. Although this approach of having a routing protocol automatically maintaining a comprehensive view of the current topology can be made to work within even quite large routing domains, such an approach does not scale to the size of the entire Internet. Fine-grained topology information is useful only in "local" situations, and is best omitted when forming a larger view of the network. Commonly used interior routing protocols include *Open Shortest Path First* (OSPF), *Intermediate System-to-Intermediate System* (IS–IS), and *Enhanced Interior Gateway Routing Protocol* (EIGRP).

The second level in the routing hierarchy is the *interdomain* routing domain. The interdomain routing environment describes how domains interconnect, but avoids the task of maintaining transit paths within each domain. In the interdomain space, a routing path to an address is described as a sequence of domains that must be transited to reach the domain that originates that particular address prefix. Today this interdomain space is maintained using Version 4 of the *Border Gateway Protocol* (BGPv4).

Each routing domain is a single administrative domain, operated within a uniform set of routing policies, and is operated independently from any other domain. The domain is in effect an autonomous unit in the overall routing architecture, and is termed an *Autonomous System* (AS). Each of these ASs is uniquely identified using an *Autonomous System Number* (ASN).

What Is an Autonomous System?

One of the best definitions of an Autonomous System can be found in an IETF document, RFC 4271[4] that describes BGPv4:

> "The classic definition of an Autonomous System is a set of routers under a single technical administration, using an *interior gateway protocol* (IGP) and common metrics to determine how to route packets within the AS, and using an inter-AS routing protocol to determine how to route packets to other ASs. Since this classic definition was developed, it has become common for a single AS to use several IGPs and sometimes several sets of metrics within an AS. The use of the term Autonomous System here stresses the fact that, even when multiple IGPs and metrics are used, the administration of an AS appears to other ASs to have a single coherent interior routing plan and presents a consistent picture of what destinations are reachable through it."

The AS Number Pool

ASNs are drawn from a 16-bit number field, allowing for 65,536 possible values.

AS 0 is reserved, and may be used to identify nonrouted networks. The largest value—AS 65,535—is also reserved. The block of ASNs from 64,512 through 65,534 is designated for private use. ASN 23,456 is reserved for use in ASN pool transition. The remainder of the values, from 1 through to 64,511 (less 23,456), are available for use in Internet routing. The number space is unstructured, because there are no internal fields in the number structure, nor is there any aggregation or summarization capability for ASNs.

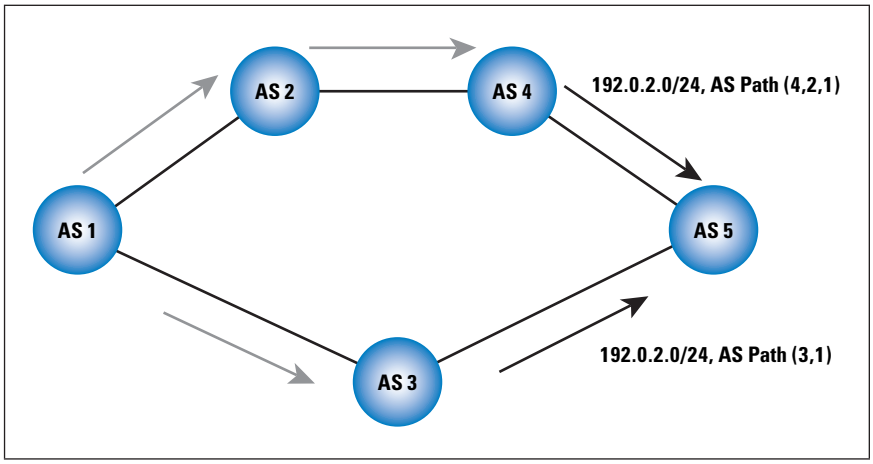
How AS Numbers Are Used in BGP

The interdomain routing space is constructed using two components: *address prefixes* and *AS numbers,* which are used as domain identifiers. Every prefix has an originating domain, known as the *Origin AS* from which reachability for the prefix is propagated across the interdomain space.

As the routing advertisement is propagated across the interdomain space, each prefix accumulates an associated "AS path." When an address prefix advertisement transits a domain, the domain effectively "signs" the prefix advertisement by prepending its ASN to the AS path associated with the address prefix. At any point in the network the AS path describes a sequence of connected domains that forms a path from the current point to the originating domain. This setup is shown in Figure 1, where AS1 originates an advertisement for the address prefix **`192.0.2.0/24`**. At AS5, the AS receives two BGP advertisements for this prefix. One has the AS path (4, 2, 1), and the other has the AS path (3, 1).

The left-most number in the AS path list is the ASN of the adjacent AS from which the address prefix advertisement was received. The sequence of numbers indicates the sequence of ASs though which this update was propagated. The right-most, or final ASN, is the AS number of the AS that originated the address prefix advertisement, or *Origin AS*.

Figure 1: AS Path Generation in BGP



The AS path serves two purposes in interdomain routing: that of a *path length* metric and a *loop detection* mechanism.

The AS path is used as a path metric in the BGP path selection algorithm. When a domain receives two different BGP advertisements for the same address prefix, the default BGP selection process is that of selection of the advertisement of the minimal-length AS path, with each AS in the path counting as a single unit of "cost." In the case of the example network in Figure 1, AS5 prefers to use the path through AS3 to reach the originating AS1, in preference to the longer path of AS4 and then AS2.

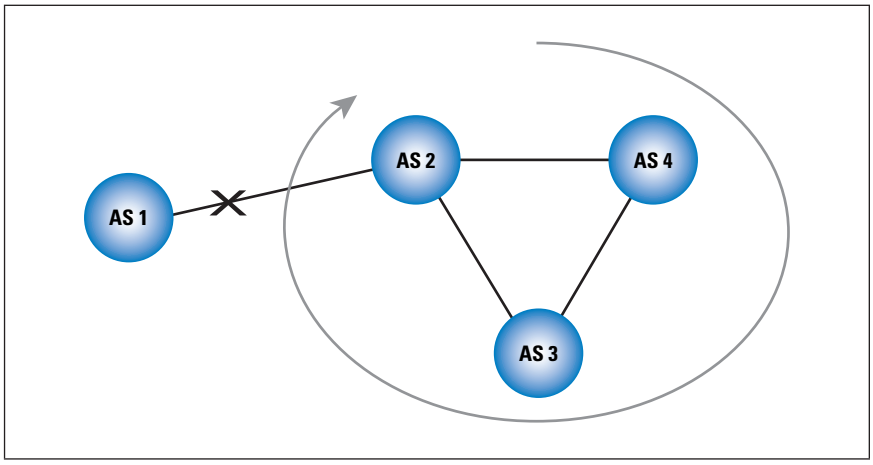
Although enumerating the AS path vector within the routing protocol is one way of passing the path cost through the routing domain, it may appear that the best path selection function could just as easily be supported by carrying a simple path cost metric of a domain transit counter, similar to that used by other distance vector routing protocols, such as *Routing Information Protocol Version 2* (RIPv2). However, the problem with distance vector protocols is the "count-to-infinity" dilemma.

To illustrate the need for explicit AS path enumeration in BGP, consider what happens when the AS path vector is replaced by a simple path cost metric. In the configuration shown in Figure 2, AS1 originates a routing advertisement toward AS2. AS2, AS3, and AS4 are interconnected in a simple loop configuration. When AS2 receives AS1's advertisement with a path cost of 1, it passes the advertisement on to both AS3 and AS4, with a path cost of 2. Both AS3 and AS4 select as their best path this advertisement from AS2 with a path metric 2, corresponding to the AS path (2, 1).

Now if the connection between AS1 and AS2 is broken, then AS2 no longer sees AS1, and withdraws its best path to the prefix through AS1. AS2 then stops advertising a path to AS3 and AS4. But AS3 is already advertising a path to AS4, with a metric of 3, corresponding to the AS path (3, 2, 1). Upon the withdrawal of the advertisement from AS2, AS4 then selects this as its next best path, with a path cost of 3. AS4 then advertises this prefix to AS2 with a path cost of 4, corresponding to the AS path (4, 3, 2, 1).

At this point, without the explicit AS path in the advertisement, AS2 cannot deduce that this advertisement is, in fact, a loop. Accordingly, AS2 accepts this path with a metric of 4 as its best path. AS2 then advertises this to AS3 with a metric of 5, corresponding to the AS path (2, 4, 3, 2, 1). AS3 updates its best path to AS1 with this new metric and then sends an update to AS4, and so on. This process continues around the loop until the path cost metric reaches some defined maximal value. The higher the maximal value for the path cost metric, the longer the time taken to detect the loop condition. The smaller the maximal path cost metric, the smaller the span of network that the protocol can encompass. Setting the maximal path cost parameter requires some considerable care, and the operation of the protocol can be extremely slow to converge in terms of loop detection.

Figure 2: Loop Formation in Distance Vector Protocols



This form of loop can be averted by replacing the path cost counter with a fully enumerated AS sequence. Continuing the example in Figure 2, when AS2 withdraws its route to AS3 and AS4, AS4 still selects the other route it has heard, but this time the selected prefix has the path (3, 2, 1). When AS4 attempts to pass this advertisement to AS2, AS2 sees its own value in the associated AS path and rejects the advertisement. At the same time AS3 withdraws its advertisement to AS4, and at that point the prefix is dropped from the entire routing system. In this way the AS path acts as an efficient routing loop detector.

The use of ASNs and AS path vectors in BGP provides an effective solution to this classic problem of loop detection, as well as providing a simple and effective path-selection process.

Who Needs an AS Number?

Not every network needs to have its own ASN. The guiding principle is that ASNs are used to express distinct interdomain routing policies, and not every network has the requirement to express its own unique set of routing policies.

In the case where a network has a single upstream connection, the routing policies of the network are precisely the same as those of its upstream service provider, and there would normally be no need for the network to use a distinct ASN. Even if the network domain uses BGP for its upstream connection, the originating domain can use a private ASN (from the number range 64,512 – 65,534) to support the BGP session to the upstream network. The upstream network strips off the private ASN when it readvertises the prefix, and the upstream network appears to the rest of the Internet as the originating AS. Even if the AS has "downstream" networks it can still use a private AS, even when the downstream ASs are using public ASNs. The stripping of the private AS removes only the instances of the private AS from the AS path, and not the public ASNs (Figure 3).

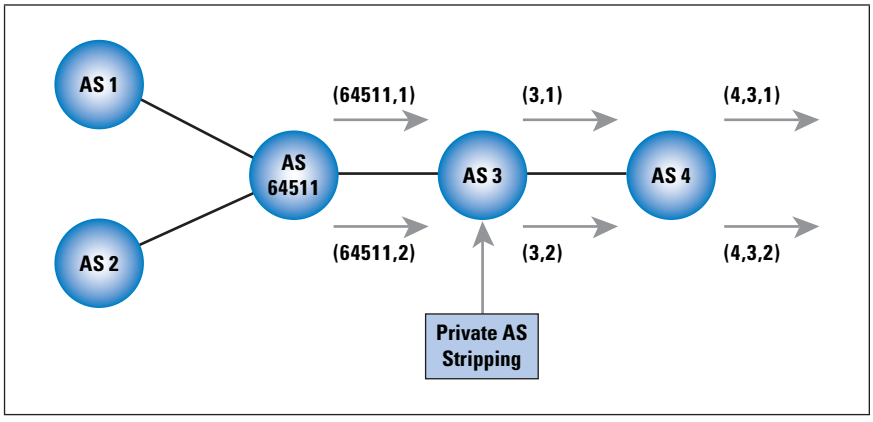


Figure 3: Use of Private AS Numbers

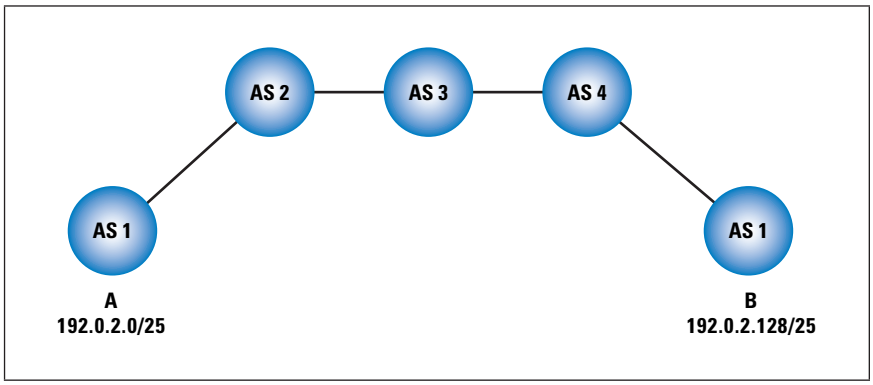
In the case where a network has two or more upstream transit connections, it is more likely that the network will use its own unique ASN. It is not always the case that a distinct ASN is required here, and the distinguishing factor is that of the network wanting to express particular routing policies. Where the network has no particular preference as to which of the upstream services should be used for incoming traffic, the network can also use a private ASN for each of its routing sessions. In such a case the external routing view would be that the prefix appears to be originated from multiple ASs.

In the case where there are multiple paths to reach the network, and where these paths need to be distinguished in the routing system by different AS paths that have the same originating AS (that is, there is a need to express a routing policy), then the network needs to use a unique ASN within the interdomain routing system.

Can an ASN Be Split Across Separated Subdomains?

There are many cases of dispersed networks that exist in multiple locations. If these locations are all administered by a single entity, it may be desirable to use a single ASN across all these domains. This scenario is possible, but considerable care needs to be exercised when designing the routing configuration. Figure 4 shows two distinct subdomains of AS1, and they are not interconnected internally.

Figure 4: Split AS



AS1 (A) advertises the prefix **192.0.2.0/25** to AS2, and this advertisement is propagated to AS2, AS3, and AS4. When AS4 passes this advertisement to the other segment of AS1 (B), this router rejects the advertisement because the associated AS path (4, 3, 2, 1) indicates that the route has already passed through AS1. Similarly, the first segment of AS1 (A) rejects the advertisement of **192.0.2.128/25** from AS2, because its path (4, 3, 2, 1) also indicates that a loop has formed. To restore complete connectivity between the distinct parts of AS1, AS1 needs to configure static routes at its edges. If AS1 (A) configures a static route to **192.0.2.128/25** pointing toward AS2, and AS1 (B) similarly configures a route to **192.0.2.0/25** through AS4, then the configuration enables full connectivity.

In more complex configurations where each of the segments of the network is multiply connected, the static route configuration becomes more complex. However, with very careful configuration, a single ASN can be distributed across multiple distinct networks.

AS Path Prepending and Path Poisoning

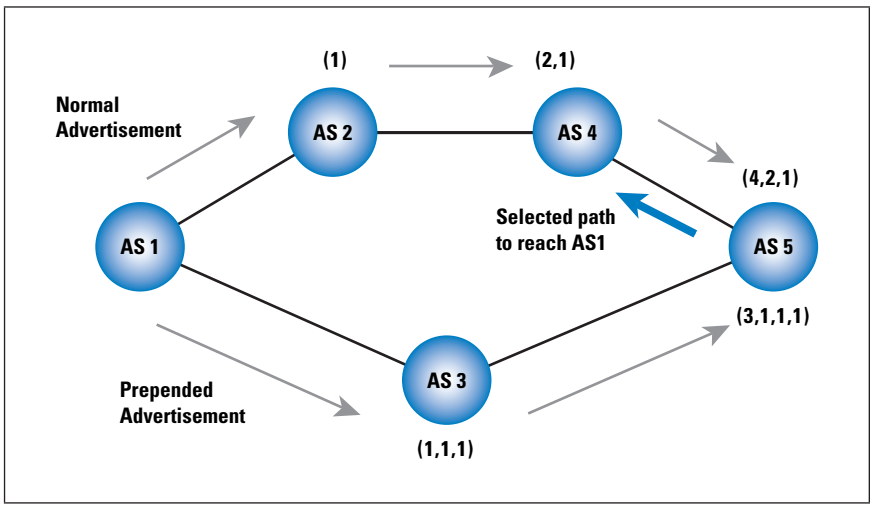
The basic mechanism of path preference in BGP is that of the AS *path length*. Where there are two advertised paths to reach a particular address prefix, the default selection algorithm in BGP is to prefer the advertisement with the *shorter* AS path length.

A multihomed domain may wish to have other domains prefer one particular path over another to reach it. This may be because the local domain wishes to optimize its traffic costs between the multiple upstream providers, balance the traffic load across multiple paths, or set up various forms of primary and backup relationships across the multiple provider upstream paths.

Although such policy preferences are often set up using BGP *communities*, BGP community signaling requires the cooperation of multiple parties in consistent interpretation of the community values. A more coarse form of expressing such policy preferences can be achieved through AS *path prepending*, a technique of deliberately extending the AS path length of a prefix advertisement by adding additional ASNs into the AS path of an advertised prefix. Normally the form of AS path prepending uses the local ASN to perform the prepending.

In the example in Figure 5, AS1 wants to express the policy to prefer incoming traffic through AS2, and use the link to AS3 only as a backup. To achieve this with AS path prepending, AS1 prepends itself twice in the AP path of the advertisement passed to AS3, in order to artificially lengthen the AS3 transit path. AS5 would have normally used the shorted AS path through AS3 to reach AS1. As a result of AS1 artificially lengthening its path to AS3, AS5 now selects the transit path through AS4 and AS2 to reach AS1.

Figure 5: AS Path Prepending

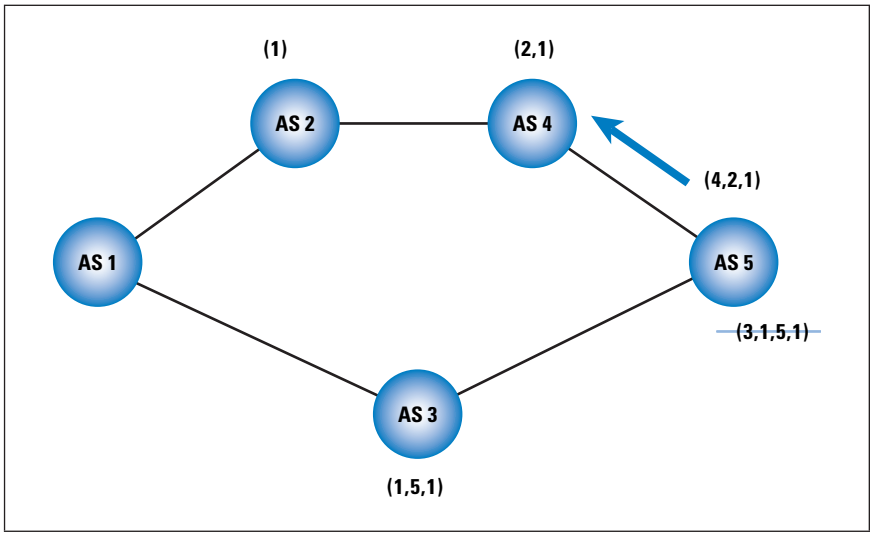


Of course AS path prepending is a very imprecise technique, and can often produce surprising results in real-world situations. A more deterministic method of traffic engineering uses additional signals attached to address prefix advertisements, through BGP communities.

A more subtle, and more controversial, prepending technique is that of so-called AS *path poisoning*, where an AS uses some other value to prepend in the AS path. In Figure 6, AS1 wants to express the policy that under no circumstances should AS5 use the transit through AS3 to reach AS1. In this case AS1 could use AS5 as the prepending value in its advertisement to AS3.

When AS5 receives this advertisement, the presence of its own ASN in the AS path means that it will not accept this advertisement, and prefers the transit path through AS4 and AS2. The difference between these two examples is that in the case where the connection between AS1 and AS2 is broken, none of AS2, AS4, or AS5 can possibly reach AS1 when this AS path poisoning technique is being used.

Figure 6: AS Path Prepending with AS Path Poisoning



AS Number Consumption

In this section we will look at the rate of consumption of ASNs, and estimate when they may be fully consumed. Of the 64,510 available AS numbers, as of January 2006 we have already allocated some 40,000, or well over half of the number pool. Two immediate questions arise—how long do we have before the number pool is completely exhausted, and what are our options for an expanded number pool that can encompass a larger interdomain routing environment?

The Factors for AS Number Consumption

Before looking at these two questions in further detail, it would be useful to understand the factors that affect AS number consumption.

From one perspective it is counterintuitive to assume that the Internet will evolve from tens of thousands of distinct routing domains to one of hundreds of thousands or even millions of distinct routing domains. It may appear that there is a reasonable level of correlation between the number of active *Internet Service Providers* (ISPs) in the Internet and the number of advertised ASNs. If forecasting a future demand for hundreds of thousands or even millions of ASNs, it would appear that we are forecasting continued fragmentation of the service provider industry with large numbers of small enterprises that, collectively, compose the Internet. This scenario does not appear to be likely.

The ISP industry is one with an underlying factor of economies of scale. Larger ISPs generally have access to more efficient use of resources and are more capable of sustaining a market share at competitive prices, with reasonable operating margins because of these economies of scale. Smaller providers tend to service niche markets, and in general are highly susceptible to pricing pressures in the competitive supply market. The overall result is strong pressure for continued aggregation in the service provider market, tending to aggregate to a smaller number of larger providers.

If the number of ASs in use is roughly commensurate to the number of service providers, then this view of the market dynamics would lead to a view that the service provider population is either in a state of equilibrium where the entrance of new niche-oriented players is much the same as the rate at which smaller players are aggregated into larger providers, or one of relatively small growth based on the larger dynamics of continued expansion of the Internet on a global basis.

In practice this has not been the case, and we see a continuous rate of consumption of new ASNs. This rate appears to be some 3,500 ASNs per year, and this consumption rate appears to have been steady since 2002 (see Figure 7). Accordingly, it appears that some additional factors affect AS number consumption rates.

One of these factors is the practice of *multihoming* at the edge of the network. Many end-site networks have business-critical needs for assured Internet connectivity, and a common way to achieve this connectivity is by using the services of two or more upstream providers. In such situations the end site may want to express different routing policies to each upstream provider, and it does so by using its own ASN and expressing these routing policies using BGP to each of its upstreams.

AS numbers are also used in other contexts. In *Multiprotocol Label Switching* (MPLS) Layer 3 networks, one form of generating the *Route Distinguisher* value for a VPN client network is through the use of concatenating the VPN host's AS number with a serial number. To what extent this semiprivate use of AS numbers in a VPN context contributes to the consumption rate of ASNs is difficult to assess, simply because the use of these numbers is not generally visible.

Even within the public Internet there are other contributory factors to AS number consumption. ISPs with diverse product portfolios may wish to express different routing policies for various product families, or express different routing policies in different regions of network coverage. Again this can be achieved through the use of distinct AS numbers of each routing policy set.

An associated contributory factor for AS Number consumption is that there is little incentive for AS Number return and recycling. With the current framework there is no direct cost to maintain an AS number allocation, and the overall characteristic of AS number allocation appears to be a "once and forever" allocation model. When AS numbers are no longer required, AS numbers generally do not return to the unallocated pool for subsequent reallocation.

Taken together, these factors lead to the conclusion that continued AS number consumption is based on a larger set of considerations than the dynamics of the service provider industry.

Accordingly, we can be a little more confident in making the assumption that the factors that have affected AS number consumption in the recent past will continue to be factors in the near-term future, leading to some further confidence in a predictive technique that uses recent consumption data to generate trends that can made predictive forecasts of future demands. We will apply this technique to AS number consumption data to make some forecasts of the time by which the current AS number pool will be exhausted.

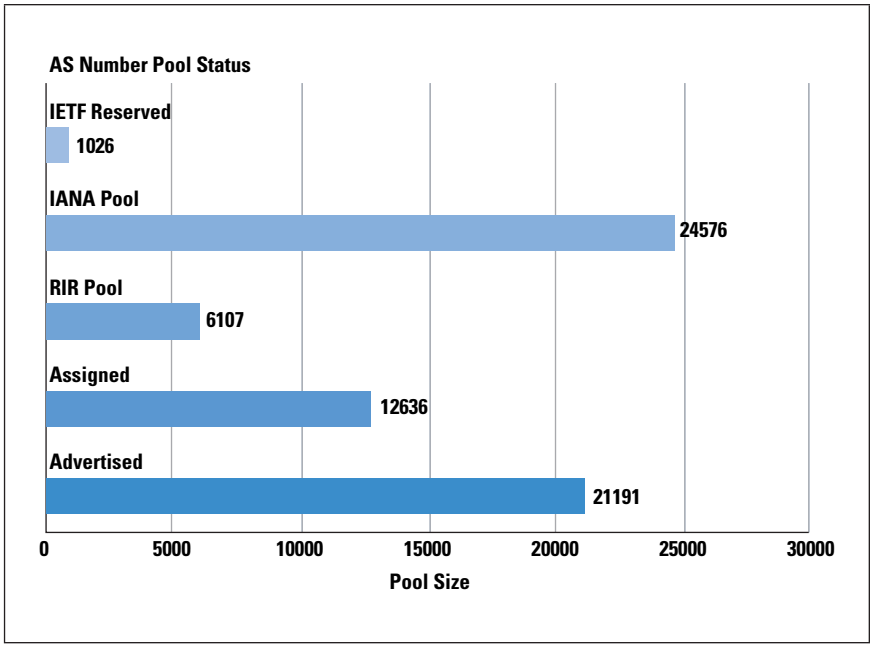
AS Number Pool Status

There are 65,536 AS numbers. As noted already, some 1,026 numbers are reserved and unable to be used in the public Internet, leaving 64,510 for use in the public Internet.

The pool of AS numbers is administered by the *Internet Assigned Numbers Authority* (IANA), and blocks of 1,024 numbers are allocated to the *Regional Internet Registries* (RIRs) periodically when the RIR's pool drops below a threshold level.

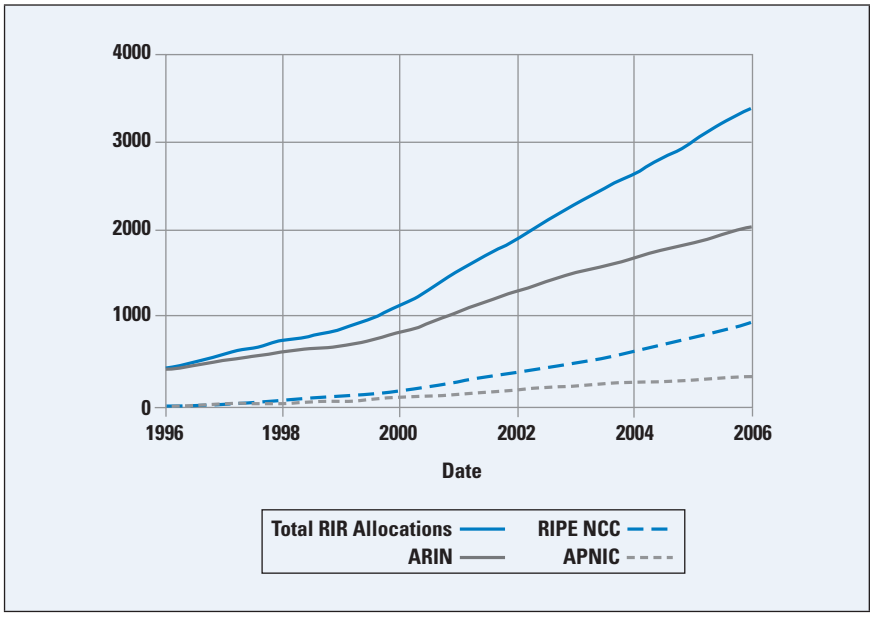
Of the 39,934 AS numbers that have been allocated by IANA by January 2006, there is a further classification of AS numbers. A working pool of numbers is held by the RIR for current assignment to ISPs. Of the assigned AS numbers, some are visibly used in the interdomain routing table of the public Internet, but others are not visible in the Internet. The breakdown of AS numbers into the RIR pool, assigned but not advertised, and assigned and advertised, as of January 2006, is shown in Figure 7. Of the 34,827 assigned AS numbers, some 21,191 are advertised; 12,636 have been allocated in the past, but are not currently advertised in the BGP routing table.

Figure 7: AS Number Status of Advertised, Unadvertised, and Unallocated Pools



The RIRs allocate ASNs to ISPs and end-user networks. A second time series can be generated, showing the cumulative sum of the RIR AS allocations (Figure 8). Not surprisingly, the time series shows the effects of the Internet boom across the period from 1999 through to late 2001 as a sharp upward trend in allocations. The subsequent market correction is also evident as a visible change in the AS allocation rate by early 2002.

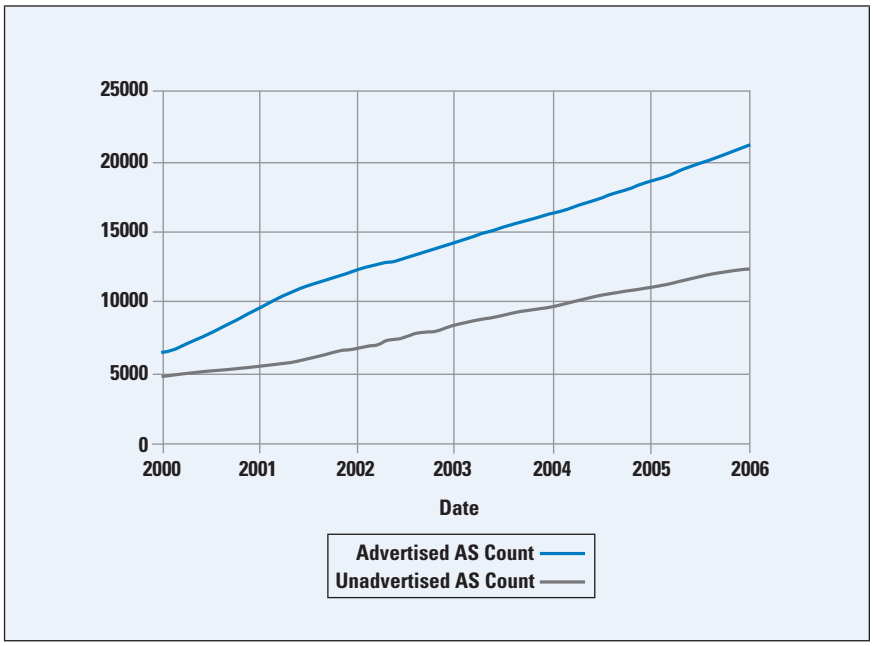
Figure 8: RIR Allocations



BGP AS Advertisements

In addition to allocation rates, a further source of ASN data is the interdomain routing table. The number of distinct ASs advertised in the interdomain routing space of the public Internet has been measured regularly since 1997. The time series of this count of advertised ASNs, and the complementary number of unadvertised ASNs, is shown in Figure 9.

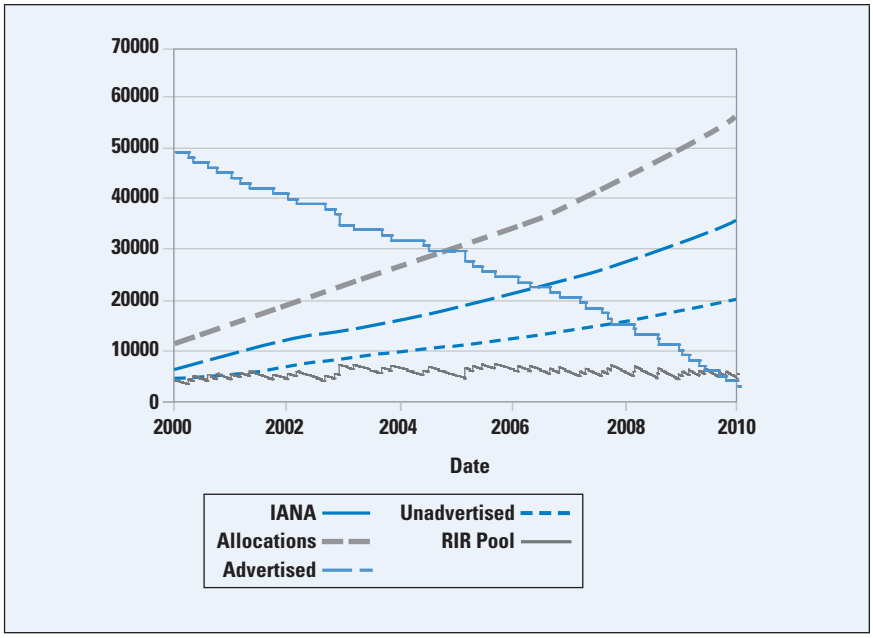
Figure 9: Advertised and Unadvertised AS Numbers



AS Number Consumption Projections

At this point, it is possible to make some projections on AS number consumption. The technique here is to use the past three years' consumption data (taking a starting point of January 2003) and derive an associated exponential function as a best fit to the 3-year data series in order to generate a trend function. This trend function is then projected forward in time to forecast the point in time when the resource reaches a certain threshold point. A considerable amount of detail is associated with this exercise, including the use of an exponential function as the best fit to the past 3 years' ASNs use rates (see **`http://www.potaroo.net/tools/asns/`**). However, for the purposes of this article it is appropriate to proceed to the outcome (Figure 10).

Figure 10: A Predictive Model of AS Number Consumption



From this model it appears that we are looking at steadily accelerating consumption of ASNs, and a projected date of late 2010 of exhaustion available AS numbers to allocate to ISPs.

The implication is that this model indicates that by late 2010 either the Internet should be using a new protocol for interdomain routing that does not rely on AS numbers at all, or, more likely, that the Internet should be using a version of BGP that supports the use of larger AS numbers that are drawn from a number pool significantly larger than 16 bits. The first option appears to be somewhat unrealistic, to say the least. And the second option, although simpler and very much the preferred path, is still going to take some time to deploy, particularly considering the growing size of the interdomain space of the Internet and the diversity of these component domains.

When contemplating a transition to a larger ASN pool, it should be remembered that every day there are more networks that will need to undertake a transition to a longer ASN field in their deployed instances of the BGP protocol.

The steps in this transition path appear to include:

- The completion of the relevant protocol standards for a larger ASN field in BGP
- The production of code in available implementations of BGP that support this protocol standard
- Various forms of testing this code, both in terms of its correct operation and interoperability and in terms of the correctness and viability of the relevant transition steps

- Developing the necessary infrastructural support system to manage the distribution of this new number pool
- A process of deployment of this protocol so that the deployment of larger ASNs can commence well before the point at which the existing AS number pool is exhausted

Even an aggressive schedule of transition across such a large and diverse network as the Internet will take many years to reach the final step. It also appears that a prudent course of action would see us reach that position not by 2010, but by 2008 at the latest, allowing us a margin of some 2 years (and some 10,000 remaining AS numbers) to complete the task.

32-Bit AS Numbers

In this part of the article we will look at the current proposal for a larger AS number pool. As of October 2005, the document defining this proposal is an IETF Internet Draft: **`draft-ietf-idr-as4bytes-12.txt`**. The proposed approach is to expand the size of the AS number pool space from 16 to 32 bits. In number terms this expands the number space from a pool of 65,536 numbers to 4,294,967,296 numbers. In terms of the current use of ASNs, the current scaling properties of the BGP routing protocol, and the use of ASs in the context of interdomain routing, a pool of some 4.3 billion numbers would easily encompass a network environment of significantly greater levels of domains, and interdomain interconnection density. Such a pool size would exceed some current guesses of the scaling capabilities of the BGP protocol by up to a further two orders of magnitude.

It is also proposed to preserve the first block of 65,536 32-bit ASNs to align with the allocations of the 16-bit numbers.

> Let's use a new form of terminology here for 32-bit ASN values, where the first 65,536 ASNs are numbers that use the form "**`0.0`**" through "**`0.65535`**." The second set of 65,536 numbers would be written as **`1.0`** through to **`1.65535`**, and so on. So here we will be using a number format of <*upper16 bits*>.<*lower 16 bits*>.

What is the inventory of concerns that need to specifically addressed in the transition to these 32-bit AS Numbers?

Obviously there is a need for some changes to the routing protocol, and an ordered interdomain transition is unrealistic to expect. More reasonable is an expectation of a piecemeal transition of domains, where individual domains transition their BGP platform to supporting 32-bit ASs in their own time. Domains that are currently using 16-bit ASs may have less reason to undergo an early transition to 32-bit AS support, whereas those domains that are assigned a nonmappable 32-bit ASN will find that they have to support 32-bit ASNs from the outset.

32-Bit Changes to BGP

BGP has two major parts within its protocol: opening up a BGP conversation with a peer BGP speaker, and then the transfrerring of protocol objects that describe reachability of address prefixes and associated attributes of these address prefixes. Both parts include AS Number components, and in considering changes to the current protocol, both parts of the protocol require some change. The message objects that need to be considered here are, therefore, the BGP OPEN message and the BGP UPDATE message.

The changes to BGP create a "NEW" BGP implementation that is capable of supporting a 32-bit ASN environment. The essential task of the changes is to define mechanisms that all NEW BGP speakers use to speak to each other and pass all ASN values in 32-bit fields. However, the Internet is way too large to set up a "flag day" at which point the entire collection of BGP speakers will undertake a switch from "OLD" BGP to NEW BGP. Accordingly, it is also necessary to define protocol interactions in NEW BGP where the transition in the Internet will be gradual and essentially uncoordinated. NEW BGP speakers will have to set up sessions with OLD BGP speakers, and of course OLD BGP speakers will also be peering with other OLD BGP speakers. The information associated with 32-bit AS paths must be passed across sections of the network that normally support only 16-bit AS paths. In other words, 32-bit AS information needs to be passed to OLD BGP speakers and between OLD BGP speakers.

The general approach adopted for transition is preserve AS path length information across the OLD and NEW BGP boundaries, while recognizing that some 32-bit AS information cannot be cleanly mapped into a 16-bit AS path. In order to preserve 32-bit information—a necessary step to prevent loop formation for 32-bit ASs—the 32-bit information is preserved across OLD transit paths and restored upon reentry into NEW BGP realms (Figure 11).

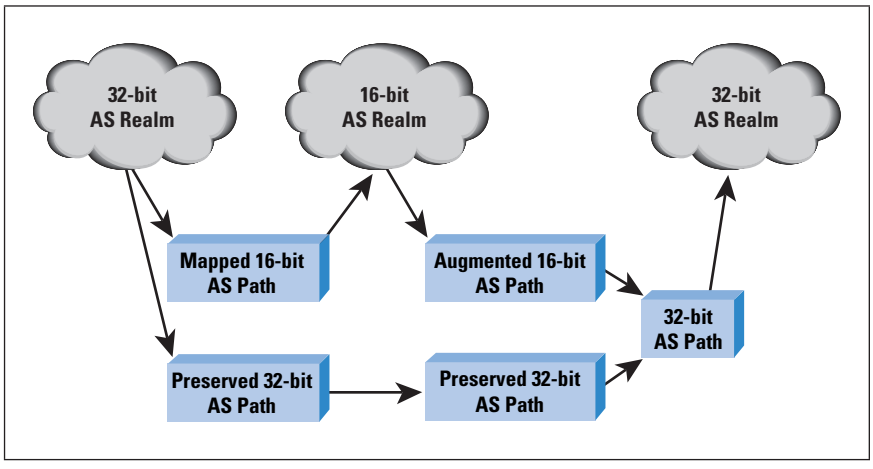


Figure 11: 16-Bit and 32-Bit AS Realms

Opening a BGP Session

The proposed approach is to initiate a NEW BGP session in a mode that is compatible with the OLD BGP protocol, and also inform the remote peer of its capability to conduct a NEW BGP conversation if the remote peer is also a NEW BGP speaker. NEW BGP speakers who open a peer session with an OLD BGP peer will ignore the NEW capability and operate their BGP peer session in OLD mode. A NEW BGP peer will respond positively to the NEW capability, and that BGP session can then operate in NEW mode.

The BGP OPEN message includes a fixed-length 16-bit *My_AS* field as well as potentially containing a capability query as part of the *Optional Parameters* section. In order to ensure that NEW and OLD speakers can communicate, this 16-bit *My_AS* field needs to be preserved in NEW BGP even when the Optional Parameters section includes the capability to undertake a NEW peering session. This may appear contradictory in the first instance, because the OPEN message then contains both a 16-bit ASN and a 32-bit AS *Capabilities Query*. The mechanism proposed for the OPEN message varies according to whether the NEW speaker is using a mappable ASN drawn from the original pool (that is, with a *My_AS* number in the range 0.0 through 0.65535), or it is using a number drawn from a higher-numbered 32-bit number block. In the first case the OPEN message would use the 16-bit mapped value in the *My_AS* field (dropping out the zero-valued high-order 16 bits of the AS value), whereas in the second case the BGP speaker would use for *My_AS* a special 16-bit value that is reserved for this purpose (AS 23456). In both cases the Optional Parameter section would include a capability code to indicate that the local BGP speaker can support 32-bit ASNs (Capability Code 65).

The side effect is that from the perspective of OLD BGP domains AS 23456 may appear to be connected to the interdomain network in many different locations. From the OLD BGP realm this does not present a protocol problem, although, as always, there is the potential here that this repeated use of AS 23456 as a 32-bit AS substitution token may create a somewhat confusing BGP view of the Internet from the perspective of the OLD BGP world.

The capability exchange uses a protocol described in RFC 3392. The NEW BGP speaker adds an optional capability field to the OPEN message. The 32-bit AS capability code 65 carries as its capability value the local 32-bit local ASN value. For a NEW peer this capability value is to be interpreted as the actual AS of the remote side, on the basis that the *My_AS* field in the body of the OPEN is either a truncation of the local 32-bit AS value (in the case of mappable 32-bit AS values), or the special value of AS 23456.

The BGP UPDATE Message

For a NEW BGP session (32-bit peering with 32 bits) the changes to the protocol are the use of 32-bit ASNs in the AS_PATH attribute of UPDATE messages. All 16-bit AS values are padded with a zero high-order 16 bits. If the AGGREGATOR attribute is used, it is similarly carried as a 32-bit value. So in the 32-bit peering, all 16-bit information is carried in mapped 32-bit ASNs (Figure 12).

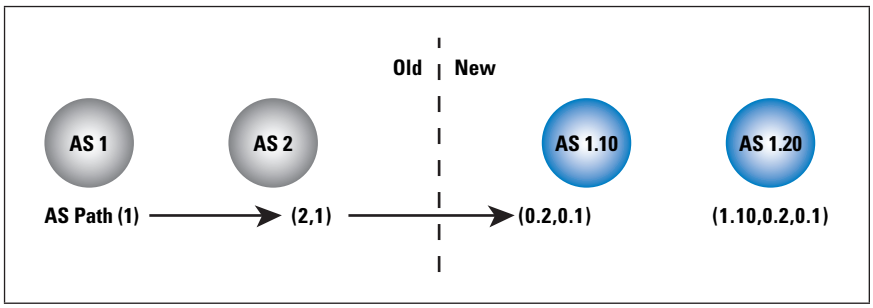


Figure 12: OLD to NEW BGP AS Path Mapping

In this way AS path length is preserved without change when translating 16-bit AS information into the 32-bit domain.

The next case is where an OLD BGP peers with a NEW BGP. We have already seen the simple case where the information is coming from a 16-bit path and there is no additional 32-bit information, and in this case the 16-bit values are simply mapped into 32-bit values, by padding the ASN values with 16 zero high-order bits. What about the reverse case where 32-bit information is being passed back into the 16-bit world?

This case has two parts: first creating an equivalent 16-bit AS path and second, packing up the 32-bit AS path information in such a way that it transits across the 16-bit domain in such a manner that that it can be reassembled in any subsequent transition into a 32-bit domain. In the first case, the equivalent path information is constructed by stripping the high-order 16 bits off the AS value, as long as this part is all zeros. Where this is not possible—and the AS path contains one or more ASNs with non-zero high-order bits—then the transition ASN, 23456, is substituted in the place of each such ASN in the AS path. In this way the AS path length metric is preserved, and the prevention of count-to-infinity loops in the 16-bit domain is avoided.

The second part to this case is packaging up the 32-bit path into the OLD BGP session in such a way that it can be unpacked at any subsequent boundary back into a 32-bit routing realm. Here the proposal calls for new transitive community attributes to be carried in OLD BGP routing realms. These attributes are defined as transitive attributes, and should be passed through the OLD BGP peering sessions without alteration. It should be noted that this is not a protocol change as such, but it does require the explicit configuration support within OLD BGP implementations of this attribute as a transitive community.

The proposed mechanism is an extended community attribute called "NEW_AS_PATH." When a NEW BGP speaker is speaking to an OLD BGP, the NEW BGP prepends its own AS value to the AS_PATH and copies this information into the NEW_AS_PATH attribute. It then translates the 32-bit AS path into a 16-bit equivalent AS path. The translation is straightforward, in that where the 32-bit AS has all zeros in the high-order 16 bits, the translation truncates the AS value to a 16-bit value, and where the high-order 16 bits are nonzero, the translation substitutes the reserved 16-bit value AS 23456 in its place (Figure 13).

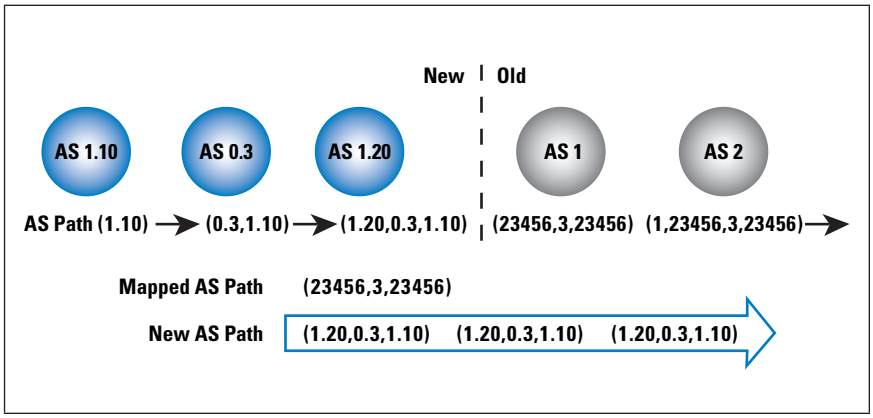
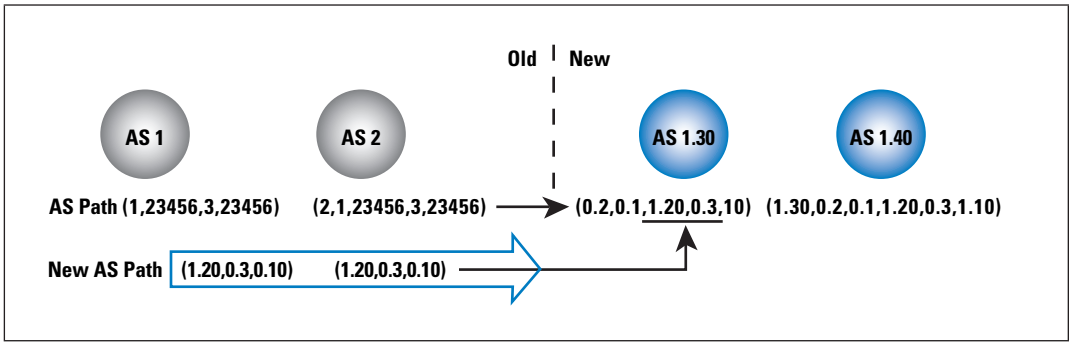


Figure 13: NEW to OLD BGP AS Path Mapping

The transit across the OLD BGP domains leaves the NEW_AS_PATH untouched, and prepends 16-bit AS values to the AS_PATH. In other words, OLD BGP behaves as it always has. The NEW_AS_PATH is passed through the OLD realm as an opaque bit block.

The next transition is one from the OLD to the NEW domain when a NEW_AS_PATH attribute is present. In this case the NEW BGP speaker takes the AS path as presented by the OLD BGP speaker and converts the 16-bit AS values to 32-bit AS values by adding 16 bits of zero padding to each entry, as before. However, in this case the NEW BGP speaker then overwrites the trailing entries with the values specified by the NEW_AS_PATH attribute. The effective result is that the 32-bit AS path that entered the 16-bit sequence is prepended with the equivalent of the 16-bit transit AS sequence. The NEW_AS_PATH attribute is then removed from the BGP Update, leaving an intact 32-bit path as the AS_PATH attribute. This scenario ensures that the resultant BGP environment can accurately detect loops in both the NEW 32-bit and OLD 16-bit realms (Figure 14).

Figure 14: OLD to NEW BGP AS Path Mapping



What if there was a routing loop that traversed a mixed sequence of NEW and OLD routing realms? The restoration of the original 32-bit AS path at the OLD-to-NEW transition ensures that the potential loop is discarded, because a 16-bit AS sees its own AS in the 16-bit AS_PATH attribute, and a 32-bit AS also sees its own value in the 32-bit AS_PATH. The transition mapping ensures that the potential routing loop is detected by BGP.

The ability to perform AS path prepending is also unaltered in this mixed NEW and OLD BGP environment. The AS simply prepends its local AS value to the AS_PATH as usual. In the case of prepending on a NEW-to-OLD boundary, the prepended AS path is mapped into the NEW_AS_PATH attribute as described previously.

Earlier in this article we noted the less common use of AS path poisoning, where the prepending uses a different ASN value in order to ensure that the particular advertisement is not learned by a remote AS. For NEW BGP speakers there is no change to this capability. For OLD BGP speakers the AS path poisoning can be directed only toward 16-bit ASs, because the OLD BGP speaker has no knowledge of the structure or content of the NEW_AS_PATH attribute.

Another part of the BGP protocol that uses ASNs is the AGGREGATOR attribute. This attribute is attached to an update message when an AS combines two or more prefixes into a single aggregate prefix (a practice that is often referred to as "proxy aggregation"). The ASN of the aggregating AS is attached to the aggregate prefix advertisement as an AGGREGATOR attribute. The same ASN translation technique applies to AGGREGATOR attribute when an advertisement is passed across a transition point. In a NEW-to-OLD transition the AGGREGATOR may be a mappable ASN, in which case the value is truncated to 16 bits and no further action is required. Otherwise the 32-bit AGGREGATOR value is rewritten into a NEW_AGGREGATOR attribute and the transition 16-bit value, AS 2356, is placed into the AGGREGATOR attribute. On an OLD-to-NEW transition the NEW_AGGREGATOR attribute is copied back into the AGGREGATOR attribute, if defined; otherwise the AGGREGATOR is padded out with leading zeros.

Transition

Transition in this scheme is relatively straightforward. NEW BGP speakers can be deployed within the network in a piecemeal fashion without any major concerns, and no changes are required for OLD BGP speakers. The size of BGP UPDATE messages is slightly longer because of the extended length of the AS PATH attribute in NEW BGP and the NEW_AS_PATH attribute that has been added in the OLD BGP environment, but it should not prove to be a major factor.

BGP loop prevention appears to be adequately addressed in all commonly encountered situations, and there appears to be no other significant transition considerations from the perspective of BGP platforms.

This scenario implies a relatively straightforward transition, in that OLD BGP speakers do not have to migrate to NEW BGP capability just because 32-bit ASNs are deployed elsewhere in the network. As long as they transmit the NEW-AS_PATH update across their domain without attempting to alter it in any way, then the 32-bit routing realm will be able to perform loop detection and shortest AS path selection in a manner that is entirely consistent with the 16-bit routing realm. Deployment of NEW BGP code is required only when the local AS is numbered from the nonmappable 32-bit ASN space.

Alternatives to AS Numbers

It is certainly a challenging task to contemplate an environment in which a 32-bit ASN space is exhausted, but one would suppose that the same consideration was in the minds of the original BGP protocol designers when they opted to use 16-bit ASNs. Of course a 32-bit number pool is not double the pool size of a 16-bit number pool—it is 65,536 times larger. That does appear to lead one to believe that this time it will be a far more challenging task to exhaust this expanded number pool.

This approach of simply extending the number space appears to offer a path of minimal disruption and minimal change in terms of operational configuration, storage, message size, and processing overheads for BGP. Nothing much has changed here except the range of the number space, and some ancillary considerations relating to transitional arrangements.

Of course, other labeling spaces remain possibilities, and a shift to a different labeling scheme could well use the same transitional approach. There is no significance in the ASN apart from its uniqueness, and any other form of name space would function equally well in terms of its role in BGP. One could use strings such as domain names, URIs, fixed-length hashes of public keys, the public keys themselves, or even IPv6 addresses as distinguishing AS identifiers.

There is no direct requirement for summarization of ASN ranges within the protocol use, no requirement within the protocol to continue to use number identifiers, and no direct requirement to stick with values that are encoded in a fixed-length field.

However, such approaches would add to the size of BGP UPDATE messages, increase the storage requirements, and, perhaps marginally, increase processing overheads for BGP. The more complex the identity space the more complex the basic task of BGP configuration and the higher the possibility of mistakes. "Borrowing" AS identifiers from another name space, such as domain names, or derived URIs, has the associated concern that the uniqueness of the space is derived from the inherent stability and uniqueness of the name space upon which the identifiers are derived. It is definitely possible that at times this trust is misplaced.

Numbers are often the simplest of identifiers. This approach represents minimal change to the installed base of BGP speakers, and there is no requirement for an existing routing domain using a 16-bit ASN and OLD BGP to make any changes to its routing environment at all. The transition appears to offer flexibility, orderly transition, and minimal disruptions to existing operational practices.

Conclusion

We are certainly running out of available 16-bit ASNs, and an industry of the size of the Internet is no longer as agile as it may have been in the past to make the necessary adjustments to alleviate this situation. At present we need to have a considerable period of advance warning of change in something as fundamental as the interdomain routing space in order to be able to integrate changes into various operational cycles of testing and transitional deployment prior to integration into production environments. The first steps that need to be taken are the completion of the technical specification of this approach in the form of an Internet standard and the production and distribution of BGP implementations that support 32-bit ASNs from the existing BGP implementation suppliers. It would be preferable to get this transition process under way in the near future, while there is still time to complete the transition well before we exhaust the current 16-bit ASN space.

For Further Reading

[1] "BGP Support for Four-octet AS Number Space," E. Chen, Q, Vohra, work in progress, (**draft-ietf-idr-as4bytes-12.txt**), November 2005.

The 32-bit AS description and the associated transition considerations. This work is expected to be completed shortly, and published as an RFC as a proposed standard document.

[2] "The AS Number Report", G. Huston, (updated on a daily basis) **http://www.potaroo.net/tools/asns**

A longer description of the numerical analysis used in the prediction of AS Number exhaustion.

[3] "ASN Missing in Action", H. Uijterwaal, R. Wilhelm, Document RIPE-353, (**http://www.ripe.net/docs/ripe-353.html**), October 2005.

Another analysis of AS Number consumption has been performed by Henk Uijterwaal and Rene Wilhelm, using the RIR AS number allocation rate as the base for the predictive exercise.

[4] "A Border Gateway Protocol 4 (BGP 4)," Y. Rekhter, Ed. T. Li, Ed. S. Hares, Ed., RFC 4271, January 2006.

[5] "Capabilities Advertisement with BGP-4," R. Chandra, J. Scudder, RFC 3392, November 2002.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for almost two decades, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector, and has served time with Telstra, where he was the Chief Scientist in the company's Internet area. Geoff is currently the Internet Research Scientist at the Asia Pacific Network Information Centre (APNIC). He has been a member of the Internet Architecture Board, and currently co-chairs two Working Groups in the IETF. He is author of a number of Internet-related books. E-mail: **gih@apnic.net**

Working with IP Addresses

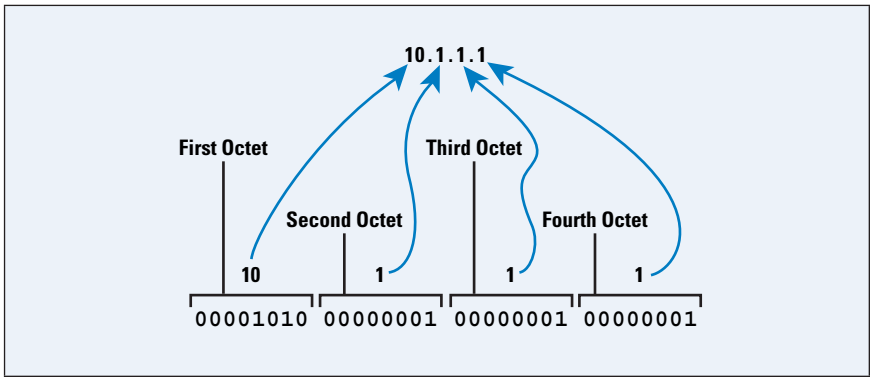
by Russ White, Cisco Systems

IP addresses, both IPv4 and IPv6, appear to be complicated when you first encounter them, but in reality they are simple constructions, and a using a few basic rules will allow you to find the important information for any situation very quickly—and with minimal math. In this article, we review some of the basics of IPv4 address layout, and then consider a technique to make working with IPv4 addresses easier. Although this is not the "conventional" method you might have been taught to work with in IP address space, you will find it is very easy and fast. We conclude with a discussion of applying those techniques to the IPv6 address space.

Basic Addressing

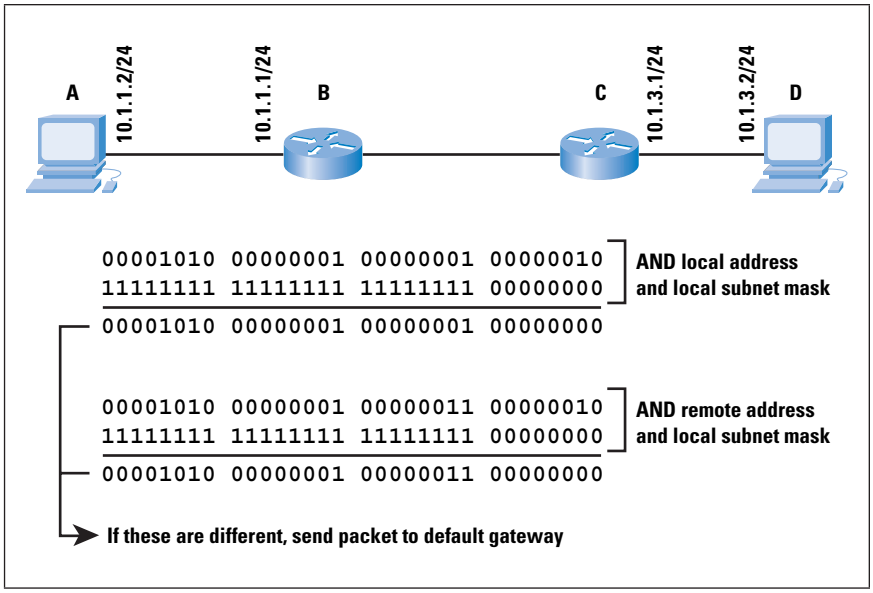
IPv4 addresses are essentially 32-bit binary numbers; computer systems and routers do not see any sorts of divisions within the IPv4 address space. To make IPv4 addresses more human-readable, however, we break them up into four sections divided by dots, or periods, commonly called "octets." An octet is a set of eight binary digits, sometimes also called a "byte." We do not use byte here, because the real definition of a byte can vary from computer to computer, whereas an octet remains the same length in all situations. Figure 1 illustrates the IPv4 address structure.

Figure 1: IPv4 Address Structure



Because each octet represents a binary (base 2) number between 0 and 2^8, each octet will be between 0 and 255. This part of IPv4 addresses is simple—but what about subnet masks? To understand a subnet mask, we need to understand how a device actually uses subnet masks to determine where to send a specific packet, as Figure 2 illustrates.

Figure 2: Subnet Masks



If host A, which has the local IP address **`10.1.1.2`** with a subnet mask of **`255.255.255.0`**, wants to send a packet to **`10.1.3.2`**, how does it know whether D is connected to the same network (broadcast domain) or not? If D is connected to the same network, then A should look for D's local Layer 2 address to transmit the packet to. If D is not connected to the same network, then A needs to send any packets destined to D to A's local default gateway.

To discover whether D is connected or not, A takes its local address and performs a logical AND between this and the subnet mask. A then takes the destination (remote) address and performs the same logical AND (using its local subnet mask). If the two resulting numbers, called the *network address* or *prefix*, match, then the destination must be on the local segment, and A can simply look up the destination in the *Address Resolution Protocol* (ARP) cache, and send the packet locally. If the two numbers do not match, then A needs to send the packet to its default gateway.

Note: ARP is a protocol used to discover the mappings between the IP addresses of devices attached to the same network as the local device and the Layer 2 address of devices attached to the same network as the local device. Essentially, a device sends an ARP broadcast containing the IP address of some other device it believes to be connected, and the device with the specified IP address replies with its Layer 2 address, providing a mapping between these two addresses.

If a subnet mask is a "dotted decimal" version of the binary subnet mask, then what is the prefix length? The prefix length is just a shorthand way of expressing the subnet mask. The prefix length is the number of bits set in the subnet mask; for instance, if the subnet mask is **`255.255.255.0`**, there are 24 1's in the binary version of the subnet mask, so the prefix length is 24 bits. Figure 3 illustrates network masks and prefix lengths.

Figure 3: Prefix Lengths

Binary Mask	Prefix Length	Subnet Mask
11111111 00000000 00000000 00000000	/8	255.0.0.0
11111111 10000000 00000000 00000000	/9	255.128.0.0
11111111 11000000 00000000 00000000	/10	255.192.0.0
11111111 11100000 00000000 00000000	/11	255.224.0.0
11111111 11110000 00000000 00000000	/12	255.240.0.0
11111111 11111000 00000000 00000000	/13	255.248.0.0
11111111 11111100 00000000 00000000	/14	255.252.0.0
11111111 11111110 00000000 00000000	/15	255.254.0.0
11111111 11111111 00000000 00000000	/16	255.255.0.0
11111111 11111111 10000000 00000000	/17	255.255.128.0
11111111 11111111 11000000 00000000	/18	255.255.192.0
11111111 11111111 11100000 00000000	/19	255.255.224.0
11111111 11111111 11110000 00000000	/20	255.255.240.0
11111111 11111111 11111000 00000000	/21	255.255.248.0
11111111 11111111 11111100 00000000	/22	255.255.252.0
11111111 11111111 11111110 00000000	/23	255.255.254.0
11111111 11111111 11111111 00000000	/24	255.255.255.0
11111111 11111111 11111111 10000000	/25	255.255.255.128
11111111 11111111 11111111 11000000	/26	255.255.255.192
11111111 11111111 11111111 11100000	/27	255.255.255.224
11111111 11111111 11111111 11110000	/28	255.255.255.240
11111111 11111111 11111111 11111000	/29	255.255.255.248
11111111 11111111 11111111 11111100	/30	255.255.255.252
11111111 11111111 11111111 11111110	/31	255.255.255.254
11111111 11111111 11111111 11111111	/32	255.255.255.255

Working with IPv4 Addresses

Now that we understand how an IPv4 address is formed and what the subnet length and prefix length are, how do we work with them? The most basic questions we face when working with an IP address follow:

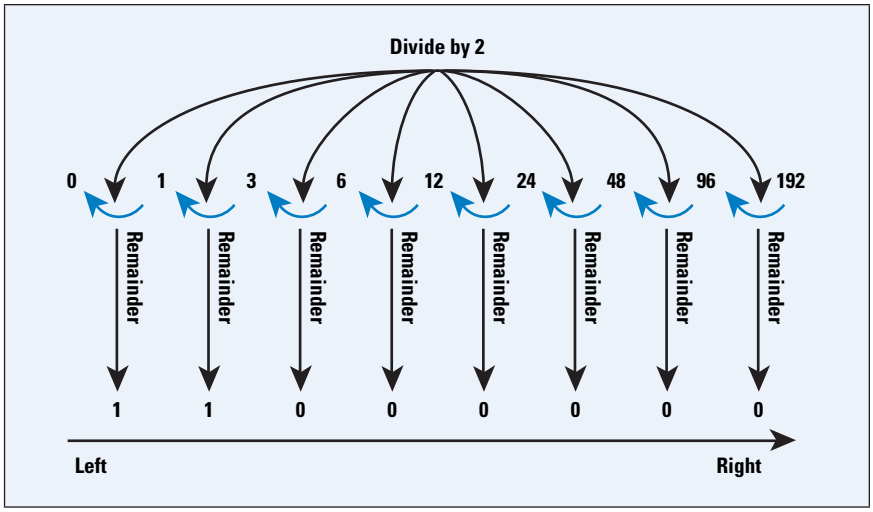
- What is the network address of the prefix?
- What is the host address?

There are two ways to find the answers to these questions: the hard way and the easy way. We cover the hard way first, and then show you the easy way.

The Hard Way

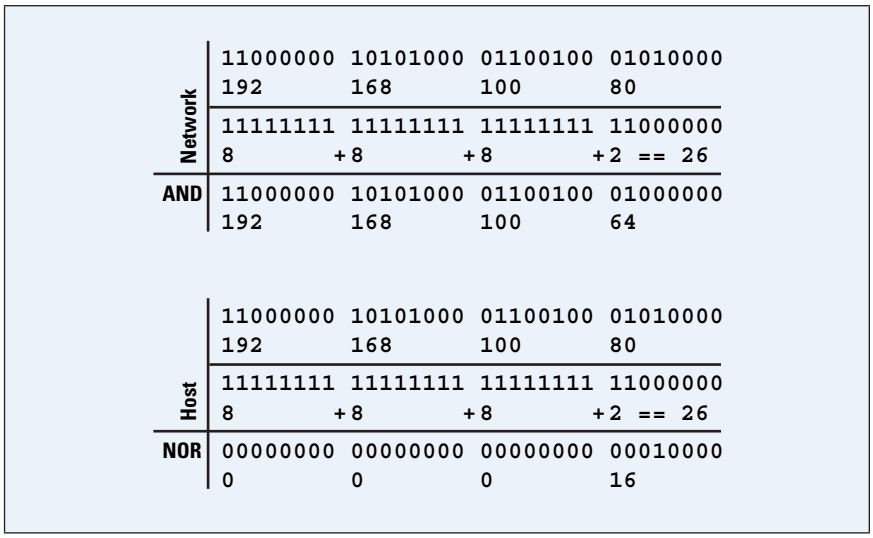
The hard way to determine the prefix and host addresses is to convert the address into binary, perform logical AND and NOR operations on the address and the subnet mask, and then convert the resulting numbers back to decimal. Figure 4 illustrates the process of converting a single octet of the IPv4 address into binary; the number converted in this case is 192.

Figure 4: Binary Conversion



The process is simple, but tedious; divide the octet value by 2, take the remainder off, and then divide by 2 again, until you reach 0. The remainders, reversed in direction, are the binary numbers representing the value of the octet. Performing this process for all four octets, we have the binary IP address, and can use logical AND and NOR operations to find the prefix (network address) and the host address, as Figure 5 shows for the address **`192.168.100.80/26`**.

Figure 5: Address Calculation



The Easy Way

All this conversion from binary to decimal and from decimal to binary is tedious— is there an easier way? Yes. First, we start with the observation that we work only with the numbers within one octet at a time, no matter what the prefix length is. We can assume all the octets before this *working octet* are part of the network address, and octets after this *working octet* are part of the host address.

The first thing we need to do, then, is to find out which octet is our *working octet*. This task is actually quite simple: just divide the prefix length by 8, discard the remainder, and add 1. The following table provides some examples.

Address	Hard Math	Working Octet
`192.158.100.80/26`	(26 ÷ 8) + 1 = 4	4
`10.1.1.48/23`	(23 ÷ 8) + 1 = 3	3
`172.31.80.10/22`	(22 ÷ 8) + 1 = 3	3

Note: Another way to look at this task is that you will ignore the octets indicated by the division. For instance, for **`192.168.100.80/26`**, *the result of dividing 26 by 8 is 3, so you will ignore the first three octets of the IP address, and work only with the fourth octet. This process has the same result.*

When we know the working octet, what do we do with it? Well, we could simply use the procedure outlined, convert the single octet to binary, perform AND and NOR operations on it with the right bits from the subnet mask, and then put it all back together to find the network and host addresses—but there is an easier way to find the network and host parts of the working octet. Start by doing the same math, only this time we want to work with the remainder rather than the result.

`192.168.100.80/26`
26 ÷ 8 = 3 with a remainder of 2

Take the remainder, and use the following table to find the corresponding *jump* within the octet; this number is the distance, in decimal form, between the network addresses within the octet.

1	2	3	4	5	6	7	8
128	64	32	16	8	4	2	1

In this chart, the first line represents the prefix length *within this octet*, the second line represents the prefix value when this bit is set to 1, the number of hosts in the subnet for this prefix length, and the *jump* between network addresses with the specified prefix length.

The number 2 corresponds to 64, so the jump is 64—there is a network at 0, 64, 128, 192, and 224 in this octet. Now all we need to do is figure out which one of those networks this address is in. This task is fairly simple: just take the largest network number that fits into the number in the working octet. In this case, the largest number that fits into 80 is 64, so our network address is **`192.168.100.64/26`**.

Now, what about the host address? That is easy when we have the network address—just subtract the network address from the IP address, and you have the host address within the network: 80 – 64 = 16. This process takes a little practice, but it is not hard when you become accustomed to the steps.

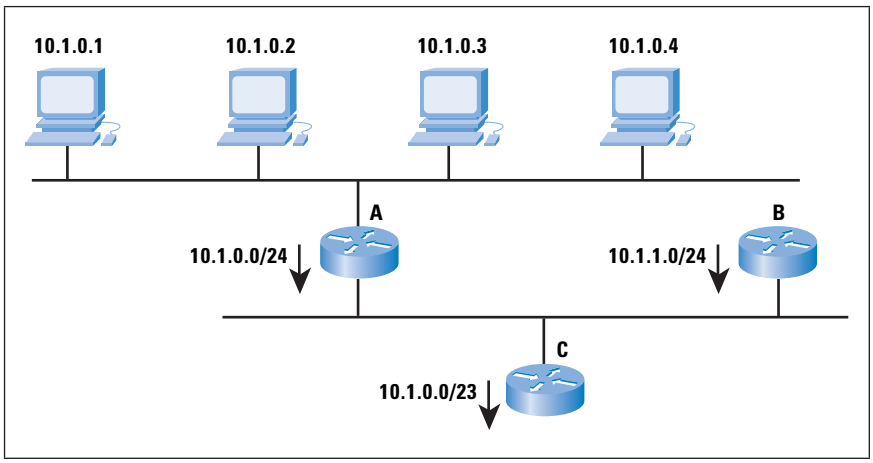
Address	Hard Math	Working Octet	Remainder	Jump	Network	Host
192.158.100.80/26	26 ÷ 8 = 3	3 + 1 = 4	2	64	192.168.100.64/26	80 – 64 = 16
10.1.1.48/23	23 ÷ 8 = 2	2 + 1 = 3	7	2	10.1.0.0/23	1 – 0 = 1.48
172.31.80.10/22	22 ÷ 8 = 2	2 + 1 = 3	6	4	172.31.80.0/22	80 – 80 = 0.10

In the second and third examples, you see that the working octet is actually the third, rather than the fourth, octet. To find the host address in these examples, you simply find the host address in the third octet, and then "tack on" the fourth octet as part of the host address as well, because part of the third octet—and all of the fourth octet—are actually part of the host address.

Summarization and Subnets

Subnets and supernets are probably the hardest part of IP addressing for most people to understand and handle quickly, but they are both based on a very simple concept—*aggregation*. Figure 6 shows how aggregation works.

Figure 6: Address Aggregation



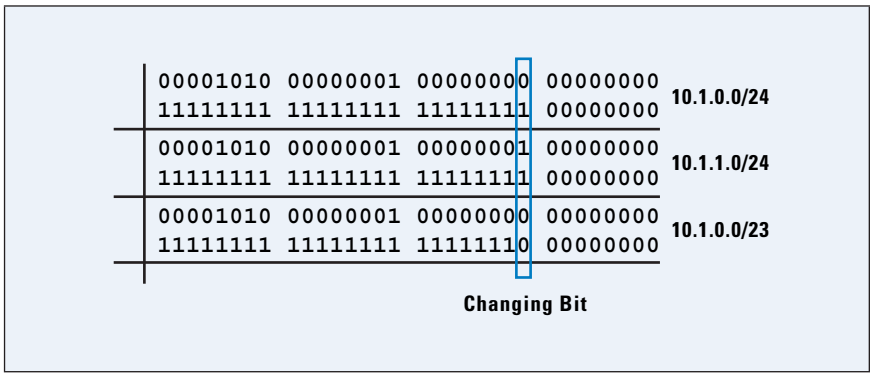
The figure shows four hosts with the addresses **10.1.0.1**, **10.1.0.2**, **10.1.0.3**, and **10.1.0.4**. Router A advertises **10.1.1.0/24**, meaning: "Any host within the address range **10.1.0.0** through **10.1.0.255** is reachable through me." Note that not all the hosts within this range exist, and that is okay—if a host within that range of addresses is reachable, it is reachable through Router A. In IP, the address that A is advertising is called a *network address*, and you can conveniently think of it as an address for the wire the hosts and router are attached to, rather than a specific device.

For many people, the confusing part comes next. Router B is also advertising **10.1.1.0/24**, which is another network address. Router C can combine—or aggregate—these two advertisements into a single advertisement. Although we have just removed the correspondence between the wire and the network address, we have not changed the fundamental meaning of the advertisement itself. In other words, Router C is saying: "Any host within the range of addresses from **10.1.0.0** through **10.1.1.255** is reachable through me." There is no wire with this address space, but devices beyond Router C do not know this, so it does not matter.

To better handle aggregated address space, we define two new terms, *subnets* and *supernets*. A subnet is a network that is contained entirely within another network; a supernet is a network that entirely contains another network. For instance, **10.1.0.0/24** and **10.1.1.0/24** are both subnets of **10.1.0.0/23**, whereas **10.1.0.0/23** is a supernet of **10.1.0.0/24** and **10.1.1.0/24**.

Now we consider a binary representation of these three addresses, and try to make more sense out of the concept of aggregation from an addressing perspective; Figure 7 illustrates.

Figure 7: Aggregation Details



By looking at the binary form of **10.1.0.0/24** and **10.1.1.0/24**, we can see that only the 24th bit in the network address changes. If we change the prefix length to 23, we have effectively "masked out" this single bit, making the **10.1.0.0/23** address cover the same address range as the **10.1.0.0/24** and **10.1.1.0/24** addresses combined.

The Hardest Subnetting Problem

The hardest subnetting problem most people face is that of trying to decide what the smallest subnet is that will provide a given number of hosts on a specific segment, and yet not waste any address space. The way this sort of problem is normally phrased is something like the following:

You have 5 subnets with the following numbers of hosts on them: 58, 14, 29, 49, and 3, and you are given the address space **10.1.1.0/24**. Determine how you could divide the address space given into subnets so these hosts fit into it.

This appears to be a very difficult problem to solve, but the chart we used previously to find the jump within a single octet actually makes this task quite easy. First, we run through the steps, and then we solve the example problem to see how it actually works.

- Order the networks from the largest to the smallest.
- Find the smallest number in the chart that fits the number of the largest number of hosts + 2 (*you cannot, except on point-to-point links, use the address with all 0's or all 1's in the host address; for point-to-point links, you can use a /31, which has no broadcast addresses*).
- Continue through each space needed until you either run out of space or you finish.

This process seems pretty simple, but does it work? Let's try it with our example.

- Reorder the numbers 58, 14, 29, 49, 3 to 58, 49, 29, 14, 3.
- Start with 58.
  - The smallest number larger than (58 + 2) is 64, and 64 is 2 bits.
  - There are 24 bits of prefix length in the address space given; add 2 for 26.
  - The first network is **10.1.1.0/26**.
  - The next network is **10.1.1.0** + 64, so we start the next "round" at **10.1.1.64**.
- The next block is 49 hosts.
  - The smallest number larger than (49 + 2) is 64, and 64 is 2 bits.
  - There are 24 bits of prefix length in the address space given; add 2 for 26.
  - We start this block at **10.1.1.64**, so the network is **10.1.1.64/26**.
  - The next network is **10.1.1.64** + 64, so we start the next "round" at **10.1.1.128**.
- The next block is 29 hosts.
  - The smallest number larger than (29 + 2) is 32, and 32 is 3 bits.
  - There are 24 bits of prefix length in the address space given; add 3 for 27.
  - We start this block at **10.1.1.128**, so the network is **10.1.1.128/27**.

  - The next network is **`10.1.1.128`** + 32, so we start the next "round" at **`10.1.1.160`**.

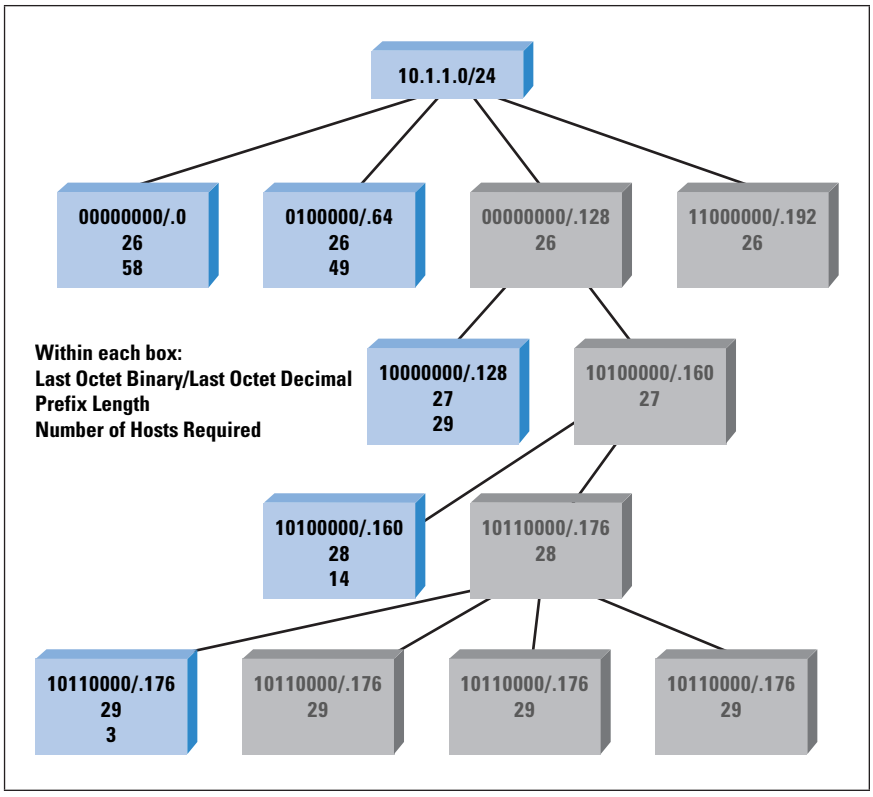
- The next block is 14 hosts.
  - The smallest number larger than (14 + 2) is 16, and 16 is 4 bits (actually equal, but it still works).
  - There are 24 bits of prefix length in the address space given; add 4 for 28.
  - We start this block at **`10.1.1.160`**, so the network is **`10.1.1.160/28`**.
  - The next network is **`10.1.1.160`** + 16, so we start the next "round" at **`10.1.1.176`**.

The last block is 3 hosts.

  - The smallest number larger than (3 + 2) is 8, and 8 is 5 bits.
  - There are 24 bits of prefix length in the address space given; add 5 for 29.
  - We start this block at **`10.1.1.176`**, so the network is **`10.1.1.176/29`**.
  - This is the last block of hosts, so we are finished.

It is a simple matter of iterating from the largest to the smallest block, and using the simple chart we used before to determine how large of a *jump* we need to cover the host addresses we need to fit onto the subnet. Figure 8 illustrates the resulting hierarchy of subnets.

Figure 8: Subnet Chart



In this illustration:

- The first line in each box contains the final octet of the network address in binary and decimal forms.
- The second line in each box contains the prefix length.
- The third line indicates the number of hosts the original problem required on that subnet.
- Gray boxes indicate blocks of address space that are unused at that level.

Working with IPv6 Addresses

IPv6 addresses appear to be much more difficult to work with—but they really are not. Although they are larger, they are still made up of the same fundamental components, and hosts and routers still use the addresses the same way. All we really need to do is realize that each pair of hexadecimal numbers in the IPv6 address is actually an octet of binary address space. The chart, the mechanisms used to find the network and host addresses, and the concepts of super and subnets remain the same.

For example, suppose we have the IPv6 address **`2002:FF10:9876:DD0A:9090:4896:AC56:0E01/63`** and we want to know what the network number is (host numbers are less useful in IPv6 networks, because they are often the MAC address of the system itself).

- 63 ÷ 8 = 7, remainder 7.
- The working octet is the 8th, which is 0A.
- Remainder 7 on the chart says the jump is 2, so the networks are **`00`**, **`02`**, **`04`**, **`06`**, **`08`**, **`0A`**, **`0C`**, and **`0E`**.
- The network is **`2002:FF10:9876:DD0A::/63`**.

The numbers are longer, but the principle is the same, as long as you remember that every *pair* of digits in the IPv6 address is a single octet.

Summary

IP addresses appear to be very complex on first approach, but their inbuilt structure actually provides easy ways to divide the problems into pieces and approach one piece of the problem at a time—the same way we design and build networks on a large scale. If you learn to use some simple techniques and understand how IP addresses are structured, they are relatively easy to work with.

For Further Reading

The following IETF *Requests for Comments* (RFCs) provide information on IP addressed and addressing structures:

[1] V. Fuller, T. Li, J. Yu, K. Varadhan, "Supernetting: an Address Assignment and Aggregation Strategy," RFC 1338, June 1992.

[2] E. Gerich, "Guidelines for Management of IP Address Space," RFC 1466, May 1993.

[3] Y. Rekhter, T. Li, "An Architecture for IP Address Allocation with CIDR," RFC 1518, September 1993.

[4] V. Fuller, T. Li, J. Yu, K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," RFC 1519, September 1993.

[5] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, "Address Allocation for Private Internets," RFC 1918, February 1996.

RUSS WHITE works for Cisco Systems in the Routing Protocols Deployment and Architecture (DNA) team in Research Triangle Park, North Carolina. He has worked in the Cisco Technical Assistance Center (TAC) and Escalation Team in the past, has co-authored several books on routing protocols, including *Advanced IP Network Design, IS–IS for IP Networks,* and co-author of *Practical BGP.* He is the co-chair of the Routing Protocols Security Working Group within the IETF. E-mail: **riw@cisco.com**

Letter to the Editor

Dear Editor,

I read with interest the article by Dave Crocker in the December 2005 issue of IPJ (Volume 8, No. 4) titled "Challenges in Anti-Spam Efforts." However, I was surprised not to find any mention of *graylisting*, an effective anti-spam technique. The technique is not a programmatic or analytical approach to the spam problem but rather relies on exploiting the general behavioral weakness of spam delivery (that spammers typically want to try an address just once for their delivery of spam contents). The technique provides a pragmatic solution to the contemporary bulk commercial e-mail problem to a large extent.

If you are planning for a sequel of this article, I would strongly advocate mentioning the technique for the general benefit of the community.

I administrate a national ISP of considerable size in Pakistan, and the extent to which graylisting has helped us in fighting against spam is amazing.

Successful spam-fighting techniques (especially those that are still far from being widely adopted and worked upon) of today make good candidates for future efforts. My enthusiasm for graylisting is chiefly a result of the benefits our company has derived from its use, but I also want to champion its use because I think it is not widely adopted among peer ISPs because of ignorance. Hence my enthusiastic advocacy of this unsung hero in the fight against spam.

Citations:
Graylisting, **http://en.wikipedia.org/wiki/Graylisting**

—Tee Emm, Supernet, Pakistan
tm@super.net.pk

The Author responds:

Dear Editor,

I appreciate Tee Emm's concern that graylisting was not explicitly cited in my article.

I must use the cliché of "limited space" as my primary excuse for omitting graylistng. The tight constraints for a brief article required some difficult choices. As I mentioned at the end of the article, the people reviewing it before publication were particularly helpful (and vigorous). The question of what detail to include was a major focus. My decision was to have only a basic review of existing techniques, because the focus of the article was on future activities.

I believe the work on detection and reaction mechanisms against "bad actors" is reasonably mature. I also believe that the creation of a trust overlay for e-mail, to permit better handling of messages from "good actors," is very early and in need of much more focus.

With that said, I think I can also generate a plausible claim that graylisting is a form of "traffic shaping," which is cited in the article.

I primarily meant the traffic shaping reference to be about the technique of tracking aggregate (statistical) flow from a specific address. However, I think that graylisting constitutes a simple—albeit quite useful—mechanism that is designed to slow down the senders, to limit their impact. As Tee Emm notes, graylisting is based on the spammers' pattern of giving up, after a single failure to send the message. That is the ultimate "shaping," I think.

Certainly a summary of existing techniques is a worthy topic. It has become quite a rich topic, and matured to a level of qualifying as an area of administration and operations specialization.

As for a follow-up article, I do not have one planned, currently. If I do another one, I hope it will be about open mechanisms for achieving authentication and assessment (vetting) of good actors.

Perhaps next year.

For reference, I should note that there has been some public follow-up on the article, when CircleID reprinted a posting I made about it: **http://www.circleid.com/posts/challenges_in_anti_spam_efforts/**

Again, I appreciate Tee Emm's interest and comment.

—Dave Crocker, Brandenburg InternetWorking
dcrocker@bbiw.net

Fragments

IETF @ 20

The *Internet Engineering Task Force* (IETF) and the *Internet Society* (ISOC) celebrate the 20th anniversary of the IETF, the world's leading Internet standards development body. The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. Its principal task today is the development and publication of technical specifications and standards for basic Internet protocols. It is open to any interested individual.

The first IETF meeting was held on the afternoon of January 16, 1986, in San Diego, California. As a community-driven activity the IETF went on to pioneer a unique, open process for standards development. Open to all, and based on principles such as "rough consensus and running code," the IETF has enabled the development of standards that have supported every aspect of the Internet's phenomenal growth.

"The IETF is unique," said Brian Carpenter, IETF Chair. "Unlike other standards bodies, there is very little in the way of formal hierarchy and there are no membership requirements or fees. The IETF welcomes broad participation by anyone interested in the future technical evolution and stability of the Internet—and IETF standards are available to all, without charge."

"There is global recognition of the achievements of the IETF in its support of the development of Internet technology. As the demands on the Internet increase, the IETF clearly has a vital role to play in ensuring that Internet technologies continue to evolve in a coherent and coordinated manner," said Leslie Daigle, chair of the *Internet Architecture Board* (IAB) which provides architectural oversight of IETF activities."

"The success of the IETF has largely been due to a pragmatic, consensus-based approach to technology standards development," noted Lynn St. Amour, President and CEO of ISOC. "Many of the principles of co-operation and collaboration that were developed in the IETF are now being successfully applied in other global forums. ISOC is proud to be associated with the IETF—we value its members' accomplishments over the last 20 years and look forward to celebrating these achievements over the course of 2006."

ISOC has declared 2006 "The Year of the IETF" and will be running several activities during the year in celebration of the IETF's 20th anniversary. For more information, see: **`http://ietf20.isoc.org`**

The Internet Society is a not-for-profit membership organization founded in 1992 to provide leadership in Internet related standards, education, and policy. With offices in Washington, DC, and Geneva, Switzerland, it is dedicated to ensuring the open development, evolution and use of the Internet for the benefit of people throughout the world. ISOC is the organizational home of the IETF and other Internet-related bodies who together play a critical role in ensuring that the Internet develops in a stable and open manner. For over 13 years ISOC has run international network training programs for developing countries and these have played a vital role in setting up the Internet connections and networks in virtually every country connecting to the Internet during this time.

ISOC Welcomes WSIS Proposal

Delegates meeting at the *World Summit on the Information Society* (WSIS) in Tunis have affirmed their commitment to build on the governance mechanisms that have enabled the Internet's incredibly successful growth.

ISOC welcomes the recognition by WSIS of how the effectiveness of the existing arrangements for Internet governance has helped make the Internet the highly robust, dynamic and geographically diverse medium that it is today.

"We are delighted that there is now much broader recognition of the achievements of the organisations that support the Internet community," said Lynn St. Amour, President and CEO of the ISOC. "These organizations, along with their open, consensus-based processes clearly have a vital role to play in the further development of the Internet. It is also significant that the WSIS debate has moved beyond the details of technical administration and on to broader issues that require increased coordination by stakeholders in order to ensure the continued stability of the Internet."

The WSIS recommendation includes a proposal for a new forum for multi-stakeholder policy dialogue—the *Internet Governance Forum*. ISOC, together with partner organizations from the Internet community, has always worked to encourage full engagement in such dialogues by all those with an interest in the Internet's future. ISOC believes that the forum's success depends upon the fullest participation by all stakeholders. At the same time, ISOC is pleased to note that the proposed forum would have no oversight function and would have no involvement in the day-to-day operations of the Internet.

"ISOC will facilitate increased cooperation and information sharing amongst all parties interested in Internet governance and we look forward to playing an active role in the new forum as is expected of us by the global community," said Lynn St. Amour. "We very much hope that the Tunis summit will lead to some real and positive outcomes that will help bring the benefits of the Internet to people everywhere—especially to those who are yet to be connected."

ISOC, along with some of its partner organisations—the *Number Resource Organisation* (NRO), the IETF, *London Internet Exchange* (LINX), the *Internet Corporation for Assigned Names and Numbers* (ICANN) and the *Council of European National Top level domain Registries* (CENTR)—were present at the *ICT 4 All* exhibition held in conjunction with WSIS.

For more information about the organizations listed above visit:

http://isoc.org

http://ietf.org

http://iab.org

http://www.intgovforum.org

http://www.linx.net

http://nro.org

http://www.centr.org

http://www.itu.int/wsis

Upcoming Events

The *Internet Engineering Task Force* (IETF) will meet in Montreal, Canada, July 9–14, 2006. For more information, visit:
http://ietf.org

ACM's *SIGCOMM 2006* will be held in Pisa, Italy, September 11–15, 2006. For more information, visit:
http://www.acm.org/sigs/sigcomm/sigcomm2006

The *North American Network Operators Group* (NANOG) will meet in St. Louis, MO October 8–10, 2006. For more information, see: **http://nanog.org**

The *American Registry for Internet Numbers* (ARIN) will meet (jointly with NANOG) in St. Louis, October 11–13, 2006. For more information, see: **http://arin.net**

The Internet Protocol Journal
Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA