

# *Carrier-sense multiple access with collision detection*

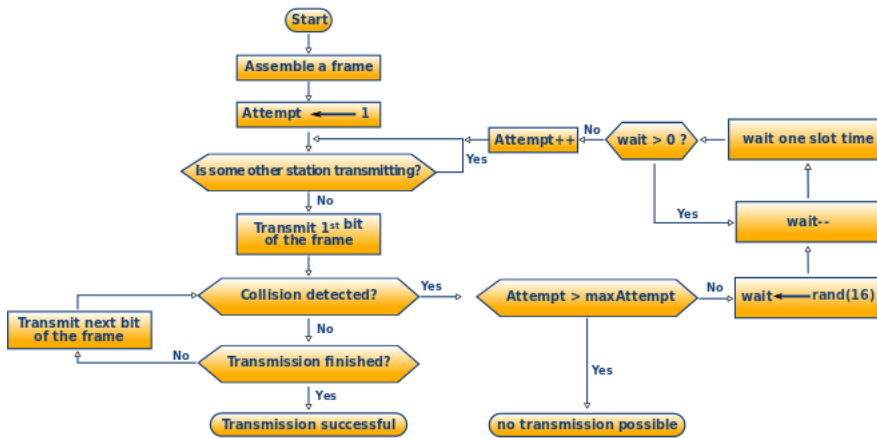
**Carrier-sense multiple access with collision detection (CSMA/CD)** is a [media access control](#) (MAC) method used most notably in early [Ethernet](#) technology for [local area networking](#). It uses [carrier](#)-sensing to defer transmissions until no other stations are transmitting. This is used in combination with collision detection in which a transmitting station detects collisions by sensing transmissions from other stations while it is transmitting a [frame](#). When this collision condition is detected, the station stops transmitting that frame, transmits a jam signal, and then waits for a random time interval before trying to resend the frame.<sup>[1]</sup>

CSMA/CD is a modification of pure [carrier-sense multiple access](#) (CSMA). CSMA/CD is used to improve CSMA performance by terminating transmission as soon as a collision is detected, thus shortening the time required before a retry can be attempted.

With the growing popularity of [Ethernet switches](#) in the 1990s, [IEEE 802.3](#) deprecated [Ethernet repeaters](#) in 2011,<sup>[2]</sup> making CSMA/CD and half-duplex operation less common and less important.

## Procedure

---



Simplified algorithm of CSMA/CD including retransmission logic used to resolve a detected collision.

The following procedure is used to initiate a transmission. The procedure is complete when the frame is transmitted successfully or a collision is detected during transmission. [3]:33

1. Is a frame ready for transmission? If not, wait for a frame.
2. Is medium idle? If not, wait until it becomes ready. [note 1]
3. Start transmitting and monitor for collision during transmission.
4. Did a collision occur? If so, go to collision detected procedure.
5. Reset retransmission counters and complete frame transmission.

The following procedure is used to resolve a detected collision. The procedure is complete when retransmission is initiated or the retransmission is aborted due to numerous collisions.

1. Continue transmission (with a jam signal instead of frame header/data/CRC) until minimum packet time is reached to ensure that all receivers detect the collision.
2. Increment retransmission counter.
3. Was the maximum number of transmission attempts reached? If so, abort transmission.
4. Calculate and wait the random backoff period based on number of collisions.
5. Re-enter main procedure at stage 1.

Methods for collision detection are media dependent. On a shared, electrical bus such as 10BASE5 or 10BASE2, collisions can be detected by comparing transmitted data with received

data or by recognizing a higher than normal signal amplitude on the bus.<sup>[4][5]</sup> On all other media, a carrier sensed on the receive channel while transmitting triggers a collision event.<sup>[6]</sup> Repeaters or **hubs** detect collisions on their own and propagate jam signals.<sup>[7][8]</sup>

The collision recovery procedure can be likened to what happens at a dinner party, where all the guests talk to each other through a common medium (the air). Before speaking, each guest politely waits for the current speaker to finish. If two guests start speaking at the same time, both stop and wait for short, random periods of time (in Ethernet, this time is measured in microseconds). The hope is that by each choosing a random period of time, both guests will not choose the same time to try to speak again, thus avoiding another collision.

## Jam signal

---

The **jam signal** or **jamming signal** is a **signal** that carries a **32-bit** binary pattern sent by a data station to inform the other stations of the collision and that they must not transmit.<sup>[9]</sup>

The maximum jam-time is calculated as follows: The maximum allowed diameter of an **Ethernet** installation is limited to 232 bits. This makes a round-trip-time of 464 bits. As the **slot time** in Ethernet is 512 bits, the difference between slot time and round-trip-time is 48 bits (6 bytes), which is the maximum *jam-time*.

This in turn means: A station noting a **collision** has occurred is sending a 4 to 6 byte long pattern composed of 16 1-0 bit combinations. Note: The size of this jam signal is clearly above the minimum allowed frame-size of 64 bytes.

The purpose of this is to ensure that any other node which may currently be receiving a frame will receive the jam signal in place of the correct 32-bit MAC CRC, this causes the other receivers to discard the frame due to a CRC error.

## Late collision

---

A **late collision** is a type of collision that happens further into the packet than is allowed for by the protocol standard in question. In 10 megabit shared medium Ethernet, if a collision error occurs after the first 512 bits of data are transmitted by the transmitting station,<sup>[10]</sup> a late collision is said to have occurred. Importantly, late collisions are not re-sent by the **NIC** unlike collisions occurring before the first 64 octets; it is left for the upper layers of the **protocol stack** to determine that there was loss of data.

As a correctly set up CSMA/CD network link should not have late collisions, the usual possible causes are full-duplex/half-duplex mismatch, exceeded Ethernet cable length limits, or defective hardware such as incorrect cabling, non-compliant number of hubs in the network, or a bad NIC.

## Local collision

---

A **local collision** is a collision that occurs at the **NIC** as opposed to on the wire. A NIC cannot detect a local collisions without attempting to send information.

On **UTP** cable, a local collision is detected on the local segment only when a station detects a signal on the RX pair at the same time it is sending on the TX pair. Since the two signals are on different pairs there is no characteristic change in the signal. Collisions are only recognized on UTP when the station is operating in **half-duplex**. The only functional difference between half and **full-duplex** operation in this regard is whether or not the transmit and receive pairs are permitted to be used simultaneously.

## Channel capture effect

---

The **channel capture effect** is a phenomenon where one user of a **shared medium** "captures" the medium for a significant time. During this period (usually 16 frames), other users are denied use of the medium. This effect was first seen in networks using CSMA/CD on Ethernet. Because of this effect, the most data-intense connection dominates the multiple-access wireless channel.<sup>[11]</sup> This happens in Ethernet links because of the way nodes "back off" from the link and attempt to re-access it. In the Ethernet protocol, when a communication collision happens (when two users of the medium try to send at the same time), each user waits for a random period of time before re-accessing the link. However, a user will wait ("back off") for a random amount of time proportional to the number of times it has successively tried to access the link. The channel capture effect happens when one user continues to "win" the link.

For example, user A and user B both try to access a quiet link at the same time. Since they detect a collision, user A waits for a random time between 0 and 1 time units and so does user B. Let's say user A chooses a lower back-off time. User A then begins to use the link and B allows it to finish sending its **frame**. If user A still has more to send, then user A and user B will cause another data collision. A will once again choose a random back-off time between 0 and 1, but user B will choose a back-off time between 0 and 3 – because this is B's second time colliding in a row. Chances are A will "win" this one again. If this continues, A will most likely win

all the collision battles, and after 16 collisions (the number of tries before a user backs down for an extended period of time), user A will have "captured" the channel.

The ability of one node to capture the entire medium is decreased as the number of nodes increases. This is because as the number of nodes increases, there is a higher probability that one of the "other" nodes will have a lower back-off time than the capturing node.

The channel capture effect creates a situation where one station is able to transmit while others are continually backing off, thus leading to a situation of short-term unfairness. Yet, the situation is long-term fair because every station has the opportunity to "capture" the medium once one station is done transmitting. The efficiency of the channel is increased when one node has captured the channel.

A negative side effect of the capture effect would be the idle time created due to stations backing off. Once one station is finished transmitting on the medium, large idle times are present because all other stations were continually backing off. In some instances, back-off can occur for so long that some stations actually discard packets because maximum attempt limits have been reached.

## Applications

---

CSMA/CD was used in now-obsolete [shared media Ethernet](#) variants ([10BASE5](#), [10BASE2](#)) and in the early versions of [twisted-pair Ethernet](#) which used [repeater hubs](#). Modern Ethernet networks, built with [switches](#) and [full-duplex](#) connections, no longer need to use CSMA/CD because each Ethernet segment, or [collision domain](#), is now isolated. CSMA/CD is still supported for backwards compatibility and for half-duplex connections. The [IEEE 802.3](#) standard, which defines all Ethernet variants, for historical reasons still bore the title "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications" until 802.3-2008, which uses new name "IEEE Standard for Ethernet".

## See also

---

- [Carrier-sense multiple access with collision avoidance](#) (CSMA/CA)

## Notes

---

1. On Ethernet, stations must additionally wait the 96 bit [interframe gap](#) period.

# References

---

1. "Carrier Sense Multiple Access Collision Detect (CSMA/CD) Explained" (<http://learn-networking.com/net-work-design/carrier-sense-multiple-access-collision-detect-csmacd-explained>) . learn-networking.com. January 29, 2008. Retrieved 2011-07-29.
  2. IEEE 802.3-2012 Clauses 9, 27, 41
  3. Heinz-Gerd Hegering; Alfred Lapple (1993). *Ethernet: Building a Communications Infrastructure* (<https://archive.org/details/ethernetbuilding0000hege>) . Addison-Wesley. ISBN 0-201-62405-2.
  4. IEEE 802.3 8.3.1.5 Collision detect thresholds
  5. IEEE 802.3 10.4.1.5 Collision detect thresholds
  6. IEEE 802.3 14.2.1.4 Collision Presence function requirements (half duplex mode only)
  7. IEEE 802.3 9.5.6 Collision handling
  8. IEEE 802.3 27.3.1.4 Collision handling functional requirements
  9. Forouzan, Behrouz A. (2010). *TCP/IP protocol suite* (4th ed.). Boston: McGraw-Hill Higher Education. p. 54. ISBN 978-0073376042.
  10. IEEE 802.3-2008 Section 1 (<http://standards.ieee.org/getieee802/802.3.html>) , IEEE (<http://ieee.org>) section 5.2.2.1.10
  11. Kopparty, S; Krishnamurthy, S. V.; Faloutsos, M.; Tripathi, S. K. (1998). "Split TCP for Mobile Ad Hoc Networks" (<http://www.cs.ucr.edu/~krish/splittcp.pdf>) (PDF). Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE. 1. pp. 138–142. doi:10.1109/GLOCOM.2002.1188057 (<https://doi.org/10.1109%2FGLOCOM.2002.1188057>) . ISBN 0-7803-7632-3. S2CID 18426 (<https://api.semanticscholar.org/CorpusID:18426>) .
- © This article incorporates public domain material from the General Services Administration document: "Federal Standard 1037C" (<https://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm>) .
  - IEEE 802.3 (<http://standards.ieee.org/getieee802/802.3.html>)
  - Ramakrishnan, K. K.; Yang, H. (1994). "The Ethernet Capture Effect: Analysis and Solution" ([http://www.research.att.com/~kkrama/papers/capture\\_camera.pdf](http://www.research.att.com/~kkrama/papers/capture_camera.pdf)) (PDF). *Conference on Local Computer Networks*. 19: 228–240. doi:10.1109/LCN.1994.386597 (<https://doi.org/10.1109%2FLCN.1994.386597>) . ISBN 0-8186-6680-3. S2CID 36231320 (<https://api.semanticscholar.org/CorpusID:36231320>) .

Retrieved from

["https://en.wikipedia.org/w/index.php?title=Carrier-sense\\_multiple\\_access\\_with\\_collision\\_detection&oldid=1040815581"](https://en.wikipedia.org/w/index.php?title=Carrier-sense_multiple_access_with_collision_detection&oldid=1040815581)

---

**Last edited 16 days ago by Kvng**

Content is available under CC BY-SA 3.0 unless otherwise noted.