There is no separate specification of the forwarding function in IP. Instead, forwarding is covered by the protocol specifications for the internet layer protocols ([INTERNET:1], [INTERNET:2], [INTERNET:3], [INTERNET:8], and [ROUTE:11]).

**5.2 FORWARDING WALK-THROUGH**

**Forwarding Algorithm**

Much of the challenge of writing router software is to maximize the rate at which the router can forward packets while still achieving the same effect of the algorithm. It is heavily dependent on the architecture of the router.

Instead, we merely point out the order dependencies among the steps:

1. A router MUST verify the IP header, before performing any actions based on the contents of the header. This allows the router to detect and discard bad packets before the expenditure of other resources.
2. Processing of certain IP options requires that the router insert its IP address into the option. The address inserted MUST be the address of the logical interface on which the packet is sent or the router’s router-id if the packet is sent over an unnumbered interface. Thus, processing of these options cannot be completed until after the output interface is chosen.
3. The router cannot check and decrement the TTL before checking whether the packet should be delivered to the router itself.
4. More generally, when a packet is delivered locally to the router, its IP header MUST NOT be modified in any way (except that a router may be required to insert a timestamp into any Timestamp options in the IP header). Thus, before the router determines whether the packet is to be delivered locally to the router, it cannot update the IP header in any way that it is not prepared to undo.

**General**

This section covers the general forwarding algorithm. This algorithm applies to **all forms of packets to be forwarded: unicast, multicast and broadcast**.

(1) The router receives the IP packet from the Link Layer.

(2) The router validates the IP header.Note that IP reassembly is not done, except on IP fragments to be queued for local delivery in step (4).

(3) The router performs most of the processing of any IP options. Some IP options require additional processing after the routing decision has been made.

(4) The router examines the destination IP address of the IP datagram, to determine how it should continue to process the IP datagram. There are three possibilities:

* The IP datagram is destined for the router, and should be queued for local delivery, doing reassembly if needed.
* The IP datagram is not destined for the router, and should be queued for forwarding.
* The IP datagram should be queued for forwarding, but (a copy) must also be queued for local delivery.

**Unicast**

The following assumes that the IP datagram was queued for forwarding. If the destination is an IP unicast address:

1. **The forwarder** determines the **next hop IP address for the packet,usually by looking up the packet’s destination in the router’s routing table.**This procedure also decides which network **interface** should be used to send the packet.
2. **The forwarder verifies that forwarding the packet is permitted. The source and destination addresses should be valid.** If the router supports administrative constraints on forwarding, those constraints must be satisfied.
3. **The forwarder** decrements (by at least one) and checks the **packet’s TTL.**
4. **The forwarder** performs any IP option processing that could not be completed in step 3.
5. **The forwarder** performs any necessary IP fragmentation. Since this step occurs after outbound interface selection (step 5), all fragments of the same datagram will be transmitted out the same interface.
6. **The forwarder** determines the Link Layer address of the packet’s next hop. The mechanisms for doing this are Link Layer-dependent.
7. The forwarder encapsulates the IP datagram (or each of the fragments thereof) in an appropriate Link Layer frame and queues it for output on the interface selected in step 5.
8. The forwarder sends an ICMP redirect if necessary.

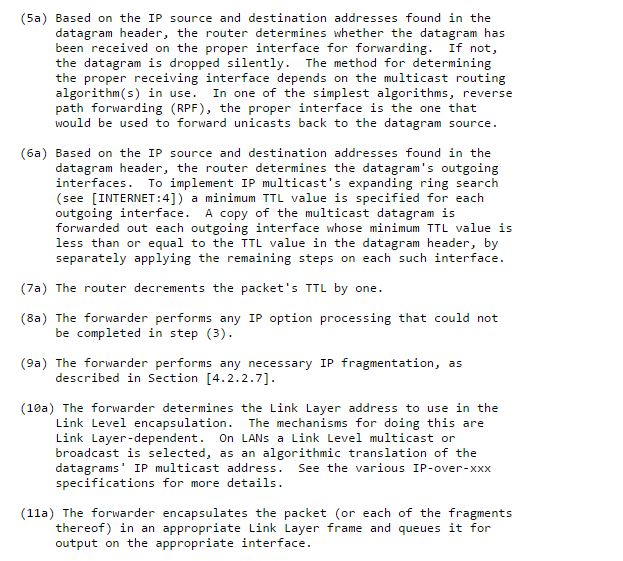
**Multicast**

If the destination is an IP multicast, the following steps are taken.

Note that the main differences between the forwarding of IP unicasts and the forwarding of IP multicasts are

* IP multicasts are usually forwarded based on both the datagram's source and destination IP addresses,
* IP multicast uses an expanding ring search,
* IP multicasts are forwarded as Link Level multicasts, and
* ICMP errors are never sent in response to IP multicast datagrams.

Note that the forwarding of IP multicasts is still somewhat experimental. As a result, the algorithm presented below is not mandatory, and is provided as an example only.

****

**IP Header Validation**

Before a router can process any IP packet, it MUST perform the following basic validity checks on the packet's IP header to ensure that the header is meaningful. If the packet fails any of the following tests, it MUST be silently discarded, and the error SHOULD be logged.

1. The packet length reported by the Link Layer must be large enough to hold the minimum length legal IP datagram (20 bytes).
2. The IP checksum must be correct.
3. The IP version number must be 4. If the version number is not 4 then the packet may be another version of IP, such as IPng or ST-II.
4. The IP header length field must be large enough to hold the minimum length legal IP datagram (20 bytes = 5 words).
5. The **IP total length field must be large enough to hold the IP datagram header**, whose length is specified in the IP header length field.

A router MUST NOT have a configuration option that allows disabling any of these tests.

**If the packet passes the second and third tests, the IP header length field is at least 4, and both the IP total length field and the packet length reported by the Link Layer are at least 16 then, despite the above rule, the router MAY respond with an ICMP Parameter Problem message, whose pointer points at the IP header length field (if it failed the fourth test) or the IP total length field (if it failed the fifth test). However, it still MUST discard the packet and still SHOULD log the error.**

These rules (and this entire document) apply only to version 4 of the Internet Protocol. **These rules should not be construed as prohibiting routers from supporting other versions of IP. Furthermore, if a router can truly classify a packet as being some other version of IP then it ought not treat that packet as an error packet within the context of this memo.**

**IMPLEMENTATION**

It is desirable for purposes of error reporting, **though not always entirely possible, to determine why a header was invalid. There are four possible reasons**:

* **The Link Layer truncated the IP header**
* **The datagram is using a version of IP other than the standard one (version 4).**
* **The IP header has been corrupted in transit.**
* **The sender generated an illegal IP header.**

It is probably desirable to perform the checks in the order listed, since we believe that this ordering is most likely to correctly categorize the cause of the error. **For purposes of error reporting, it may also be desirable to check if a packet that fails these tests has an IP version number indicating IPng or ST-II; these should be handled according to their respective specifications.**

**Additionally, the router SHOULD verify that the packet length reported by the Link Layer is at least as large as the IP total length recorded in the packet's IP header. If it appears that the packet has been truncated, the packet MUST be discarded, the error SHOULD be logged, and the router SHOULD respond with an ICMP Parameter Problem message whose pointer points at the IP total length field.**

Finally, if the destination address in the IP header is not one of the addresses of the router, the router SHOULD verify that the packet does not contain a Strict Source and Record Route option. If a packet fails this test (if it contains a strict source route option), the router SHOULD log the error and SHOULD respond with an ICMP Parameter Problem error with the pointer pointing at the offending packet's IP destination address.

**Local Delivery Decision**

When a router receives an IP packet, it must decide whether the packet is addressed to the router (and should be delivered locally) or the packet is addressed to another system (and should be handled by the forwarder). There is also a hybrid case, where certain IP broadcasts and IP multicasts are both delivered locally and forwarded. A router MUST determine which of the these three cases applies using the following rules.

* An unexpired source route option is one whose pointer value does not point past the last entry in the source route. If the packet contains an unexpired source route option, the pointer in the option is advanced until either the pointer does point past the last address in the option or else the next address is not one of the router's own addresses. In the latter (normal) case, the packet is forwarded (and not delivered locally) regardless of the rules below.
* The packet is delivered locally and not considered for forwarding in the following cases:
  + The packet's destination address exactly matches one of the router's IP addresses,
  + The packet's destination address is a limited broadcast address ({-1, -1}), or
  + The packet's destination is an IP multicast address which is never forwarded (such as 224.0.0.1 or 224.0.0.2) and (at least) one of the logical interfaces associated with the physical interface on which the packet arrived is a member of the destination multicast group.
* The packet is **passed to the forwarder AND delivered locally in the following cases:**
  + The packet's destination address is an IP broadcast address that addresses **at least one of the router's logical interfaces but does not address any of the logical interfaces associated with the physical interface on which the packet arrived**
  + The packet's destination is an IP multicast address which is permitted to be forwarded (unlike 224.0.0.1 and 224.0.0.2) and (at least) **one of the logical interfaces associated with the physical interface on which the packet arrived is a member of the destination multicast group.**
* **The packet is delivered locally if the packet's destination address is an IP broadcast address (other than a limited broadcast address) that addresses at least one of the logical interfaces associated with the physical interface on which the packet arrived.** The packet is ALSO passed to the forwarder unless the link on which the packet arrived uses an IP encapsulation that does not encapsulate broadcasts differently than unicasts (e.g., by using different Link Layer destination addresses).
* The packet is passed to the forwarder in all other cases.

**IMPLEMENTATION**

Packets received as Link Layer broadcasts are generally not forwarded. It may be advantageous to avoid passing to the forwarder packets it would later discard because of the rules in the section.

**Some Link Layers (either because of the hardware or because of special code in the drivers) can deliver to the router copies of all Link Layer broadcasts and multicasts it transmits. Use of this feature can simplify the implementation of cases where a packet has to both be passed to the forwarder and delivered locally, since forwarding the packet will automatically cause the router to receive a copy of the packet that it can then deliver locally.** One must use care in these circumstances to prevent treating a received loop-back packet as a normal packet that was received (and then being subject to the rules of forwarding, etc.).

Even without such a Link Layer, it is of course hardly necessary to make a copy of an entire packet to queue it both for forwarding and for local delivery, though care must be taken with fragments, since reassembly is performed on locally delivered packets but not on forwarded packets. One simple scheme is to associate a flag with each packet on the router's output queue that indicates whether it should be queued for local delivery after it has been sent.

**Determining the Next Hop Address**

When a router is going to forward a packet, it must determine whether it can send it directly to its destination, or whether it needs to pass it through another router. If the latter, it needs to determine which router to use. This section explains how these determinations are made.

This section makes use of the following definitions:

* LSRR - IP Loose Source and Record Route option
* SSRR - IP Strict Source and Record Route option
* Source Route Option - an LSRR or an SSRR
* **Ultimate Destination Address** - where the packet is being sent to: the last address in the source route of a source-routed packet, or the destination address in the IP header of a non-source-routed packet
* **Adjacent** - reachable without going through any IP routers
* **Next Hop Address** - the IP address of the adjacent host or router to which the packet should be sent next
* **IP Destination Address** - the ultimate destination address, except in source routed packets, where it is the next address specified in the source route
* **Immediate Destination** - the node, System, router, end-system, or whatever that is addressed by the IP Destination Address.

### IP Destination Address

If:

* the destination address in the IP header is one of the addresses of the router,
* the packet contains a Source Route Option, and
* the pointer in the Source Route Option does not point past the end of the option,

**then the next IP Destination Address is the address pointed at by the pointer in that option.**

If:

* the destination address in the IP header is one of the addresses of the router,
* the packet contains a Source Route Option, and
* the pointer in the Source Route Option points past the end of the option,

**then the message is addressed to the system analyzing the message.**

A router **MUST use the IP Destination Address, not the Ultimate Destination Address (the last address in the source route option)**, when determining how to handle a packet.

It is an error for more than one source route option to appear in a datagram. **If it receives such a datagram, it SHOULD discard the packet and reply with an ICMP Parameter Problem message whose pointer points at the beginning of the second source route option.**

**Local/Remote Decision**

After it has been determined that the IP packet needs to be forwarded according to the rules specified in Section ,the following algorithm MUST be used to determine if the Immediate Destination is directly accessible.

1. For **each network interface that has not been assigned any IP address, compare the router-id of the other end of the line to the IP Destination Address. If they are exactly equal, the packet can be transmitted through this interface.**
2. If no network interface has been selected in the first step, for each IP address assigned to the router:
   1. Isolate the network prefix used by the interface.

IMPLEMENTATION

The result of this operation will usually have been computed and saved during initialization.

* 1. Isolate the corresponding set of bits from the IP Destination Address of the packet.
  2. Compare the resulting network prefixes. If they are equal to each other, the packet can be transmitted through the corresponding network interface.

1. If the destination was neither the router-id of a neighbor on an unnumbered interface nor a member of a directly connected network prefix, the IP Destination is accessible only through some other router. In the case of a host that is not also a router, this may be the configured default router.

**The Next Hop Routing Protocol (NHRP) enables IP entities to determine the "optimal" link layer address to be used to traverse such a link layer network towards a remote destination.**

1. If the selected "next hop" is reachable through an interface configured to use NHRP, then the following additional steps apply:
   1. **Compare the IP Destination Address to the destination addresses in the NHRP cache.** If the address is in the cache, then send the datagram to the corresponding cached link layer address.
   2. **If the address is not in the cache, then construct an NHRP request packet containing the IP Destination Address.** This message is sent to the NHRP server configured for that interface. This may be a logically separate process or entity in the router itself.
   3. **The NHRP server will respond with the proper link layer address to use to transmit the datagram and subsequent datagrams to the same destination.** The system MAY transmit the datagram(s) to the traditional "next hop" router while awaiting the NHRP reply.

### Next Hop Address

**The router applies the algorithm in the previous section to determine if the IP Destination Address is adjacent. If so, the next hop address is the same as the IP Destination Address. Otherwise, the packet must be forwarded through another router to reach its Immediate Destination. The selection of this router is the topic of this section**.

If the packet contains an SSRR, the router MUST discard the packet and reply with an ICMP Bad Source Route error. Otherwise, the router looks up the IP Destination Address in its routing table to determine an appropriate next hop address.

Per the IP specification, a Strict Source Route must specify a sequence of nodes through which the packet must traverse; **the packet must go from one node of the source route to the next, traversing intermediate networks only. Thus, if the router is not adjacent to the next step of the source route, the source route can not be fulfilled.** Therefore, the router rejects such with an ICMP Bad Source Route error.

The goal of the next-hop selection process is to examine the entries in the router's Forwarding Information Base (FIB) and select the best route (if there is one) for the packet from those available in the FIB.

Conceptually, any route lookup algorithm starts out with a set of candidate routes that consists of the entire contents of the FIB. The algorithm consists of a series of steps that discard routes from the set. **These steps are referred to as Pruning Rules.** -Normally, when the algorithm terminates there is exactly one route remaining in the set. If the set ever becomes empty, the packet is discarded because the destination is unreachable. **It is also possible for the algorithm to terminate when more than one route remains in the set. In this case, the router may arbitrarily discard all but one of them, or may perform "loadsplitting" by choosing whichever of the routes has been least recently used.**

With the exception of rule 3 (Weak TOS), a router MUST use the following Pruning Rules when selecting a next hop for a packet. If a router does consider TOS when making next-hop decisions, the Rule 3 must be applied in the order indicated below. These rules MUST be (conceptually) applied to the FIB in the order that they are presented.

1. Basic Match

**This rule discards any routes to destinations other than the IP Destination Address of the packet. For example, if a packet's IP Destination Address is 10.144.2.5, this step would discard a route to net 128.12.0.0/16 but would retain any routes to the network prefixes 10.0.0.0/8 and 10.144.0.0/16, and any default routes.**

More precisely, we assume that each route has a destination attribute, called **route.dest** and a corresponding prefix length, called **route.length**, to specify which bits of **route.dest** are significant**. The IP Destination Address of the packet being forwarded is ip.dest. This rule discards all routes from the set of candidates except those for which the most significant route.length bits of route.dest and ip.dest are equal.**

**For example**, if a packet's IP Destination Address is 10.144.2.5 and there are network prefixes 10.144.1.0/24, 10.144.2.0/24, and 10.144.3.0/24, this rule would keep only 10.144.2.0/24; it is the only route whose prefix has the same value as the corresponding bits in the IP Destination Address of the packet.

1. Longest Match

Longest Match is a refinement of Basic Match, described above. **After performing Basic Match pruning, the algorithm examines the remaining routes to determine which among them have the largest route.length values. All except these are discarded.**

**For example**, if a packet's IP Destination Address is 10.144.2.5 and there are network prefixes 10.144.2.0/24, 10.144.0.0/16, and 10.0.0.0/8, then this rule would keep only the first (10.144.2.0/24) because its prefix length is longest.

1. Weak TOS

Each route has a type of service attribute, called **route.tos**, whose possible values are assumed to be identical to those used in the TOS field of the IP header. **Routing protocols that distribute TOS information fill in route.tos appropriately in routes they add to the FIB; routes from other routing protocols are treated as if they have the default TOS (0000). The TOS field in the IP header of the packet being routed is called ip.tos.**

**The set of candidate routes is examined to determine if it contains any routes for which route.tos = ip.tos. If so, all routes except those for which route.tos = ip.tos are discarded.** **If not, all routes except those for which route.tos = 0000 are discarded from the set of candidate routes.**

1. Best Metric

Each route has a metric attribute, called **route.metric, and a routing domain identifier, called route.domain. Each member of the set of candidate routes is compared with each other member of the set. If route.domain is equal for the two routes and route.metric is strictly inferior for one when compared with the other, then the one with the inferior metric is discarded from the set.** The determination of inferior is usually by a simple arithmetic comparison, though some protocols may have structured metrics requiring more complex comparisons.

1. Vendor Policy

Vendor Policy is sort of a **catch-all to make up for the fact that the previously listed rules are often inadequate to choose from the possible routes.** Vendor Policy pruning rules are extremely vendor-specific.

This algorithm has two distinct disadvantages. Presumably, a router implementor might develop techniques to deal with these disadvantages and make them a part of the Vendor Policy pruning rule.

1. **IS-IS and OSPF route classes are not directly handled.**
2. **Path properties other than type of service (e.g., MTU) are ignored.**

It is also worth noting a deficiency in the way that TOS is supported: **routing protocols that support TOS are implicitly preferred when forwarding packets that have non-zero TOS values.**

The Basic Match and Longest Match pruning rules generalize the treatment of a number of particular types of routes. These routes are selected in the following, decreasing, order of preference:

1. **Host Route: This is a route to a specific end system.**
2. **Hierarchical Network Prefix Routes: This is a route to a particular network prefix. Note that the FIB may contain several routes to network prefixes that subsume each other (one prefix is the other prefix with additional bits). These are selected in order of decreasing prefix length.**
3. **Default Route: This is a route to all networks for which there are no explicit routes. It is by definition the route whose prefix length is zero.**

If, after application of the pruning rules, **the set of routes is empty (i.e., no routes were found), the packet MUST be discarded and an appropriate ICMP error generated (ICMP Bad Source Route if the IP Destination Address came from a source route option; otherwise, whichever of ICMP Destination Host Unreachable or Destination Network Unreachable is appropriate**

**Note:**

A **forwarding information base** (**FIB**), also known as a forwarding table or CAM table, is most commonly used in **network** bridging, routing, and similar functions to find the proper interface to which the input interface should forward a packet.

### Administrative Preference

One suggested mechanism for the Vendor Policy Pruning Rule is to use administrative preference, which is a simple prioritization algorithm. The idea is to manually prioritize the routes that one might need to select among.

Each route has associated with it a preference value, **based on various attributes of the route (specific mechanisms for assignment of preference values are suggested below). This preference value is an integer in the range [0..255], with zero being the most preferred and 254 being the least preferred. 255 is a special value that means that the route should never be used.** The first step in the Vendor Policy pruning rule discards all but the most preferable routes (and always discards routes whose preference value is 255).

**This policy is not safe in that it can easily be misused to create routing loops.** Since no protocol ensures that the preferences configured for a router is consistent with the preferences configured in its neighbors, network managers must exercise care in configuring preferences.

* **Address Match**   
  It is useful to be able to assign a single preference value to all routes (learned from the same routing domain) to any of a specified set of destinations, where the set of destinations is all destinations that match a specified network prefix.
* **Route Class**  
  For routing protocols which maintain the distinction, it is useful to be able to assign a single preference value to all routes (learned from the same routing domain) which have a particular route class **(intra-area, inter-area, external with internal metrics, or external with external metrics)**.
* **Interface**  
  It is useful to be able to assign a single preference value to all routes (learned from a particular routing domain) that would cause packets to be routed out a particular logical interface on the router (logical interfaces generally map one-to-one onto the router's network interfaces, except that any network interface that has multiple IP addresses will have multiple logical interfaces associated with it).
* **Source router**  
  It is useful to be able to assign a single preference value to all routes (learned from the same routing domain) that were learned from any of a set of routers, where the set of routers are those whose updates have a source address that match a specified network prefix.
* **Originating AS**   
  For routing protocols which provide the information, it is useful to be able to assign a single preference value to all routes (learned from a particular routing domain) which originated in another particular routing domain. For **BGP routes, the originating AS is the first AS listed in the route's AS\_PATH attribute. For OSPF external routes, the originating AS may be considered to be the low order 16 bits of the route's external route tag if the tag's Automatic bit is set and the tag's Path Length is not equal to 3.**
* External route tag   
  It is useful to be able to assign a single preference value to all OSPF external routes (learned from the same routing domain) whose external route tags match any of a list of specified values. Because the external route tag may contain a structured value, it may be useful to provide the ability to match particular subfields of the tag.
* AS path   
  It may be useful to be able to assign a single preference value to all BGP routes (learned from the same routing domain) whose AS path "matches" any of a set of specified values. It is not yet clear exactly what kinds of matches are most useful. A simple option would be to allow matching of all routes for which a particular AS number appears (or alternatively, does not appear) anywhere in the route's AS\_PATH attribute. A more general but somewhat more difficult alternative would be to allow matching all routes for which the AS path matches a specified regular expression.

### Load Splitting

At the end of the Next-hop selection process, multiple routes may still remain. A router has several options when this occurs. It may arbitrarily discard some of the routes. It may reduce the number of candidate routes by comparing metrics of routes from routing domains that are not considered equivalent. It may retain more than one route and employ a load-splitting mechanism to divide traffic among them. Perhaps the only thing that can be said about the relative merits of the options is that load-splitting is useful in some situations but not in others, so a wise implementor who implements load-splitting will also provides a way for the network manager to disable it.