

About Ethernet 10/100/1000 Copper Taps

Patrick Leong
October 18, 2007

Abstract

This article presents an overview of the various Ethernet 10/100/1000 physical layer technologies for the copper medium. It discusses the pros and cons of active versus passive tapping and why active tapping is preferred for Gigabit Ethernet running over the copper medium. The switch-over behavior of active relay-based tapping is also presented.

10, 100 and 1000 – It is more than adding zeroes

The Ethernet physical layer is typically implemented in a chip called PHY. In the old days each PHY can handle only one port. Today there are PHY chips that each chip can handle multiple ports and supports both the copper and optical media. For example, see PHY chips from Broadcom or Marvell. Also, the PHY may be integrated as part of a larger chip that has other functionalities such as the MAC layer, or even as part of the CPU chip.

Most users of Ethernet and Gigabit Ethernet pay little attention to how the physical layer works. From a user's perspective, typically the parameters that involve configuring the PHY are enabling or disabling auto-negotiation, setting up the auto-negotiation advertisement parameters such as the speed, duplex and flow control, or forcing them to certain values if auto-negotiation is disabled. There are obviously a lot more parameters in the PHY that can be configured, such as enabling or disabling automatic MDI crossover (AUTO-MDIX), the LED modes etc, but most of them are handled by the device driver and are invisible to the user.

Over the past some 25 years, the speed of Ethernet has increased from 10 Mbps in the early 1980's, to 100 Mbps (Fast Ethernet) in the mid-1990's, to 1000 Mbps (Gigabit Ethernet) in the late 1990's. We also have 10Gbps Ethernet since the early 2000's. The IEEE 802.3 standard governs these various technologies. Note that there are a few versions of Ethernet within each speed category. This reflects the fact that the technology is an evolving one, and that the market place plays a role in selecting the winners based on cost, features and availabilities. Table 1 shows a few of the Ethernet versions from each speed category for the copper medium. The common ones are marked with an asterisk.

Ethernet Version	Transmit Symbol Rate (Baud)	Encoding	Cabling	Duplex	IEEE Reference
10BASE-T*	10 MBd	Manchester (binary 1 = low->high, binary 0 = high->low transition in middle of a bit period)	2 pairs of CAT 3 (or better) UTP, one for each direction	Full or Half	802.3
100BASE-T2	25 MBd	PAM5x5 (each nibble->2 quinary symbols {-2, -1, 0, +1, +2})	2 pairs of CAT 3 (or better) UTP, each pair handles both directions	Full or Half	802.3 clause 32
100BASE-T4	33 Mbd	8B/6T (each octet->6 ternary symbols {-1, 0, +1}. Each 6 bit symbol, representing a data octet, is sent over one of the 3 pairs. Each byte being encoded and transmitted on the pairs in a round robin fashion.	4 pairs of CAT 3 (or better) UTP, 3 pairs for data transmission, 1 pair to detect collision	Half only	802.3 clause 23
100BASE-TX*	125 MBd	4B/5B (each nibble->5 bit binary code->MLT-3 code)	2 pairs of CAT 5 UTP or Type 1 STP, one pair for TX and the other pair for RX	Full or Half	802.3 clause 25
1000BASE-T* (Auto-negotiation always enabled)	125 MBd	4D-PAM5(each octet->4 quinary symbols {-2, -1, 0, +1, +2}. The 4 PAM5 symbols are sent simultaneously over four wire pairs.)	4 pairs of CAT 5 UTP	Full or Half	802.3 clause 40
1000BASE-CX (Auto-negotiation always enabled)	125 MBd	8B/10B (each octet->10 bit binary code)	2 pairs of 150-ohm STP, one for TX, the other for RX	Full or Half	802.3 clause 39

One Baud (Bd) = One transmitted symbol per second
STP = Shield Twisted Pair
UTP = Unshielded Twisted Pair

Table 1: Common Ethernet Technologies for the Copper Medium

Among these Ethernet versions, not just the speeds are different, but also the cabling, duplex mode and the underlying encoding of the data bits are different. Each data octet or nibble is encoded in a symbol, and the symbol may contain more than one binary bit, hence the baud rate is distinct from the transmitted bit rate. Data encoding has long been used to provide clock recovery; bits recognition; error detection/correction and to maintain the DC balance of the medium.

One amazing thing with the current PHY chips is that a typical PHY today can support almost all of these Ethernet versions through proper register configurations. This is a lot of technologies integrated into the same chip!

The faster the baud, the harder to sniff

There are a few objectives to achieve when tapping an Ethernet copper connection.

1. The tap should not change the electrical characteristic of the tapped cable. It should avoid drawing too much power from the cable and it should minimize changing the impedance of the cable.
2. The tap should not generate errors back to the tapped line.
3. The tap must receive a clean copy of the tapped traffic.
4. The tap must be able to receive traffic from both directions in a full-duplex connection.
5. The tapped line must not be affected in case the tap fails.
6. The tap should have high performance and low cost.

Note that we assume the user is performing legal tapping in this discussion. The user is the network operator wants to monitor and maintain his or her own network. We do not address the likelihood that a tap can be discovered by an external source. The latter belongs to a totally different profession beyond the knowledge of the author.

Given that all Fast Ethernet and Gigabit Ethernet traffic are encoded in symbols, it makes sense to put PHY chips on the taps so that the tapped traffic terminates at the PHY chips. The PHY chips then decode the tapped traffic back to the original data octets, which can then be re-generated to feed the back end monitoring or security tools.

Generally speaking, there are 3 ways to tap an Ethernet copper connection.

The first way is through direct connections to the wires of the tapped cable, as shown in figure 1.

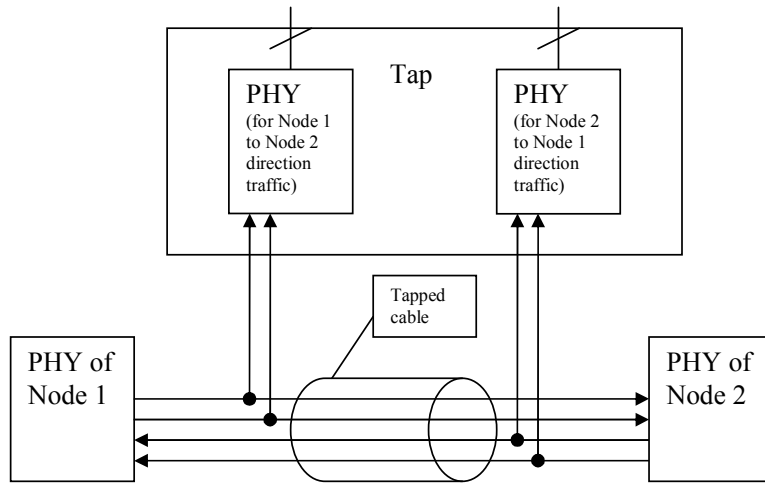


Figure 1: Tapping through direct wire connections

The advantage of this method is that the tapped connection is not affected even if the tap fails. The disadvantage of this method is that the tap interferes electrically with the tapped cable. It introduces changes to the impedance and capacitance of the cable and also draws away power from the tapped source. This problem gets worse as the line speed increases. For Gigabit Ethernet, this method imposes challenges in properly terminating the connections so they do not create reflections, distortion of signals and generation of crosstalk, yet simultaneously trying to minimize the amount of power drawn from the wires being tapped.

The second tapping method is through inductive tapping, as shown in figure 2.

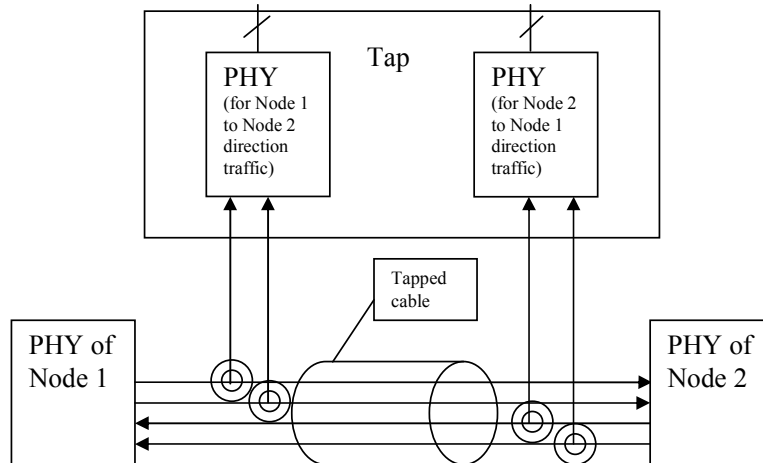


Figure 2: Inductive tapping

Inductive tapping is based on picking up the magnetic field generated by the current flowing through each cable wire. Through a solenoid coil, the magnetic field is re-converted back to electrical signals and sent to the tap.

Theoretically, this method has the advantage that there is no direct connection between the tap and the wires being tapped. In practice, one cannot tap directly on a pair of twisted pair wires because the magnetic fields of the twisted pair wires cancel out each other. If we separate out the twisted pair wires, then we introduce changes in impedance to the cable. At high line speed, the magnetic fields from each wire can interfere with each other through mutual induction. Also, the re-generated digital pulses will carry a lot of noise that it may not give a true representation of the original traffic. Integrating all these induction coils into a compact copper tap presents another challenge.

The third method is through active tapping with relays for fail-over protection, as shown in figure 3.

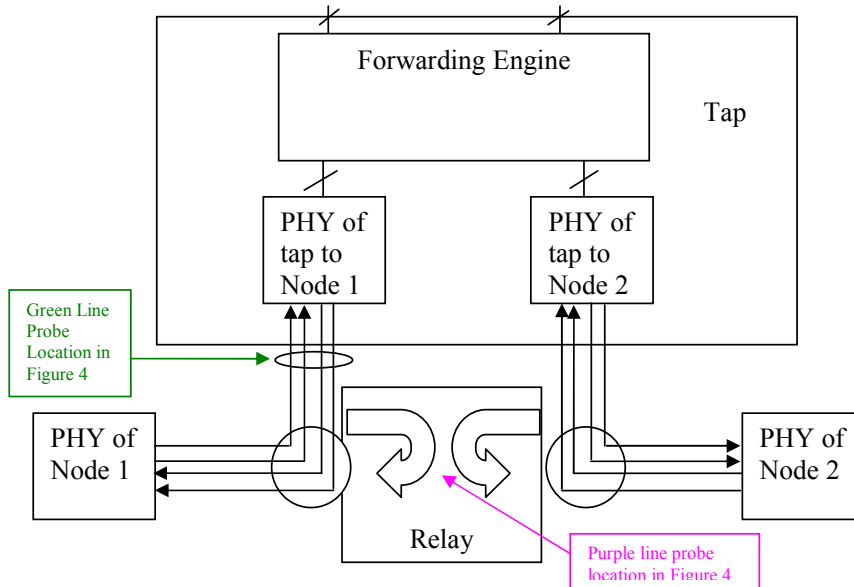


Figure 3: Active relay tapping

Active relay tapping terminates the connection of each node to the tap. The tap then uses a forwarding engine to forward the traffic from the left hand connection to the right hand and vice versa, and meanwhile it creates copies of the same traffic and send them to the backend monitoring and security tools.

The disadvantage of this method is that the forwarding engine is an active electronic component that may fail over time or the tap may fail if electricity goes out. The role of the relay is to rapidly close the circuit of the connection being monitored in case the tap fails, such that node 1 connects to node 2 as if there is simply a straight wire between them. The end result is that the tapped connection is rapidly resumed if the active tap fails.

This method turns out to be the best for tapping Gigabit Ethernet traffic. It does not involve changing the impedance along the cable. From the system standpoint, this is as if there are two separate standard-compliance Ethernet connections: one from node 1 to the left PHY of the tap; another from node 2 to the right PHY of the tap. This method also works well for 10Mbps and 100 Mbps Ethernet connections.

Switch-over behavior of an active relay copper tap

Typically, there are 8 relays in an active Ethernet copper tap. Each relay handles one of the 8 wires within a CAT 5 cable. The switch-over time of a relay is about 0.5 millisecond, as shown in figure 4.

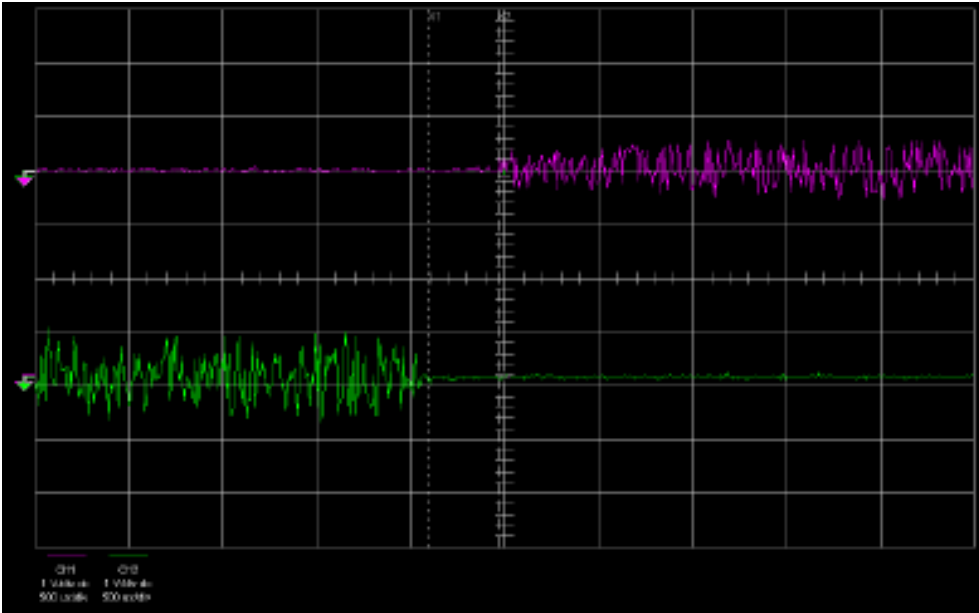


Figure 4: Oscilloscope measurements of the switch-over time of a relay

In figure 4, each horizontal division represents 500 micro-seconds. Before switching from active mode to passive mode, traffic flows from node 1 to the tap, then through the forwarding engine to node 2, and vice versa in the other direction. Therefore signals are seen on the green line for the first 4 time divisions. At the 4.25th division, relay switch over occurs, and therefore no more traffic goes to the PHY of the tap. The green line suddenly becomes quiet. Then at about the beginning of the 5th time division, the purple line suddenly sees signals. This means that physical connection between node 1 and node 2 is now established. Hence the switch-over time before physical connection is re-established is about 1 time division, or 0.5 millisecond.

During a relay switch over, although the physical connection is re-established within half a millisecond, the link between node 1 and node sometimes may not be re-established until 2 to 3 seconds later. Other times the link may be re-established right after physical connection is re-established.

We monitored the trace behavior during such situations and found out that, if the link LED goes off and comes back on in 2-3 seconds, the purple signal, which is seen 0.5 millisecond after the relay switch over, suddenly disappears and stays disappeared for 2-3

seconds and then comes back out again. At this moment the link LED is also up. In the case where the link LED's stay solid at the end nodes, we observe that the signal at the purple line stays unchanged. Note that to re-establish a link, the PHY has to go through a state machine exchanging a number of control code groups. The device driver software that periodically polls the PHY for link status may have sampled the link down event and may then reset the PHY and/or MAC. The time to come out of a PHY or MAC reset is usually much longer, in the order of 1 or 2 seconds. Hence this can contribute to the perceived 2-3 seconds overall delay because both ends may have their PHY or MAC modules resetting.

No free lunch

Technically, there are more challenges to come up with a passive copper tap that works at Gigabit Ethernet speed or higher. Using an active relay copper tap is the current best choice, although there may be occasions where the link can be down for 2-3 seconds during a relay switch-over. This has to do with the process of re-establishing link between the end nodes. The tap relay typically re-establishes physical connection within a millisecond or less.

Deleted: If there is such a solution, the cost of manufacturing it will likely be very high because copper taps are not high volume commodities.

Deleted: second