**Reliability:**

the network infrastructure is considered inherently unreliable at any single network element or transmission medium and is dynamic in terms of availability of links and nodes. No central monitoring or performance measurement facility exists that tracks or maintains the state of the network. For the benefit of reducing [network complexity](https://en.wikipedia.org/wiki/Network_complexity), the intelligence in the network is purposely located in the [end nodes](https://en.wikipedia.org/wiki/End_node).

As a consequence of this design, the Internet Protocol only provides [best-effort delivery](https://en.wikipedia.org/wiki/Best-effort_delivery) and its service is characterized as [unreliable](https://en.wikipedia.org/wiki/Reliability_(computer_networking)). In network architectural language, it is a [connectionless protocol](https://en.wikipedia.org/wiki/Connectionless_protocol), in contrast to [connection-oriented communication](https://en.wikipedia.org/wiki/Connection-oriented_communication). Various error conditions may occur, such as [data corruption](https://en.wikipedia.org/wiki/Data_corruption), [packet loss](https://en.wikipedia.org/wiki/Packet_loss) and duplication. Because routing is dynamic, meaning every packet is treated independently, and because the network maintains no state based on the path of prior packets, different packets may be routed to the same destination via different paths, resulting in [out-of-order delivery](https://en.wikipedia.org/wiki/Out-of-order_delivery) to the receiver.

All error conditions in the network must be detected and compensated by the participating end nodes. The [upper layer protocols](https://en.wikipedia.org/wiki/Upper_layer_protocol) of the Internet protocol suite are responsible for resolving reliability issues. For example, a host may [buffer](https://en.wikipedia.org/wiki/Data_buffer) network data to ensure correct ordering before the data is delivered to an application.

IPv4 provides safeguards to ensure that the IP packet header is error-free. A routing node calculates a [checksum](https://en.wikipedia.org/wiki/Checksum) for a packet. If the checksum is bad, the routing node discards the packet.

**Path MTU Discovery:**

The IPv4 internetworking layer has the ability to automatically [fragment](https://en.wikipedia.org/wiki/IP_fragmentation) the original datagram into smaller units for transmission. In this case, IP provides re-ordering of fragments delivered out of order.

 An IPv6 network does not perform fragmentation or reassembly, and as per the [end-to-end principle](https://en.wikipedia.org/wiki/End-to-end_principle), requires end stations and higher-layer protocols to avoid exceeding the network's MTU.

* It is usually preferable IP datagrams be of the largest size that does not require fragmentation anywhere along the path from the source to the destination. (For the case against fragmentation, see.) This **datagram size** is referred to as the Path MTU (PMTU), and it is equal to the minimum of the **MTUs of each hop in the path**.
* In this memo, we describe a technique for using the Don't Fragment (DF) bit in the IP header to dynamically discover the PMTU of a path.
* The basic idea is that a source host initially assumes that the PMTU of a path is the (known) MTU of its first hop, and sends all datagrams on that path with the DF bit set.
* If any of the datagrams are too large to be forwarded without fragmentation by some router along the path, that router will discard them and return ICMP Destination Unreachable messages with a code meaning "fragmentation needed and DF set" [[7](https://tools.ietf.org/html/rfc1191#ref-7)]. Upon receipt of such a message ("Datagram Too Big" message), **the source host** reduces its assumed **PMTU** for the path.
* The PMTU discovery process ends when the host's estimate of the PMTU is low enough that its datagrams can be delivered without fragmentation.
* Normally, the host continues to set DF in all datagrams, so that if the route changes and the new PMTU is lower, it will be discovered. PTMU should not be **less than 68**, since every router "must be able to forward a datagram of **68 octets** without fragmentation"
* Hosts using PMTU Discovery MUST detect decreases in Path MTU as fast as possible.
* But an attempt to detect an increase (by sending a datagram larger than the current

estimate) MUST NOT be done **less than 5 minutes** after a **Datagram Too Big message** has been received for the given destination, or less than **1 minute** after a previous, successful attempted increase. Werecommend setting these timers **at twice their minimum values** (10minutes and 2 minutes, respectively).