LECTURE 4
Interdomain Internet Routing

The goal of this lecture is to explain how routing between different administrative domains works in the Internet. We discuss how Internet Service Providers (ISPs) exchange routing information (and packets) between each other, and how the way in which they buy service from and sell service to each other and their customers influences the technical research agenda of Internet routing in the real-world. We discuss the salient features of the Border Gateway Protocol, Version 4 (BGP4), the current interdomain routing protocol in the Internet.

■ 4.1 Autonomous Systems

An abstract, highly idealized view of the Internet is shown in Figure 4-1, where end-hosts hook up to routers, which hook up with other routers to form a nice connected graph of essentially "peer" routers that cooperate nicely using routing protocols that exchange "shortest-path" or similar information and provide global connectivity. The same view posits that the graph induced by the routers and their links has a large amount of redundancy and the Internet's routing algorithms are designed to rapidly detect faults and problems in the routing substrate and route around them. Some would even posit that the same routing protocols today perform load-sensitive routing to dynamically shed load away from congested paths on to less-loaded paths.

Unfortunately, while simple, this abstraction is actually quite misleading. The real story of the Internet routing infrastructure is that the Internet service is provided by a large number of commercial enterprises, generally in competition with each other. Cooperation, required for global connectivity, is generally at odds with the need to be a profitable commercial enterprise, which often occurs at the expense of one's competitors—the same people with whom one needs to cooperate. How this is achieved in practice (although there's lots of room for improvement), and how we might improve things, is an interesting and revealing study of how good technical research can be shaped and challenged by commercial realities.

A second pass at developing a good picture of the Internet routing substrate is shown in Figure 4-2, which depicts a group of Internet Service Providers (ISPs) somehow cooper-

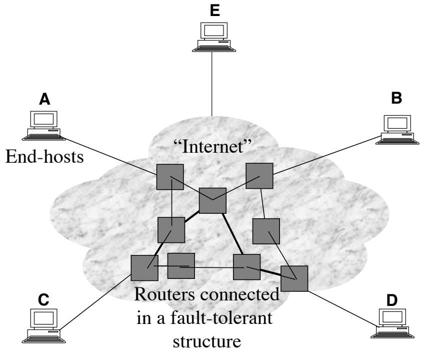


Figure 4-1: This is a rather misleading abstraction of the Internet routing layer.

ating to provide global connectivity to end-customers. This picture is closer to the truth, but the main thing it's missing is that not all ISPs are created equal. Some are bigger and more "connected" than others, and still others have global reachability in their routing tables. There are names given to these "small," "large," and "really huge" ISPs: *Tier-3 ISPs* are ones that have a small number of usually localized (in geography) end-customers; *Tier-2 ISPs* generally have regional scope (*e.g.,* state-wide, region-wide, or non-US country-wide), while *Tier-1 ISPs*, of which there are a handful, have global scope in the sense that their routing tables actually have routes to all currently reachable Internet prefixes (*i.e.,* they have no default routes). This organization is shown in Figure 4-3.

The current wide-area routing protocol, which exchanges *reachability information* about routeable IP-address prefixes between routers at the boundary between ISPs, is *BGP* (Border Gateway Protocol, Version 4) [13, 14]. More precisely, the wide-area routing architecture is divided into *autonomous systems* (ASes) that exchange reachability information. An AS is owned and administered by a single commercial entity, and implements some set of policies in deciding how to route its packets to the rest of the Internet, and how to export its routes (its own, those of its customers, and other routes it may have learned from other ASes) to other ASes. Each AS is identified by a unique 16-bit number.

A different routing protocol operates within each AS. These routing protocols are called *Interior Gateway Protocols* (IGPs), and include protocols like Routing Information Protocol (RIP) [8]. Open Shortest Paths First (OSPF) [11], Intermediate System-Intermediate System (IS-IS) [12], and E-IGRP. In contrast, interdomain protocols like BGP are also called EGPs (Exterior Gateway Protocols). Operationally, a key difference between EGPs like BGP and IGPs is that the former is concerned with providing *reachability information* and facilitating *routing policy* implementation in a *scalable* manner, whereas the latter are typically concerned with optimizing a path metric. Scalability is typically not a major concern in the design of IGPs (and all known IGPs don't scale as well as BGP does).

The rest of this lecture is in two parts: first, we will look at inter-AS relationships (transit and peering); then, we will study some salient features of BGP. We don't have time to

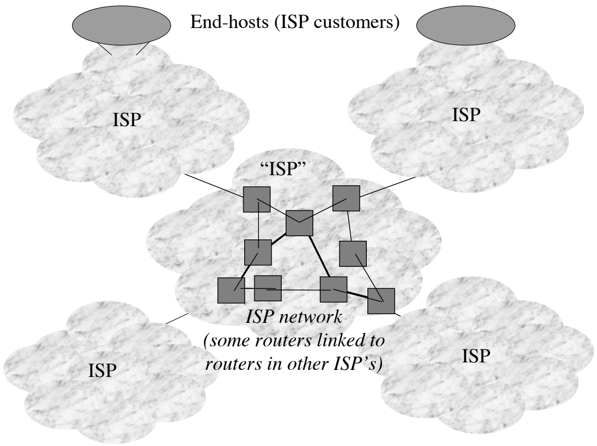


Figure 4-2: The Internet is actually composed of many competing Internet Service Providers (ISPs) that cooperate to provide global connectivity. This picture suggests that all ISPs are "equal," which isn't actually true.

survey IGPs in this lecture, but you should be familiar with the more well-known ones like RIP and OSPF (or at least with distance-vector and link-state protocols). To learn more about IGPs if you're not familiar with them, read a standard networking textbook (*e.g.*, Peterson & Davie, Kurose & Ross, Tanenbaum) or a book on routing protocols (*e.g.*, Huitema).

■ 4.2 Inter-AS Relationships: Transit and Peering

The Internet is composed of many different types of ASes, from universities to corporations to regional Internet Service Providers (ISPs) to nationwide ISPs. Smaller ASes (*e.g.*, universities, corporations, etc.) typically purchase Internet connectivity from ISPs. Smaller regional ISPs, in turn, purchase connectivity from larger ISPs with "backbone" networks.

Consider the picture shown in Figure 4-4. It shows an ISP, with AS number X, directly connected to a *provider* (from whom it buys Internet service) and a few *customers* (to whom it sells Internet service). In addition, the figure shows two other ISPs to whom it is directly connected, with whom X exchanges routing information via BGP.

The different types of ASes lead to different types of business relationships between them, which in turn translate to different policies for exchanging and selecting routes. There are two prevalent forms of AS-AS interconnection. The first form is *provider-customer transit* (aka "transit"), wherein one ISP (the "provider" P in Figure 4-4) provides access to all (or most) destinations in its routing tables. Transit almost always is meaningful in an inter-AS relationship where financial settlement is involved; the provider charges its customers for Internet access, in return for forwarding packets on behalf of customers to destinations (and in the opposite direction in many cases). Another example of a transit relationship in Figure 4-4 is between X and its customers (the C_is).

The second prevalent form is called *peering*. Here, two ASes (typically ISPs) provide

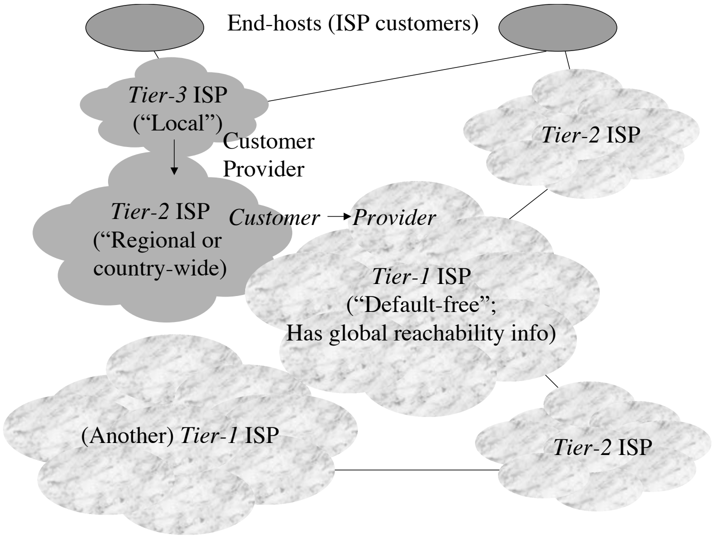

End-hosts (ISP customers)
Tier-3 ISP ("Local")
Customer
Provider
Tier-2 ISP
Tier-2 ISP ("Regional or country-wide)
Customer → *Provider*
Tier-1 ISP ("Default-free"; Has global reachability info)
(Another) *Tier-1* ISP
Tier-2 ISP

Figure 4-3: A more accurate picture of the wide-area Internet routing infrastructure, with various types of ISPs defined by their respective reach. *Tier-1* ISPs have "default-free" routing tables (i.e., they don't have any default routes), and typically have global reachability information. There are a handful of these today (about five or so).

mutual access to a subset of each others' routing tables. The subset of interest here is their own transit customers (and the ISPs own internal addresses). Like transit, peering is a business deal, but it may not involve financial settlement. While paid peering is common in some parts of the world, in many cases they are reciprocal agreements. As long as the traffic ratio between the concerned ASs is not highly asymmetric (*e.g.,* 4:1 is a commonly believed and quoted ratio), there's usually no financial settlement. Peering deals are almost always under non-disclosure and are confidential.

■ 4.2.1 Peering v. Transit

A key point to note about peering relationships is that they are often between business competitors. The common reason for peering is the observation by each party that a non-trivial fraction of the packets emanating from each one is destined for the other's direct transit customers. Of course, the best thing for each of the ISPs to try to do would be to wean away the other's customers, but that may be hard to do. The next best thing, which would be in their mutual interest, would be to avoid paying transit costs to *their* respective providers, but instead set up a transit-free link between each other to forward packets for their direct customers. In addition, this approach has the advantage that this more direct path would lead to better end-to-end performance (in terms of latency, packet loss rate, and throughput) for their customers. It's also worth noticing that a Tier-1 ISP usually will find it essential to be involved in peering relationships with other ISPs (especially other Tier-1 ISPs) to obtain global routing information in a default-free manner.

Balancing these potential benefits are some forces against peering. Transit relationships generate revenue; peering relationships usually don't. Peering relationships typically need

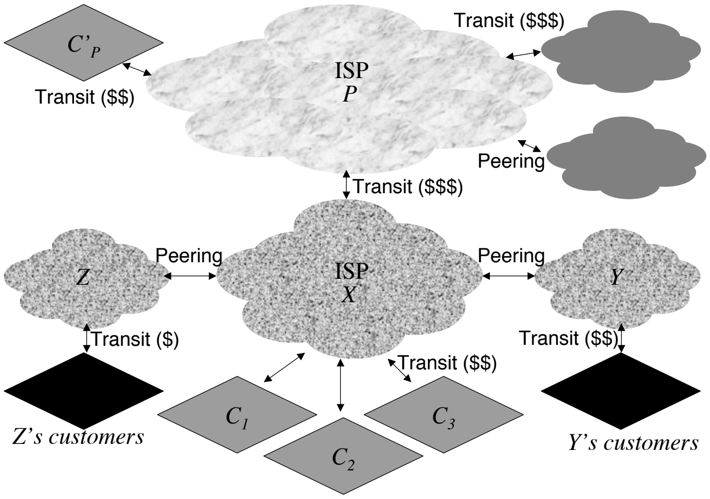


Figure 4-4: Inter-AS relationships; transit and peering.

to be renegotiated often, and asymmetric traffic ratios require care to handle in a way that's mutually satisfactory. Above all, these relationships are often between competitors vying for the same customer base.

In the discussion so far, we have implicitly used an important property of current interdomain routing: *A route advertisement from B to A for a destination prefix is an agreement by B that it will forward packets sent via A destined for any destination in the prefix.* This (implicit) agreement implies that one way to think about Internet economics is to view ISPs as charging customers for entries in their routing tables. Of course, the data rate of the interconnection is also crucial, and is the major determinant of an ISP's pricing policy.

■ 4.2.2 Exporting Routes: Route Filtering

Each AS (ISP) needs to make decisions on which routes to export to its neighboring ISPs using BGP. The reason why export policies are important is that no ISP wants to act as transit for packets that it isn't somehow making money on. Because packets flow in the opposite direction to the (best) route advertisement for any destination, an AS should advertise routes to neighbors with care.

Transit customer routes. To an ISP, its customer routes are likely the most important, because the view it provides to its customers is the sense that *all* potential senders in the Internet can reach them. It is in the ISP's best interest to advertise routes to its transit customers to as many other connected ASes as possible. The more traffic that an ISP carries on behalf of a customer, the "fatter" the pipe that the customer would need, implying higher revenue for the ISP. Hence, if a destination were advertised from multiple neighbors, an ISP should prefer the advertisement made from a customer over all other choices (in particular, over peers and transit providers).

Transit provider routes. Does an ISP want to provide *transit* to the routes exported by its provider to it? Most likely not, because the ISP isn't making any money on providing such transit facilities. An example of this situation is shown in Figure 4-4, where C'_P is a customer of P, and P has exported a route to C'_P to X. It isn't in X's interest to advertise this route to everyone, *e.g.*, to other ISPs with whom X has a peering relationship. An important exception to this, of course, is X's transit customers who are paying X for service—the service X provides its customers C_i's is that they can reach any location on the Internet via X, so it makes sense for X to export as many routes to X as possible.

Peer routes. It usually makes sense for an ISP to export only selected routes from its routing tables to other peering ISPs. It obviously makes sense to export routes to all of ones transit customers. It also makes sense to export routes to addresses within an ISP. However, it does not make sense to export an ISP's transit provider routes to other peering ISPs, because that may cause a peering ISP to use the advertising ISP to reach a destination advertised by a transit provider. Doing so would expend ISP resources but not lead to revenue.

The same situation applies to routes learned from other peering relationships. Consider ISP Z in Figure 4-4, with its own transit customers. It doesn't make sense for X to advertise routes to Z's customers to another peering ISP (Y), because X doesn't make any money on Y using X to get packets to Z's customers!

These arguments show that most ISPs end up providing *selective transit*: typically, full transit capabilities for their own transit customers in both directions, some transit (between mutual customers) in a peering relationship, and transit only for one's transit customers (and ISP-internal addresses) to one's providers.

The discussion so far may make it sound like BGP is the only way in which to exchange reachability information between an ISP and its customers or between two ASes. That is not true—a large fraction of end-customers (typically customers who don't provide large amounts of further transit and/or aren't ISPs) do not run BGP sessions with their providers. The reason is that BGP is complicated to configure, administer, and manage, and isn't particularly useful if the set of addresses in the customer is relatively invariant. These customers interact with their providers via *static routes*. These routes are usually manually configured. Of course, information about customer address blocks will in general be exchanged by a provider using BGP to other ASes (ISPs) to achieve global reachability to the customer premises.

■ 4.2.3 Importing Routes

The previous section described the issues considered by an AS (specifically, routers in an AS involved in BGP sessions with routers in other ASes) while deciding which routes to export. In a similar manner, when a router hears many possible routes to a destination network, it needs to decide which route to install in its forwarding tables.

This is a fairly involved process in BGP and requires a consideration of several attributes of the advertised routes. At this stage, we consider only one of the many things that a router needs to consider, but it's the most important consideration. It has to do with who advertised the route. Typically, when a router (*e.g.*, X in Figure 4-4) hears advertisements to its transit customers from other ASes (*e.g.*, because the customer is multi-homed), it needs to ensure that packets to the customer do not traverse additional ASes unnecessarily. This usually means that customer routes are prioritized over routes to the same network advertised by providers or peers. Second, peer routes are likely more preferable to provider routes, since the purpose of peering was to exchange reachability information about mutual transit customers. These two observations imply that typically routes are imported in the following priority order:

$$\textit{customer} > \textit{peer} > \textit{provider}$$

This rule (and many others like it) can be implemented in BGP using a special attribute that's locally maintained by routers in an AS, called the LOCAL PREF attribute. The first rule in route selection with BGP is to pick a route based on this attribute. It is only if this attribute is *not* set for a route, are other attributes of a route even considered. Note, however, that in practice most routes in most ASes are not selected using the LOCAL PREF attribute; other attributes like the length of the AS path tend to be quite common. We discuss these other route attributes and the details of the BGP route selection process, also called the *decision process*, in the next section.

■ 4.3 BGP

We now turn to how reachability information is exchanged using BGP, and how routing policies like the ones explained in the previous section can be expressed and enforced. We start with a discussion of the main design goals in BGP and summarize the protocol. Most of the complexity in wide-area routing is not in the protocol, but in how BGP routers are configured to implement policy, and in how routes learned from other ASes are disseminated within an AS. The rest of the section discusses these issues.

■ 4.3.1 Design Goals

The design of BGP, and its current version (4), was motivated by three important needs:

1. **Scalability.** The division of the Internet into ASes under independent administration was done while the backbone of the then Internet was under the administration of the NSFNet. An important requirement for BGP was to ensure that the Internet routing infrastructure remained scalable as the number of connected networks increased.

2. **Policy.** The ability for each AS to implement and enforce various forms of routing policy was an important design goal. One of the consequences of this was the development of the BGP attribute structure for route announcements, and allowing route filtering.

3. **Cooperation under competitive circumstances.** BGP was designed in large part to handle the transition from the NSFNet to a situation where the "backbone" Internet infrastructure would no longer be run by a single administrative entity. This structure implies that the routing protocol should allow ASes to make purely local decisions on how to route packets, from among any set of choices.

In the old NSFNET, the backbone routers exchanged routing information over a tree topology, using a routing protocol called EGP. (While the modern use of the term EGP is as a family of exterior gateway protocols, its use in the context of NSFNET refers to the specific one used in that network.) Because the backbone routing information was exchanged over a tree, the routing protocol was relatively simple. The evolution of the Internet from a singly administered backbone to its current commercial structure made the NSFNET EGP obsolete and required a more sophisticated protocol.

■ 4.3.2 The Protocol

As protocols go, the operation of BGP is quite straightforward. The basic operation of BGP—the protocol state machine, the format of routing messages, and the propagation of routing updates—are all defined in the protocol standard [13]. BGP runs over TCP, on a well-known port (179). To start participating in a *BGP session* with another router, a router sends an OPEN message after establishing a TCP connection to it on the BGP port. After the OPEN is completed, both routers exchange their tables of all active routes (of course, applying all applicable route filtering rules). This process may take several minutes to complete, especially on sessions that have a large number of active routes.

After this initialization, there are two main types of messages on the BGP session. First, BGP routers send route UPDATE messages sent on the session. These updates only send any routing entries that have changed since the last update (or transmission of all active routes). There are two kinds of updates: *announcements*, which are changes to existing routes or new routes, and *withdrawals*, which are messages that inform the receiver that the named routes no longer exist. A withdrawal usually happens when some previously announced route can no longer be used (e.g., because of a failure or a change in policy). Because BGP uses TCP, which provides reliable and in-order delivery, routes do not need to be periodically announced, unless they change.

But, in the absence of periodic routing updates, how does a router know whether the neighbor at the other end of a session is still functioning properly? One possible solution might be for BGP to run over a transport protocol that implements its own "is the peer alive" message protocol. Such messages are also called "keepalive" messages. TCP, however, does not implement a transport-layer "keepalive", so BGP uses its own. Each BGP session has a configurable keepalive timer, and the router guarantees that it will attempt to send at least one BGP message during that time. If there are no UPDATE messages, then the router sends the second type of message on the session: KEEPALIVE messages. The absence of a certain number BGP KEEPALIVE messages on a session causes the router to terminate that session. The number of missing messages depends on a configurable times called the *hold timer*; the specification recommends that the hold timer be at least as long as the keepalive timer duration negotiated on the session.

More details about the BGP state machine may be found in [2, 13].

Unlike many IGP's, BGP does not simply optimize any metrics like shortest-paths or delays. Because its goals are to provide reachability information and enable routing policies, its announcements do not simply announce some metric like hop-count. Rather, they have the following format:

$$IP\ prefix : Attributes$$

where for each announced IP prefix, one or more attributes are also announced. There are a substantial number of standardized attributes in BGP, and we'll look at some of them in

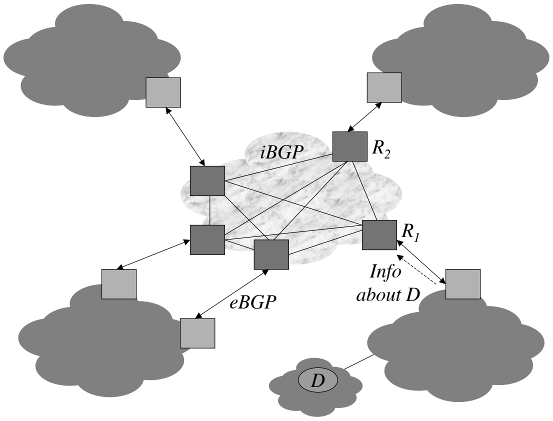


Figure 4-5: eBGP and iBGP.

more detail in the rest of this lecture.

Recall that one BGP attribute has already been introduced, the LOCAL PREF attribute. This attribute isn't disseminated with route announcements, but is an important attribute used locally while selecting a route for a destination. When a route is advertised from a neighboring AS, the receiving BGP router consults its configuration and may set a LOCAL PREF for this route.

■ 4.3.3 Disseminating Routes within an AS: eBGP and iBGP

There are two types of BGP sessions: *eBGP* sessions are between BGP-speaking routers in different ASes, while *iBGP* sessions are between BGP routers in the same AS. They serve different purposes, but use exactly the same protocol.

eBGP is the "standard" mode in which BGP is used; after all BGP was designed to exchange network routing information between different ASes in the Internet. eBGP sessions are shown in Figure 4-5, where the BGP routers implement route filtering rules and exchange a subset of their routes with routers in other ASes.

In general, each AS will have more than one router that participates in eBGP sessions with neighboring ASes. During this process, each router will obtain information about some subset of all the prefixes that the entire AS knows about. Each such eBGP router must disseminate routes to the external prefix to all the other routers in the AS. This dissemination must be done with care to meet two important goals:

1. *Loop-free forwarding.* After the dissemination of eBGP learned routes, the resulting routes (and the subsequent forwarding paths of packets sent along those routes) picked by all routers should be free of deflections and forwarding loops [4, 7].

2. *Complete visibility.* One of the goals of BGP is to allow each AS to be treated as a single monolithic entity. This means that the several eBGP-speaking routes in the AS must exchange external route information so that they have a complete view of all external routes. For instance, consider Figure 4-5, and prefix D. Router R_2 needs to know how

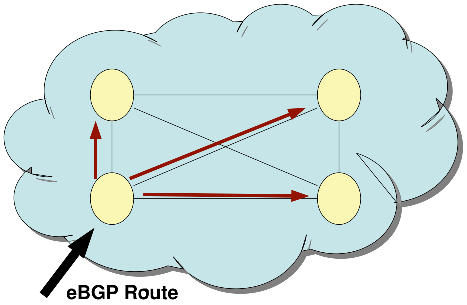


Figure 4-6: Small ASes establish a "full mesh" of iBGP sessions. Each circle represents a router within an AS. Only eBGP-learned routes are re-advertised over iBGP sessions.

> to forward packets destined for D, but R_2 hasn't heard a direct announcement on any of its eBGP sessions for D.[1] By "complete visibility", we mean the following: *for every external destination, each router picks the same route that it would have picked had it seen the best routes from each eBGP router in the AS.*
>
> The dissemination of externally learned routes to routers inside an AS is done over *internal BGP* (iBGP) sessions running in each AS.

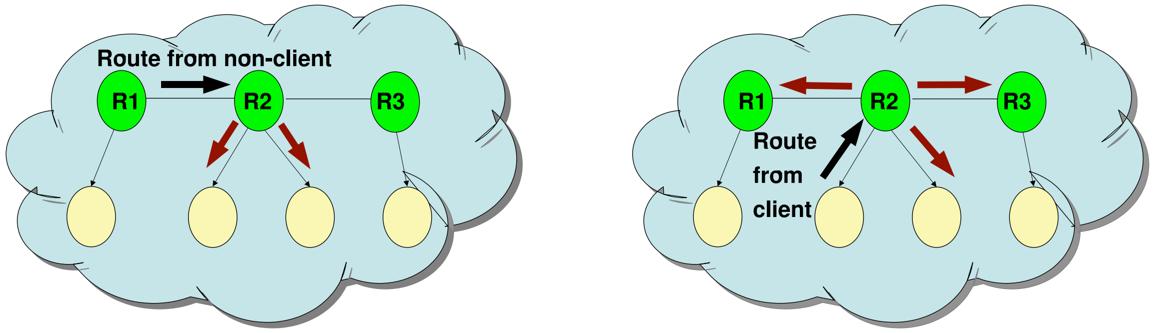
An important question concerns the topology over which iBGP sessions should be run. One possibility is to use an arbitrary connected graph and "flood" updates of external routes to all BGP routers in an AS. Of cours, an approach based on flooding would require additional techniques to avoid routing loops. The original BGP specification solved this problem by simply setting up a *full mesh* of iBGP sessions (see Figure 4-6, where every eBGP router maintains an iBGP session with every other BGP router in the AS. Flooding updates is now straightforward; an eBGP router simply sends UPDATE messages to its iBGP neighbors. An iBGP router does not have to send any UPDATE messages because it does not have any eBGP sessions with a router in another AS.

It is important to note that *iBGP is not an IGP* like RIP or OSPF, and it cannot be used to set up routing state that allows packets to be forwarded correctly between internal nodes in an AS. Rather, iBGP sessions, running over TCP, provide a way by which routers inside an AS can use BGP to exchange information about external routes. In fact, iBGP sessions and messages are themselves routed between the BGP routers in the AS via whatever IGP is being used in the AS!

One might wonder why iBGP is needed, and why one can't simply use whatever IGP is being used in the AS to also send BGP updates. There are several reasons why introducing eBGP routes into an IGP is inconvenient. The first reason is that most IGPs don't scale as well as BGP does, and often rely on periodic routing announcements rather than incremental updates (*i.e.*, their state machines are different). Second, IGPs usually don't implement the rich set of attributes present in BGP. To preserve all the information about routes gleaned from eBGP sessions, it is best to run BGP sessions inside an AS as well.

The requirement that the iBGP routers be connected via a complete mesh limits scalability: a network with e eBGP routers and i other interior routers requires $e(e-1)/2 + ei$ iBGP

[1] It turns out that each router inside doesn't know about all the external routes to a destination. Rather, we will strive for each router being able to discover the best routes of the egress routers in the AS for a destination.



(a) Routes learned from non-clients are re-advertised to clients only.

(b) Routes learned from clients are re-advertised over all iBGP sessions.

Figure 4-7: Larger ASes commonly use route reflectors, which advertise some iBGP-learned routes, as described above. Directed edges between routers represent iBGP sessions from route reflectors to clients (*e.g.*, router *R*2 is a route reflector with two clients). As in Figure 4-6, all routers re-advertise eBGP-learned routes over all iBGP sessions.

sessions in a full-mesh configuration. While this quadratic scaling is not a problem for a small AS with only a handful of routers, large backbone networks typically have more several hundred routers, requiring tens of thousands of iBGP sessions. This quadratic scaling does not work well in those cases.

As a result, two methods to handle this have arisen, both based on manual configuration into some kind of hierarchy. The first method is to use *route reflectors* [1], while the second sets up *confederations* of BGP routers [15]. We briefly summarize the main ideas in route reflection in this lecture, and refer the interested reader to RFC 3065 [15] for a discussion of BGP confederations.

A route reflector is a BGP router that can be configured to have *client* BGP routers. A route reflector selects a single best route to each destination prefix and announces that route to all of its clients. An AS with a route reflector configuration follows the following rules in its route updates:

1. If a route reflector learns a route via eBGP or via iBGP from one of its clients, the it re-advertises that route over all of its sessions to its clients.

2. If a route reflector learns a route via iBGP from a router that is not one of its clients, then it re-advertises the route to its client routers, *but not over any other iBGP sessions*.

Having only one route reflector in an AS causes a different scaling problem, because it may have to support a large number of client sessions. More importantly, if there are multiple egress links from the AS to a destination prefix, a single route-reflector configuration may not use them all well, because all the clients would inherit the single choice made by the route reflector. To solve this problem, many networks deploy multiple route reflectors, organizing them hierarchically. Figure 4-7 shows an example route reflector hierarchy and how routes propagate from various iBGP sessions.

BGP route updates propagate differently depending on whether the update is propagating over an eBGP session or an iBGP session. An eBGP session is typically a *point-to-point* session: that is, the IP addresses of the routers on either end of the session are directly connected with one another and are typically on the same local area network. There are some exceptions to this practice (*i.e.*, "multi-hop eBGP" [5]), but directly connected eBGP sessions is normal operating procedure. In the case where an eBGP session is point-to-point, the next-hop attribute for the BGP route is guaranteed to be reachable, as is the other end of the point-to-point connection. A router will advertise a route over an eBGP session regardless of whether that route was originally learned via eBGP or iBGP.

On the other hand, an iBGP session may exist between two routers that are *not* directly connected, and it may be the case that the next-hop IP address for a route learned via iBGP is more than one IP-level hop away. In fact, as the next-hop IP address of the route is typically one of the border routers for the AS, this next hop may not even correspond to the router on the other end of the iBGP session, but may be several *iBGP* hops away. In iBGP, the routers thus rely on the AS's internal routing protocol (*i.e.*, its IGP) to both (1) establish connectivity between the two endpoints of the BGP session and (2) establish the route to the next-hop IP address named in the route attribute.

Configuring an iBGP topology to correctly achieve loop-free forwarding and complete visibility is non-trivial. Incorrect iBGP topology configuration can create many types of incorrect behavior, including persistent forwarding loops and oscillations [7]. Route reflection causes problems with correctness because not all route reflector topologies satisfy visibility (see [6] and references therein).

■ 4.3.4 BGP Policy Expression: Filters and Rankings

BGP allows policy expression by allowing network operators to configure routers to *manipulate* route attributes when disseminating routes. Network operators can configure routers to perform the following policy-driven tasks:

1. Control how a router ranks candidate routes and select paths to destinations.
2. Control the "next hop" IP address for the advertised route to balance load.
3. "Tag" a route to control how the ranking and filtering functions on other routers treat the route.

We're now in a position to understand what the anatomy of a BGP route looks like and how route announcements (and withdrawals) allow a router to compute a forwarding table from all the routing information. This forwarding table typically has one chosen path in the form of the egress interface (port) on the router, corresponding to the next neighboring IP address, to send a packet destined for a prefix. Recall that each router implements the longest prefix match on each packet's destination IP address.

■ 4.3.5 Exchanging Reachability: NEXT HOP Attribute

A BGP route announcement has a set of attributes associated with each announced prefix. One of them is the NEXT HOP attribute, which gives the IP address of the router to send the packet to. As the announcement propagates across an AS boundary, the NEXT HOP field is changed; typically, it gets changed to the IP address of the border router of the AS the announcement came from.

Route Attribute	Description
Next Hop	IP Address of the next-hop router along the path to the destination. On eBGP sessions, the next hop is set to the IP address of the border router. On iBGP sessions, the next hop is not modified.
AS path	Sequence of AS identifiers that the route advertisement has traversed.
Local Preference	This attribute is the first criteria used to select routes. It is not attached on routes learned via eBGP sessions, but typically assigned by the import policy of these sessions; preserved on iBGP sessions.
Multiple-Exit Discriminator (MED)	Used for comparing two or more routes from the same neighboring AS. That neighboring AS can set the MED values to indicate which router it prefers to receive traffic for that destination. *By default, not comparable among routes from different ASes.*

Table 4-1: Important BGP route attributes.

The above behavior is for eBGP speakers. For iBGP speakers, the first router that introduces the route into iBGP sets the NEXT HOP attribute to its so-called loopback address (the address that all other routers within the AS can use to reach the first router). All the other iBGP routers within the AS *preserve* this setting, and use the ASes IGP to route any packets destined for the route (in the reverse direction of the announcement) toward the NEXT HOP IP address. In general, packets destined for a prefix flow in the opposite direction to the route announcements for the prefix.

Length of AS Paths: ASPATH Attribute

Another attribute that changes as a route annoucement traverses different ASes is the ASPATH attribute, which is a *vector* that lists all the ASes (in reverse order) that this route announcement has been through. Upon crossing an AS boundary, the first router prepends the unique identifier of its own AS and propagates the announcement on (subject to its route filtering rules). This use of a "path vector"—a list of ASes per route—is the reason BGP is classified as a *path vector protocol*.

A path vector serves two purposes. The first is *loop avoidance*. Upon crossing an AS boundary, the router checks to see if its own AS identifier is already in the vector. If it is, then it discards the route announcement, since importing this route would simply cause a routing loop when packets are forwarded.

The second purpose of the path vector is to help pick a suitable path from among multiple choices. If no LOCAL PREF is present for a route, then the ASPATH length is used to decide on the route. Shorter ASPATH lengths are preferred to longer ones. However, it is important to remember that BGP isn't a strict shortest-ASPATH protocol (classical path

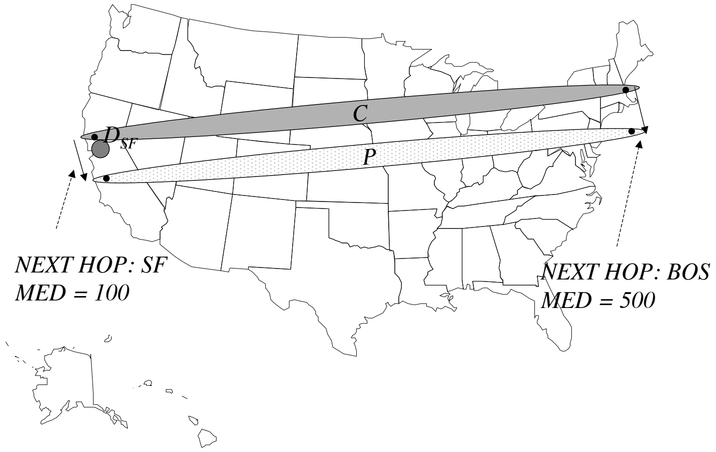


Figure 4-8: MED's **are useful in many situations,** ***e.g.*****, if** C **is a transit customer of** P**, to ensure that cross-country packets to** C **traverse** P**'s (rather than** C**'s wide-area network). However, if** C **and** P **are in a peering relationship,** MED **may (and often will) be ignored. In this example, the** MED **for** D_SF **is set to 100 at the SF exchange point, and 500 in Boston, so** P **can do the right thing if it wants to.**

vector protocols would pick shortest vectors), since it pays attention to routing policies. The LOCAL PREF attribute is always given priority over ASPATH. Many routes in practice, though, end up being picked according to shortest-ASPATH.

Choosing Between Multiple Exit Points: MED Attribute

There are many situations when two ASes are linked at multiple locations, and one of them may prefer a particular transit point over another. This situation can't be distinguished using LOCAL PREF (which decides which AS' announcement to import) or shortest ASPATH (since they would be equal). A BGP attribute called MED, for *multi-exit discriminator* is used for this.

It's best to understand MED using an example. Consider Figure 4-8 which shows a provider-customer relationship where both the provider P and customer C have national footprints. Cross-country bandwidth is a much more expensive resource than local bandwidth, and the customer would like the provider to incur the cost of cross-country transit for the customer's packets. Suppose we want to route packets from the east coast (Boston) destined for D_{SF} to traverse P's network and not C's. We want to prevent P from transiting the packet to C in Boston, which would force C to use its own resources and defeat the purpose of having P as its Internet provider.

A MED attribute allows an AS, in this case C, to tell another (P) how to choose between multiple NEXT HOP's for a prefix D_{SF}. Each router will pick the smallest MED from among multiple choices coming from the same neighbor AS. No semantics are associated with how MED values are picked, but they must obviously be picked and announced consistently amongst the eBGP routers in an AS. In our example, a MED of 100 for the *SF*

NEXT HOP for prefix D_{SF} and a MED of 500 for the BOS NEXT HOP for the same prefix accomplishes the desired goal.

An important point to realize about MED's is that they are usually ignored in AS-AS relationships that don't have some form of financial settlement (or explicit arrangement, in the absence of money). In particular, most peering arrangements ignore MED. This leads to a substantial amount of *asymmetric routes* in the wide-area Internet, as we'll see in the next lecture. For instance, if P and C were in a peering relationship in Figure 4-8, cross-country packets going from C to P would traverse P's wide-area network, while cross-country packets from P to C would traverse C's wide-area network. Both P and C would be in a hurry to get rid of the packet from their own network, a form of routing sometimes called *hot-potato routing*. In contrast, a financial arrangement would provide an incentive to honor MED's and allow "cold-potato routing" to be enforced.

The case of large content hosts peering with tier-1 ISPs is an excellent real-world example of cold-potato routing. For instance, an ISP might peer with a content-hosting provider to obtain direct access to the latter's customers (popular content-hosting sites), but does not want the hosting provider to free-load on its backbone. This can be achieved by insisting that its MEDs be honored.[2]

Putting It All Together

So far, we have seen the most important BGP attributes: LOCAL PREF, ASPATH, and MED. We are now in a position to discuss the set of rules that BGP routers in an AS use to select a route from among multiple choices.

These rules are shown in Table 4-2, in priority order. These rules are actually slightly vendor-specific; for instance, the Router ID tie-break is not the default on Cisco routers, which select the "oldest" route in the hope that this route would be the most "stable."

Priority	Rule	Remarks
1	LOCAL PREF	Highest LOCAL PREF (§4.2.3). *E.g.,* Prefer transit customer routes over peer and provider routes.
2	ASPATH	Shortest ASPATH length (§4.3.5) *Not* shortest number of Internet hops or delay.
3	MED	Lowest MED preferred (§4.3.5). May be ignored, esp. if no financial incentive involved.
4	eBGP > iBGP	Did AS learn route via eBGP (preferred) or iBGP?
5	IGP path	Lowest IGP path cost to next hop (egress router). If all else equal so far, pick shortest internal path.
6	Router ID	Smallest router ID (IP address). A random (but unchanging) choice; some implementations use a different tie-break such as the oldest route.

Table 4-2: How a BGP-speaking router selects routes. There used to be another step between steps 2 and 3 in this table, but it's not included in this table because it is now obsolete.

[2] I thank Nick Feamster for educating me about this operational use of MEDs.

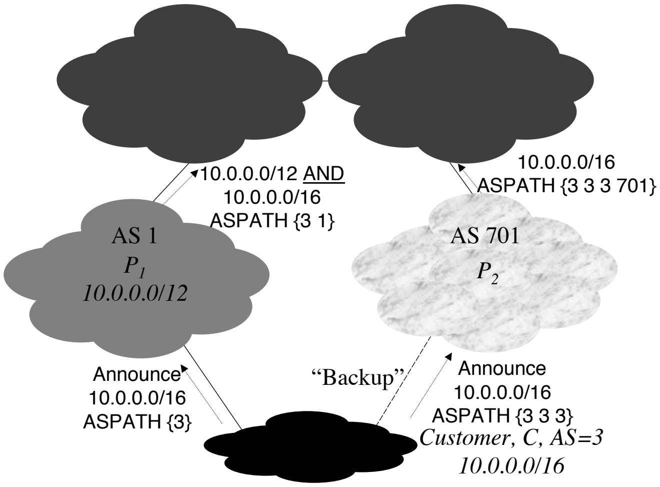


Figure 4-9: Customer C is multi-homed with providers P_1 and P_2 and uses provider-based addressing from P_1. C announces routes to itself on both P_1 and P_2, but to ensure that P_2 is only a backup, it uses a hack that pads the ASPATH attribute. However, notice that P_1 must announce (to its providers and peers) *explicit* routes on both its regular address block *and* on the customer block, for otherwise the path through P_2 would match based on longest prefix in the upstream ASes!

■ 4.4 Failover and Scalability

BGP allows multiple links (and eBGP sessions) between two ASes, and this may used to provide some degree of fault tolerance and load balance. Overall, however, BGP wasn't designed for rapid fault detection and recovery, so these mechanisms are generally not particularly useful over short time scales. Furthermore, upon the detection of a fault, a router sends a withdrawal message to its neighbors. To avoid massive route oscillations, the further propagation of such route announcements is *damped*. Damping causes some delay (configurable using a timer) before problems can be detected and recovery initiated, and is a useful mechanism for scalability.

With BGP, faults may take minutes to detect and it may take several minutes for routes to converge to a consistent state afterwards.

■ 4.4.1 Multi-homing: Promise and Problems

Multi-homing typically refers to a technique by which a customer can exchange routes and packets over multiple distinct provider ASes. An example is shown in Figure 4-9, which shows the topology and address blocks of the concerned parties. This example uses *provider-based addressing* for the customer, which allows the routing state in the Internet backbones to scale better because transit providers can aggregate address blocks across several customers into one or a small number of route announcements to their respective providers.

Today, multi-homing doesn't actually work while still preserving the scalability of

Researchers	Finding	Time-frame
Paxson	Serious routing pathology rate of 3.3%	1995
Labovitz *et al.*	10% of routes available less than 95% of the time	1997
Labovitz *et al.*	Less than 35% of routes available 99.99% of the time	1997
Labovitz *et al.*	40% of path outages take 30+ minutes to repair	2000
Chandra *et al.*	5% of faults last more than 2 hours, 45 minutes	2001
Andersen *et al.*	Between 0.23% and 7.7% of overlay "path-hours" experienced serious 30-minute problems in 16-node overlay	2001

Table 4-3: Internet path failure observations, as reported by several studies.

the routing infrastructure. Figure 4-9 shows why. Here the customer (C) address block 10.0.0.0/16 needs to be advertised not only from provider P_2 to the rest of the Internet, but *also* from provider P_1. If P_1 didn't do so, then longest prefix matching would cause all packets to the customer to arrive via P_2's link, which would defeat the purpose of using P_2 only as a backup path.

Now, given that this route needs to be advertised on both paths, how does C ensure that both paths aren't used? One hack to achieve this is by *padding* the exported ASPATH attribute. On the path through P_1, the normal ASPATH is announced, while on the path through P_2, a longer path is advertised by padding it with C's AS number multiple times.

A good way to do extensive multi-homing without affecting routing scalability is a good open problem. In addition to the fact that customer routes must be advertised along multiple paths, effective multi-homing today is often not possible unless the customer has a large address block. To limit the size of their routing tables, many ISPs will not accept routing announcements for fewer than 8192 contiguous addresses (a "/19" netblock).[3] Small companies, regardless of their fault-tolerance needs, do not often require such a large address block, and cannot effectively multi-home. Notice that provider-based addressing doesn't really work, since this requires handling two distinct sets of addresses on its hosts. It is unclear how *on-going* connections (*e.g.*, long-running ssh tunnels, which are becoming increasingly common) on one address set can seamlessly switch on a failure in this model.

■ 4.4.2 Convergence Problems

BGP does not always converge quickly after a fault is detected and routes withdrawn. Depending on the eBGP session topology between ASes, this could involve the investigation of many routes before route convergence occurs. The paper by Labovitz *et al.* from ACM SIGCOMM 2000 explains this in detail, and shows that under some conditions this could take a super-exponential number of steps.

In practice, it's been observed that wide-area routes are often (relative to what's needed for "mission-critical" applications) unavailable. Although extensive data is lacking, the observations summarized in Table 4-3 are worth noting.[4]

[3] This used to be the case, although it seems as if they have relaxed this to /24 in many cases now.

[4] These are, unfortunately, a few years old. I need to update the table to include newer results; *e.g.*, from Nick Feamster's dissertation.

4.4.3 Correctness Problems

In the next lecture, we will study routing correctness and anomalies in depth. We discuss three key correctness properties: *route validity*, *path visibility*, and *route safety*, and show how current interdomain routing violates all three properties some of the time. We also discuss steps that could be taken to achieve correct routing in practice.

4.5 Summary

This lecture looked at issues in wide-area unicast Internet routing, focusing on real-world issues. We first looked at inter-AS relationships and dealt with transit and peering issues. We then discussed many salient features and quirks of BGP, the prevalent wide-area routing protocol today.

BGP is actually a rather simple protocol, but its operation in practice is extremely complex. Its complexity stems from configuration flexibility, which allows for a rich set of attributes to be exchanged in route announcements. There are a number of open and interesting research problems in the area of wide-area routing, relating to policy, failover, scalability, configuration, and correctness. Despite much activity and impressive progress over the past few years, interdomain routing remains hard to understand and model.

The next lecture will discuss routing correctness and routing anomalies.

Acknowledgments

This lecture has evolved over the past four years, thanks in large part to the wonderful collaboration that Nick Feamster and I have had. A few subsections and two figures of these notes are edited from Nick's dissertation. Thanks also to Jennifer Rexford and Mythili Vutukuru for several discussions.

References

[1] T. Bates, R. Chandra, and E. Chen. *BGP Route Reflection - An Alternative to Full Mesh IBGP*. Internet Engineering Task Force, Apr. 2000. RFC 2796. (Cited on page 11.)

[2] I. V. Beijnum. *BGP*. O'Reilly and Associates, Sept. 2002. (Cited on page 8.)

[3] D. Clark and D. Tennenhouse. Architectural Consideration for a New Generation of Protocols. In *Proc. ACM SIGCOMM*, pages 200–208, Philadelphia, PA, Sept. 1990. (Cited on page 35.)

[4] R. Dube. A Comparison of Scaling Techniques for BGP. *ACM Computer Communications Review*, 29(3):44–46, July 1999. (Cited on page 9.)

[5] Cisco IOS IP Comand Reference, ebgp-multihop. `http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_command_reference_chapter09186a00800ca79d.html`, 2005. (Cited on page 12.)

[6] N. Feamster and H. Balakrishnan. Detecting BGP Configuration Faults with Static Analysis. In *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, pages 43–56, Boston, MA, May 2005. (Cited on page 12.)

[7] T. Griffin and G. Wilfong. On the Correctness of IBGP Configuration. In *Proc. ACM SIGCOMM*, pages 17–29, Pittsburgh, PA, Aug. 2002. (Cited on pages 9 and 12.)

[8] C. Hedrick. *Routing Information Protocol*. Internet Engineering Task Force, June 1988. RFC 1058. (Cited on page 2.)

[9] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM*, pages 314–329, Stanford, CA, Aug. 1988. (Cited on page 37.)

[10] P. Karn. MACA – A New Channel Access Method for Packet Radio. In *Proc. 9th ARRL Computer Networking Conference*, 1990. (Cited on page 39.)

[11] J. Moy. *OSPF Version 2*, Mar. 1994. RFC 1583. (Cited on page 2.)

[12] D. Oran. *OSI IS-IS intra-domain routing protocol*. Internet Engineering Task Force, Feb. 1990. RFC 1142. (Cited on page 2.)

[13] Y. Rekhter and T. Li. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force, Mar. 1995. RFC 1771. (Cited on pages 2 and 8.)

[14] Y. Rekhter, T. Li, and S. Hares. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force, Oct. 2004. `http://www.ietf.org/internet-drafts/draft-ietf-idr-bgp4-26.txt` Work in progress, expired April 2005. (Cited on page 2.)

[15] P. Traina, D. McPherson, and J. Scudder. *Autonomous System Confederations for BGP*. Internet Engineering Task Force, Feb. 2001. RFC 3065. (Cited on page 11.)