

## Definition#

Unnumbered interfaces are point-to-point interfaces which aren't assigned their own unique IP address/netmask. They reuse the IP address of another interface on the system, which conserves IP addresses. This IP address is referred to as the "router-ID".

## Adding routes#

Normally, adding a route through a specific interface would be done by using the route command with the '-iface' option, followed by the IP address of the interface to use. With an unnumbered interface, this option won't work because the interface doesn't have a specific IP address. There is another obscure and relatively unknown option to the route command. It's the '-ifp' option. It can be followed by an interface name. Here is an example:

```
# route add -net 10.3.0.0/16 <gateway> -ifp ppp0:
```

Where <gateway> is the other end of the ppp link

## Really Unnumbered?#

The definition of an unnumbered interface is vague enough that an interface with no IP address could also be considered unnumbered. On some operating systems it is possible to just bring up an interface and use it for routing traffic to A.B.C.D without even having assigned an address to the interface:

```
# ifconfig if0 up
# route add A.B.C.D -link if0:
```

This isn't currently possible on QNX, as the interfaces on our systems will either be Ethernet related, which requires some degree of configuration to allow ARP to work or they will be PPP interfaces brought up with pppd, which will assign an IP address to the interface during the IPCP negotiation. Other interfaces types, like ATM or point-to-point Ethernet (**add link**) might allow such behaviour.

While it is interesting, having an interface with no IP address isn't a useful feature. A router **needs** to have at least one IP address to be a router, so this address can be assigned to the unnumbered interface without changing the way the system will route. While it is perfectly fine to have an interface with no IP address (as far as any standards go), it could confuse routing software or other software which queries the stack about which interfaces are present.

## Joining multicast groups with unnumbered interfaces#

The standard API to join a multicast group involves an ip\_mreq struct.

```
/*
 * Argument structure for IP_ADD_MEMBERSHIP and IP_DROP_MEMBERSHIP.
 */
struct ip_mreq {
    struct in_addr imr_multiaddr; /* IP multicast address of group */
    struct in_addr imr_interface; /* local IP address of interface */
};
```

Received multicast won't be a problem, but for sending it will. The IP address of the unnumbered interface is common to other interfaces, so it's not a unique identifier. The QNX TCP/IP stack allows an interface index to be used instead of a local IP address. Any address that would be part of 0.0.0.0/8 is interpreted as interface index by the stack.

## Remote address of a point-to-point link<sup>#</sup>

Typically the remote address of a point-to-point link is known. This is especially true when using pppd (which negotiates the addresses), but on other types of interfaces (ATM, ptp Ethernet, etc..) this address has to be manually configured. It would be ideal if the local address were configured using a startup script and the remote address could be detected by a routing daemon, but some third party routing daemons may not work correctly. Two problems occur:

1. They try to validate a received multicast or broadcast packet's source IP address by making sure they are directly connected to it or a network to which it belongs, but this can't be done if the remote address of the point-to-point link is unknown.
2. They only keep track of the routes based on IP addresses (as opposed to keeping interface information). Routes advertised by the remote end of the point-to-point link may not be reachable if the peer's IP address is used as a gateway. For example, if the peer is 1.1.1.1 and has a route to 4.4.4.4, the routing software might not consider 4.4.4.4 as reachable, because 1.1.1.1 isn't directly connected (1.1.1.1 might be considered reachable through the ppp interface, but not in the same sense as a directly connected host would be).

Some RFCs, like the OSPF RFC, seem to imply that it shouldn't be necessary to know the peer IP address of the link.