鏈路層發現協議([LLDP](http://baike.baidu.com/view/1930049.htm))是一個廠商無關的二層協定，它允許網路設備在本地[子網](http://baike.baidu.com/view/65511.htm" \t "_blank)中通告自己的設備標識和性能。

## [編輯本段](http://baike.baidu.com/view/1930049.htm)產生背景

目前，網路設備的種類日益繁多且各自的配置錯綜複雜，為了使不同廠商的設備能夠在網路中相互發現並交互各自的系統及配置資訊，需要有一個標準的資訊交流平臺。

LLDP (Link Layer Discovery Protocol，鏈路層發現協議)提供了一種標準的鏈路層發現方式，將本端設備的的主要能力、管理位址、設備標識、介面標識等資訊組織成不同的[TLV](http://baike.baidu.com/view/3575978.htm) (Type/Length /Value，類型/長度/值)，並封裝在LLDPDU (Link Layer Discovery Protocol Data Unit，鏈路層發現協定資料單元)中發佈給與自己直連的鄰居，鄰居收到資訊後將其以標準MIB (Management Information Base，管理資訊庫)的形式保存起來，以供[網路管理系統](http://baike.baidu.com/view/2523209.htm" \t "_blank)查詢及判斷鏈路的通信狀況。

## [編輯本段](http://baike.baidu.com/view/1930049.htm)簡介

鏈路層發現協議(Link Layer Discovery Protocol)--IEEE802.1ab

The Link Layer Discovery Protocol or LLDP is a vendor-neutral Layer 2 protocol that allows a network device to advertise its identity and capabilities on the local network. The protocol was formally ratified as IEEE standard 802.1AB-2005 in May 2005. It supersedes proprietary protocols like Cisco Discovery Protocol, Extreme Discovery Protocol and Nortel Discovery Protocol (also known as SONMP).

簡單說來，LLDP是一種鄰近發現協議。它為乙太網網路設備，如交換機、路由器和[無線局域網](http://baike.baidu.com/view/5470.htm" \t "_blank)接入點定義了一種標準的方法，使其可以向網路中其他節點公告自身的存在，並保存各個鄰近設備的發現資訊。例如設備配置和設備識別等詳細資訊都可以用該協定進行公告。具體來說，LLDP定義了一個通用公告資訊集、一個傳輸公告的協定和一種用來存儲所收到的公告資訊的方法。要公告自身資訊的設備可以將多條公告資訊放在一個局域網資料包內傳輸，傳輸的形式為類型長度值(TLV)域。

## [編輯本段](http://baike.baidu.com/view/1930049.htm)工作模式

TxRx：既發送也接收LLDP[報文](http://baike.baidu.com/view/175122.htm)。

Tx：只發送不接收LLDP[報文](http://baike.baidu.com/view/175122.htm)。

Rx：只接收不發送LLDP[報文](http://baike.baidu.com/view/175122.htm)。

Disable：既不發送也不接收LLDP[報文](http://baike.baidu.com/view/175122.htm)。

當埠的LLDP工作模式發生變化時，埠將對協議[狀態機](http://baike.baidu.com/view/1906565.htm" \t "_blank)進行初始化操作。為了避免埠工作模式頻繁改變而導致埠不斷執行初始化操作，可配置埠初始化延遲時間，當埠工作模式改變時延遲一段時間再執行初始化操作。

## [編輯本段](http://baike.baidu.com/view/1930049.htm)LLDP報文

### 概念

封裝有LLDPDU的[報文](http://baike.baidu.com/view/175122.htm" \t "_blank)稱為LLDP報文，其封裝格式有兩種：Ethernet II和SNAP（Subnetwork Access Protocol，[子網](http://baike.baidu.com/view/65511.htm" \t "_blank)訪問協議）。

### 發送機制

當埠工作在TxRx或Tx模式時，設備會週期性地向鄰居設備發送LLDP報文。如果設備的本地配置發生變化則立即發送LLDP[報文](http://baike.baidu.com/view/175122.htm)，以將本地資訊的變化情況儘快通知給鄰居設備。但為了防止本地資訊的頻繁變化而引起LLDP[報文](http://baike.baidu.com/view/175122.htm)的大量發送，每發送一個LLDP報文後都需延遲一段時間後再繼續發送下一個報文。

當設備的工作模式由Disable/Rx切換為TxRx/Tx，或者發現了新的鄰居設備(即收到一個新的LLDP[報文](http://baike.baidu.com/view/175122.htm)且本地尚未保存發送該報文設備的資訊)時，該設備將自動啟用快速發送機制，即將LLDP報文的發送週期縮短為1秒，並連續發送指定數量的LLDP報文後再恢復為正常的發送週期。

### 接收機制

當埠工作在TxRx或Rx模式時，設備會對收到的LLDP[報文](http://baike.baidu.com/view/175122.htm)及其攜帶的TLV進行有效性檢查，通過檢查後再將鄰居資訊保存到本地，並根據TTL(Time To Live，生存時間) TLV中TTL的值來設置鄰居資訊在本地設備上的老化時間，若該值為零，則立刻老化該鄰居資訊。

# Link Layer Discovery Protocol (LLDP, IEEE 802.1AB)

The Link Layer Discovery Protocol (LLDP) is a vendor neutral layer 2 protocol that can be used by a station attached to a specific LAN segment to advertise its identity and capabilities and to also receive same from a physically adjacent layer 2 peer.

## Protocol dependencies

* [Ethernet](http://wiki.wireshark.org/Ethernet): Typically, LLDP uses [Ethernet](http://wiki.wireshark.org/Ethernet) as its "transport" protocol. The Ethernet type for LLDP is 0x88cc.
* Other IEEE 802 networks: LLDP can also use other 802 networks as a "transport" protocol, with a SNAP header with an Ethernet type of 0x88cc.

LLDP Data Units (LLDPDUs) are sent to the destination MAC address 01:80:c2:00:00:0e. This address is defined as the "LLDP\_Multicast" address. This address is defined within a range of addresses reserved by the IEEE for protocols that are to be constrained to an individual LAN. LLDPDUs can be directly encoded with an Ethertype value of 0x88cc or they may be encapsulated within an SNAP-encoded LLC frame.

## Wireshark

Support for LLDP (and the TIA's LLDP-MED extensions) is available since Wireshark 0.10.13.

## Display Filter

To display only the LLDP based traffic use: lldp

## Capture Filter

To capture only the LLDP based traffic use: ether proto 0x88cc