# White Paper on the subject of the AC coupled diode discovery method

adapted from a class project:
Santa Clara University, COEN233 Computer Networks, July 2000, Rick Brooks

**Abstract:**

The electrical power used by computing and networking devices is most often taken for granted; we just assume that the power is working properly, and we don't really care how it works. There is, however, a large class of low power network appliances and devices for which it is either too expensive or too awkward to physically hook up power in the conventional way. Still other remote devices are in need of multiple and redundant power sources to improve communication uptime and reliability. Furthermore, devices such as IP telephones would be simpler if they needed only a single cable for data and power. A method is proposed for the discovery of DTE "powerable" devices through a unique digital signature. Powerable, or powered devices are designed to take power off of the UTP data cable, allowing both power and data to use the same cable. A discovery process, located in the Power Sourcing Equipment, looks for a unique identity network, located in the DTE, or Powered Device at the other end of the link. Once the discovery process is successful, the application of power on the UTP data cable is possible and generally safe. Optionally, network management could be used for the control of large numbers of these types of powered links, therefore allowing for the provisioning of powered devices. This paper concentrates on the data link layer implementation of the discovery and power control processes, with some references to the expected physical layer interfaces.

**Outline of a Solution at the Top Level:**

A high level decision diagram needs to be developed that describes what tests need to be made and what actions need to be taken on a link that employs smart power. When the PSE first powers up, it must go into an initialization, or safe state, where both power and discovery processes are not active. If commanded by network management, the discovery process is initiated which sends out very small amounts of power in the form of pulses, designed so that they cannot harm any connected devices. The pulses probe for the existence of the identity network in the PD. If, and only if, the discovery process has found the identity network, and if it is also enabled by the network management function, the power process can begin. Power can then be energized to the link, and power faults are continuously monitored. If there are any subsequent power faults, or if the physical link is broken and the power delivered drops below a given threshold, then the power is turned off and the system goes back into the safe state. At this point, the whole process can be repeated in the same manner.

A simplified outline of the discovery and power process is shown in Figure 3. The safe state always occurs first, followed be the discovery process, and then the power delivery process. The diagram below represents the high level flow that is implemented in the Verilog code [(14)].

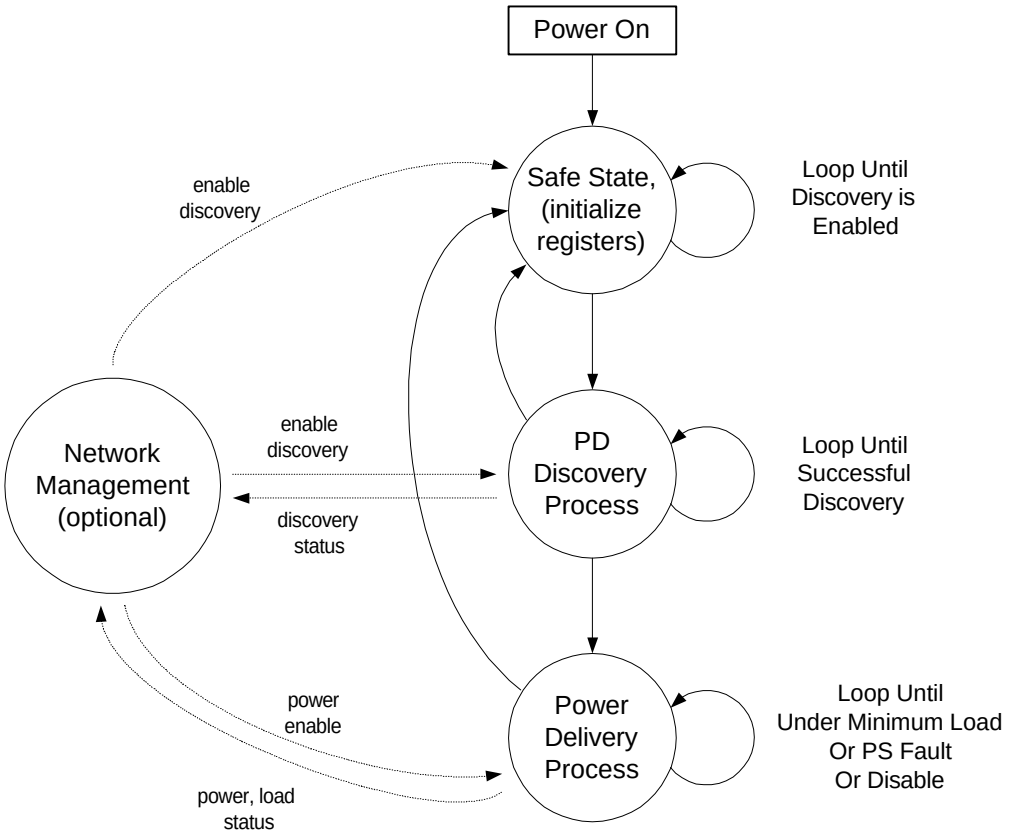


**Figure 3, Top Level Process Flow for Discovery and Power**

**Top Level Process Description:**

The chronological steps that are typically followed during the operational life of the power link are described below, and follow the process flow shown in Figure 3, above. Although the network management function may be considered optional, it is key to the control and monitoring of large deployments of LAN networks with PSE's and PD's involving large amounts of distributed power.

1) Safe State - physically safe, since both the discovery and power processes are disabled. protocol:

    a) all registers are initialized to a safe condition.
    b) the safe state is the default state, and all other states are reachable only under the control of network management, through the use of enable commands.

2) PD Discovery Process - goal: uniquely determine if a PD is present, search for the identity network protocol:

    a) monitor for "discovery enable" from network management, return to safe state if disabled
    b) create a series of discovery frames, i.e. patterns of positive and negative polarity pulses which are sent down the link to the identity network in the PD, where the idle, or off time, between the pulses is always a pseudo random length of time.
    c) count the consecutive number of successful detection frames, until a certain confidence level has been achieved, that the device on the link is indeed a PD
    d) for each unsuccessful discovery frame, that is a frame which does not detect the identity network of a PD, zero the counter of the number of successful consecutive discovery frames
    e) repeat as long as necessary for successful discovery, unless disabled by network management

3) Power Delivery Process - energize the power source and monitor for power fault conditions protocol:

    a) following successful discovery, monitor for "power enable", or "discovery/power enable" from network management
    b) turn the power on and wait for the power level to be above a minimum level
    c) monitor for power faults: "over voltage" or "over current" conditions in the power supply
    d) monitor for an open circuit of the PD - for example, the cable is unplugged
    e) continuously monitor the power enable from the network management
    f) continue in the power state until a power fault, or a link unplug or disable condition exists, in which case, go to the safe state

**The Physical Layer:**

A brief word is needed regarding the physical layer topology expected for the powered LAN network. The power sourcing equipment (PSE) has the responsibility for discovery and power. It is where the smart power process is carried out. The three basic parts of the PSE that are not found in non-powered LAN legacy devices are:

- the power supply, which is dedicated to the particular powered port
- the power delivery process, which is responsible for controlling and managing the power on that port
- the discovery process, which is responsible for detecting the identity network located in the PD, at the other end of the cable

An electrical loop is formed between the power supply output capacitor, the discovery transformers, the data cable, and the identity network, as shown in Figure 4. The discovery process stimulates current into this loop through the use of transformer coupling as shown. The first transformer transmits a positive pulse, while at the same time, the second transformer receives a signal proportional to the loop current. That signal is synchronously detected. Next the second transformer transmits a negative pulse, while the first transformer receives a signal proportional to the loop current. That signal is also synchronously detected.

As shown in Figure 4, the identity network is located in the PD, at the other end of the physical link. The load in the PD is designed not to interact with the discovery process, and appears as an open circuit. However, the identity network responds differently to a positive pulse compared to a negative pulse. It is this difference that we can detect.

The power delivery process interfaces to the power supply in the PSE, as shown. The power supply is off during the discovery process, and only energizes during the power delivery process.

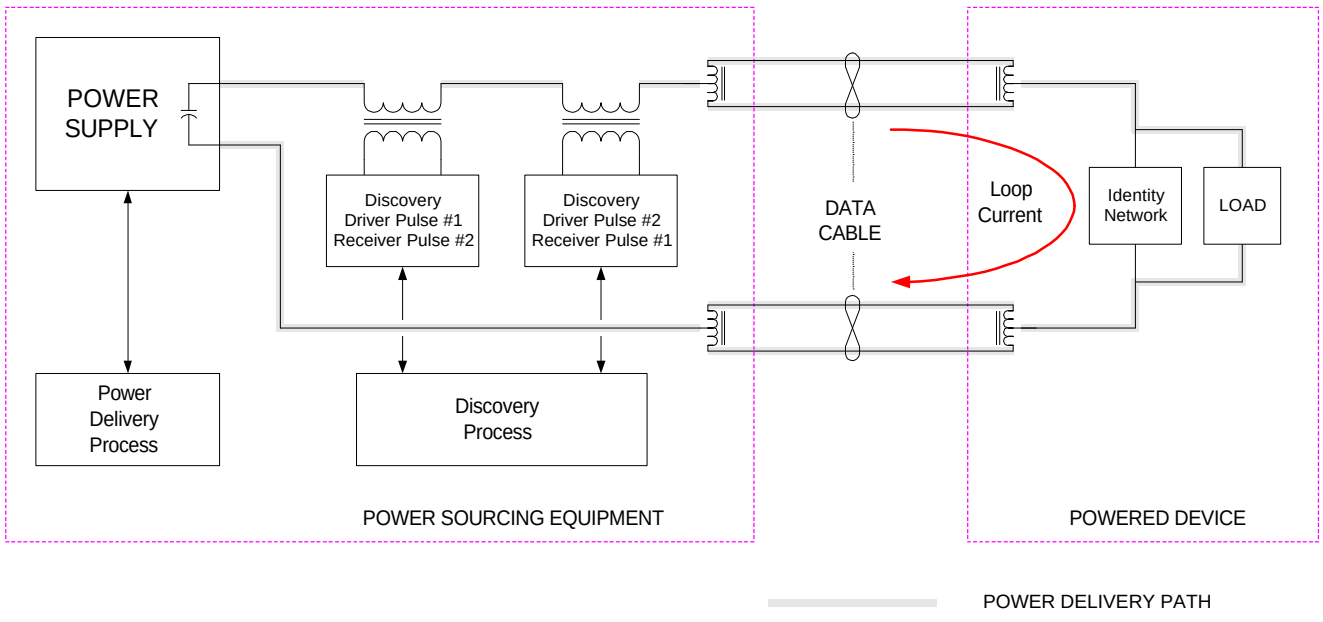


**Figure 4, Block Diagram of the Physical Layer Topology**

**The Discovery Process:**

The discovery process is executed at the data link layer, and it is responsible for communicating, through the physical layer, with the PD, if such a device is connected. The discovery process is independent of actual data delivery, since data can be sent and received if both ends of the link already are powered. The discovery process could be implemented in hardware or in software. I describe herein a hardware implementation in a programmable logic device, or PLD. The PLD code is written in Verilog in this case. The proposed code can be found at the hyperlink, reference [14]. The finite state machines shown in the above figures represent a simplified version of the behavior of this Verilog code.

- Identity Network located in the PD

The identity network proposed herein is implemented as a low cost passive network in the physical layer of the PD. Future implementations could be designed by using a micro-powered circuit capable of responding in multiple secondary ways, instead of just with the message: "I am here". However, for the purposes of this paper, I describe the simplest form of the identity network, which, if "discovered", merely says that: "I am here".

The proposed passive identity network is shown in Figure 5: in part a) as a block diagram, and in part b) as an actual physical layer schematic. The identity network passes pulsed current in one direction, and does not pass pulsed current in the other direction. This basic difference allows the digital techniques of the discovery process to provide for a means of discovery. One key difference between this network and a simple diode is that this network can be stimulated, by using high duty cycle discovery pulses, into behaving as an open circuit. Whereas, by using the normal low duty cycle pulses, it behaves as a diode. By utilizing this feature, the PSE discovery process can both detect and verify that the device on the physical layer is indeed the device it is looking for. The goal here is to increase the confidence level of the discovery process, leading to a system that is more robust overall. We don't want a system that falsely supplies power to legacy, non-qualified devices.


**a)** **b)**

**Figure 5, proposed PD identity network: a) block diagram form, b) schematic form**

- The Discovery Frame

Part of the discovery protocol is the creation of a set of discovery frames. These frames are generated by the PSE and contain timing that is under the sole control of the discovery finite state machine. The process does not wait for any message to return; it just keeps running. The protocol is therefore automatically and deterministically live. The discovery frame caries out a form of synchronous detection, where the response of the identity network to the stimulus is recorded. The protocol specifies that the discovery frames be repeated, and that there must be a minimum number of consecutive successful discovery frames before the discovery process is deemed successful. The requirement here is arbitrarily set at 256 consecutive successful discovery frames to help insure a more robust system. Each discovery frame consists of five distinct events, or states:

1) a positive going discovery pulse state, which performs the first part of synchronous detection consisting of:
    - ramp up state
    - sustained level state
    - ramp down state
2) a idle state #1 in which there is no loop current flow, and whose time duration is of a pseudo random length, and constrained to be within certain limits.
3) a negative going discovery pulse state, which performs the second part of synchronous detection consisting of:
    - ramp up state
    - sustained level state
    - ramp down state
4) another idle state (idle state #2) in which there is no current flow, and whose time duration is a new pseudo random length, also constrained to be within certain limits.
5) the handoff state in which the results of the two parts of synchronous data detection are compiled and compared against the criteria for successful discovery: 256 consecutive successful frames.

The single discovery frame is designed to have a pseudo random duration, since both the idle state, #1 and #2 are of pseudo random duration. This form results in a more robust system when compared with the use of fixed timing. For example, in the case where two such PSE devices are connected together by mistake, the two will not interact and cause a false detection, within any reasonable probability. Refer to Figure 6 for a time domain view of the induced loop EMF of a single discovery frame.

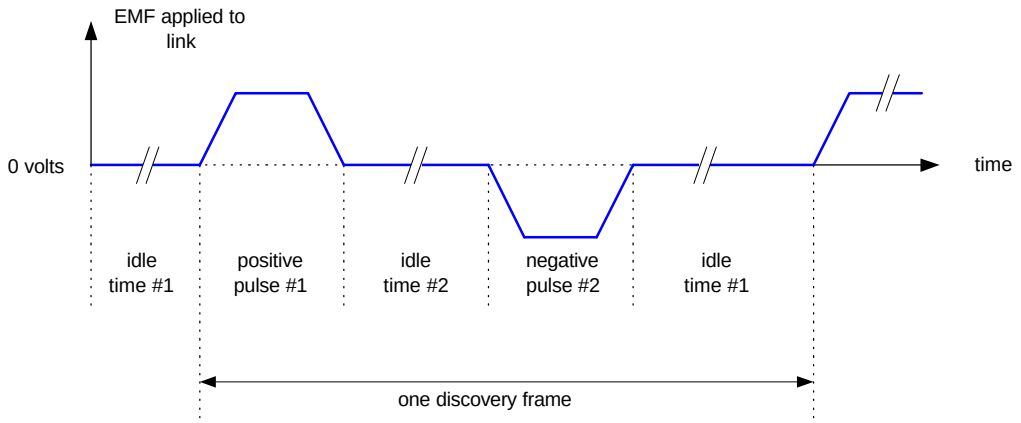


**Figure 6, a single discovery frame in the time domain**

**The Discovery Frame Finite State Machine:**

The discovery process FSM, built in the Verilog code, is shown in simplified form in Figure 7, below. The default and safe state is called “start”. In the “start” state, all of the registers are initialized to be safe, so the discovery and the power process state machines are not active. When the network management enables the process by setting the bit “enable_disc” true, then the discovery process begins and we enter the “idlephase1” state. The bit “enable_disc” is checked at each subsequent state in the discovery frame FSM, and if “enable_disc” remains true, then the discovery process continues, otherwise the FSM goes to the “start” state.

In the “idlephase1” state, the 1st idle part of the discovery frame is generated. During this period, there are no discovery pulses generated. The “idlephase1” state calls up the general purpose timer by setting the bit "idletimerstart" true. The timer then grabs a pseudo random code from the “rand_idletime[11:0]” register, which is a 12 bit snapshot of the pseudo random number generator, described later. The “rand_idletime” register contains a number, which apart from being a pseudo random number, is also guaranteed to be greater than 100000000 binary (256 decimal) clock cycles, or about 40.96 µs. This need comes from the electrical nature of the identity network that requires a minimum length of idle time in order to recover it's operating point between pulses.

Snippet of Verilog code below that defines rand_idletime:

```
rand_idletime[11:0] =       { ds[0] , ds[1] , ds[2] , 1'b1 , ds[3] , ds[4] , ds[5] , ds[6] , ds[7] , ds[8] , ds[9] , ds[10]};
```

When the general purpose timer is done, it sets the bit “timerdone” true, then the "idlephase1" state sets the “idletimerstart” bit false, and then we jump into the “waitphase1” state.

The state “waitphase1” calls the pulse #1 state machine by setting the bit “ramppulsego1” true. The pulse #1 state machine, which is described later, is responsible for generating the positive discovery pulses, also called the odd numbered pulses. Once the pulse #1 state machine is through generating the positive pulse, it then sets the bit “ramppulsedone1” true, the “waitphase1” state sets “ramppulsego1” bit false, and we proceed.

The next state is called “idlephase2”. This state works the same way that the “idlephase1” state worked, as described above. It should be noted that a new pseudo random timer value is selected each time either the “idlephase1” or the “idlephase2” states occur. Again, when the timer is done, the next state is “waitphase2”.

The state “waitphase2” calls the pulse #2 state machine, which is described later, and is responsible for generating the negative discovery pulses, also called the even numbered pulses. This state works in an analogous way compared with the “waitphase1” state described above. Again, once the pulse #2 state machine is through generating the negative pulse, it then sets the bit “ramppulsedone2” true, and the “waitphase1” state sets the bit “ramppulsego2” false, and we proceed. The next state is called the “handoffphase” state.

The “handoffphase” state is where a determination is made whether the discovery process has been successful or not. The results of the synchronous detection process, which is described later, is checked against the requirement for a certain number of consecutive successful discovery frames. If a discovery frame was successful, and the number of consecutive counts is now satisfied, then we go to the power process in which the DTE power is energized. The power process is described below. If the latest discovery frame was successful, but the required number of consecutive counts is not yet high enough (recall that we need a count of 256) then the appropriate success counter is incremented, and the FSM continues back to “idlephase1” for another discovery frame to start again. If, on the other hand the latest discovery frame has not been successful in detecting the identity network, then the counters for counting consecutive successful detections are set to zero, and the FSM continues back to “idlephase1” for the discovery frame to start again, essentially from the beginning.

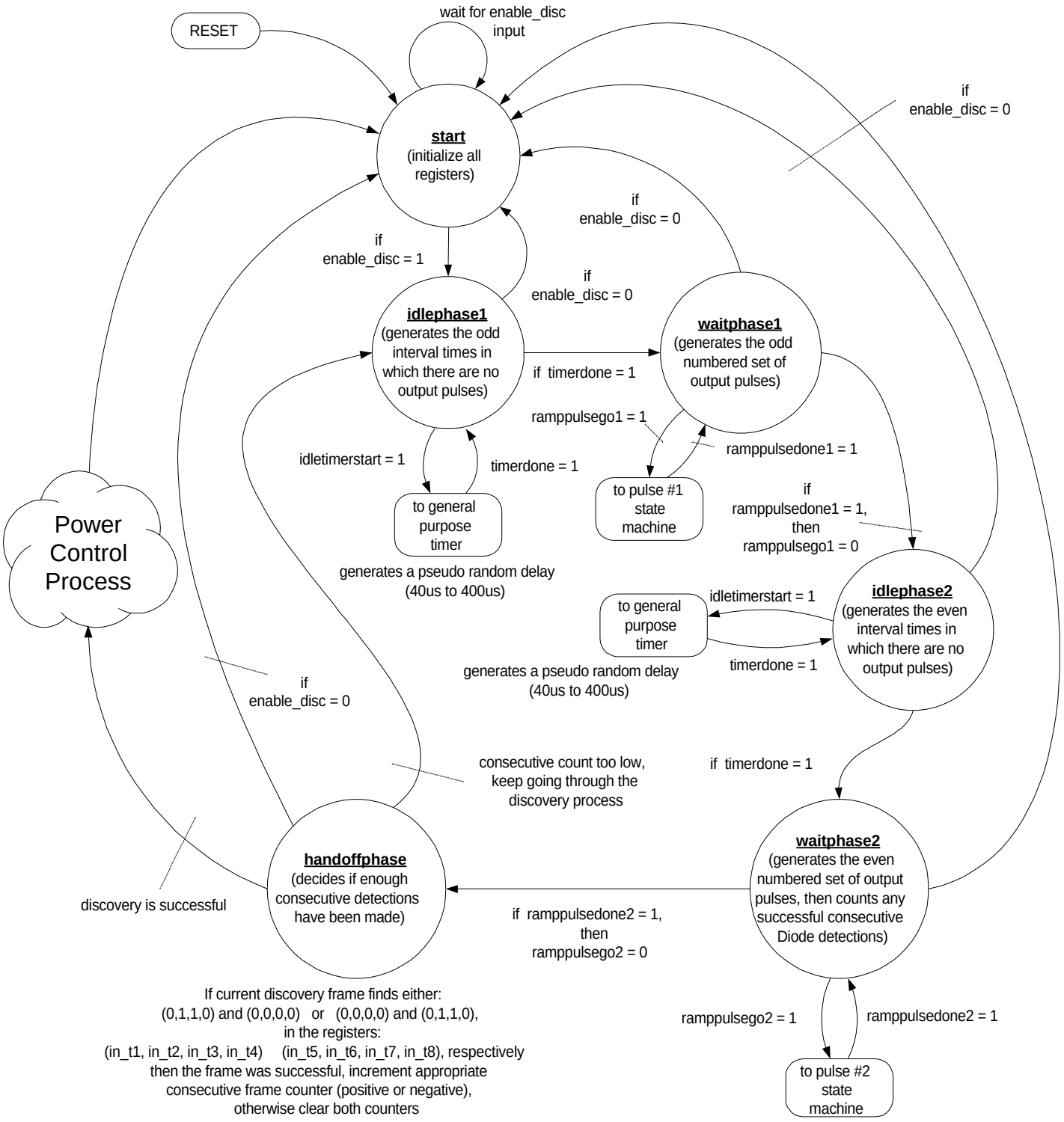


**Figure 7, Discovery Process, Top Level Finite State Machine**

- The Pulse #1 Finite State Machine

The pulse #1 FSM normally is in the “no_pulse1” state, which is the default state, and no outputs are driven and no discovery pulse is produced. The pulse #1 FSM waits for a bit from the main discovery FSM, called “ramppulsego1”, to be true. When that bit is set, then the pulse #1 FSM proceeds to the “rampup1” state.

The “rampup1” state produces a shaped rise time of the discovery pulse by sequencing the 8 outputs, called drv1a to drv1h, from off to on, at separate times. The external hardware sums and filters these 8 signals to produce a shaped transmit pulse. This shaping mechanism is required in order to translate the 3 ns rise and fall times of a PLD into the 3 µs rise and fall times needed in this application, which minimizes the required channel bandwidth for discovery. After the “rampup1” state, the FSM goes to the “pulse1” state.

The “pulse1” state simply holds all of the 8 outputs, drv1a to drv1h, in an active state. Following a fixed delay, the FSM proceeds to the “rampdown1” state.

The “rampdown1” state is similar to the “rampup1” state, except that it sequences the 8 outputs, drv1a to drv1h, from on to off. Therefore the discovery pulse that is created ramps back down to zero in a well behaved, "shaped" manner. The bit “ramppulsedone1” is set true, and the FSM proceeds to the “wait1” state. The “wait1” state simply waits a short period until the bit “ramppulsego1” has been set to false by the main discovery FSM, then goes back to the “no_pulse1” state.

During any of the above steps, the pulse #1 FSM will go directly back to the default state “no_pulse1” if the bit "enable_disc" is set false by network management.

The timing of the pulse #1 FSM is deterministic, therefore the process is live. Refer to Figure 8 for the Pulse #1 Finite State Machine diagram.

- The Pulse #2 Finite State Machine

The Pulse #2 FSM, is very similar to the Pulse #1 FSM, except that it generates negative polarity pulses into the identity network, rather than positive pulses.

The pulse #2 FSM normally is in the “no_pulse2” state, which is the default state; no outputs are driven, and no discovery pulse is produced. The pulse #2 FSM waits for a bit from the main discovery FSM, called “ramppulsego2”, to be true. When that bit is set, then the pulse #2 FSM proceeds to the “rampup2” state.

The “rampup2” state produces a shaped rise time of the discovery pulse by sequencing the 8 outputs, called drv2a to drv2h, from off to on, at separate times. The external hardware sums and filters these 8 signals to produce a shaped transmit pulse. Again, this shaping mechanism is required in order to create the 3 µs rise and fall times needed in this application. After the “rampup2” state, the FSM goes to the “pulse2” state.

The “pulse2” state simply holds all of the 8 outputs, drv2a to drv2h, in an active state. Following a fixed delay, the FSM proceeds to the “rampdown2” state.

The “rampdown2” state is similar to the “rampup2” state, except that it sequences the 8 outputs, drv2a to drv2h, from on to off. Therefore, the discovery pulse that is created ramps back down to zero in a well behaved, "shaped" manner. The bit “ramppulsedone2” is set true, and the FSM proceeds to the “wait2” state. The “wait2” state simply waits a short period until the bit “ramppulsego2” has been set to false by the main discovery FSM, then goes back to the “no_pulse2” state.

During any of the above steps, the pulse #2 FSM will go directly back to the default state “no_pulse2” if the bit "enable_disc" is set false by network management.

The timing of the pulse #2 FSM is deterministic, therefore the process is live. Refer to Figure 9 for the Pulse #2 Finite State Machine diagram.

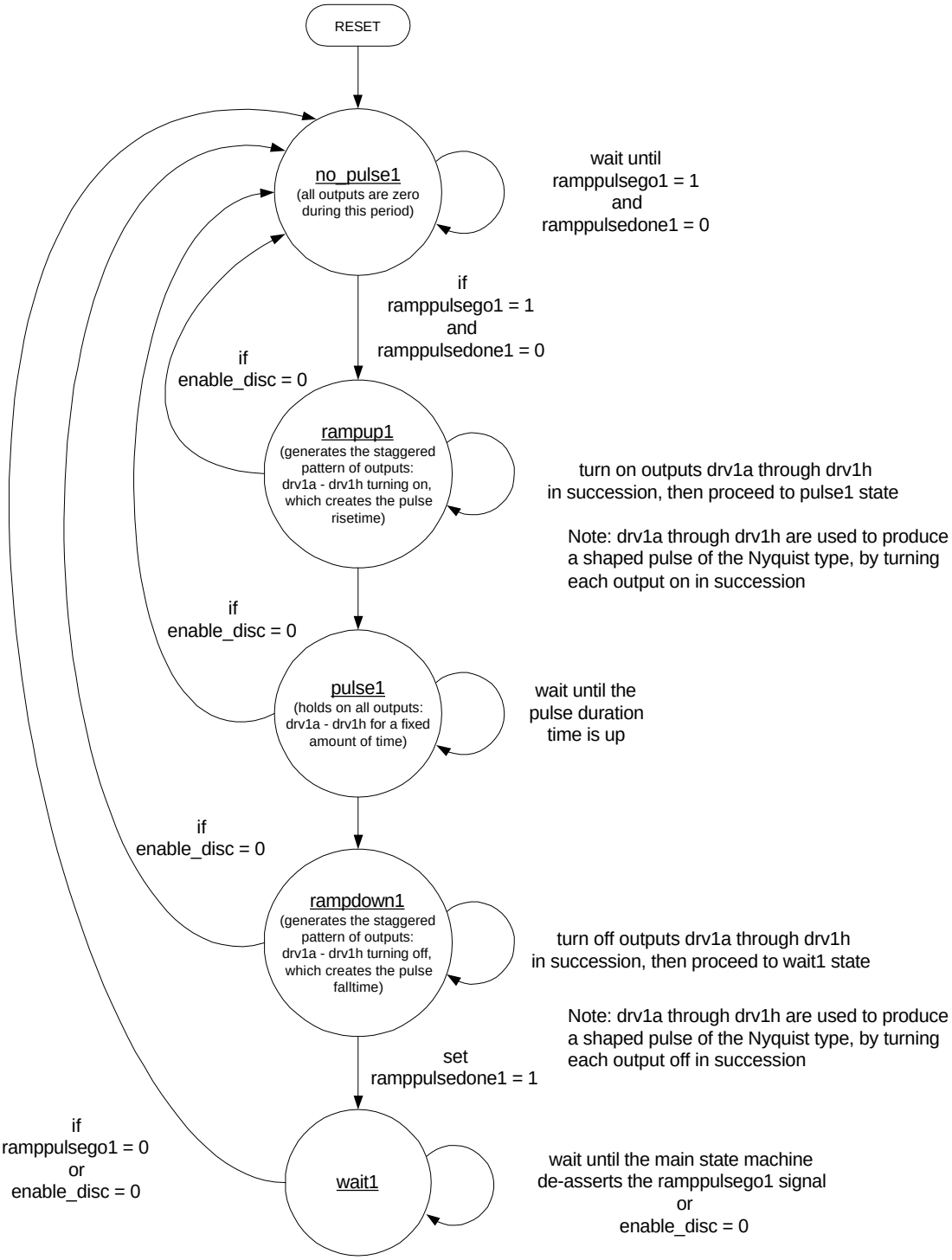


**Figure 8, Pulse #1 Finite State Machine, containing synchronous detection**

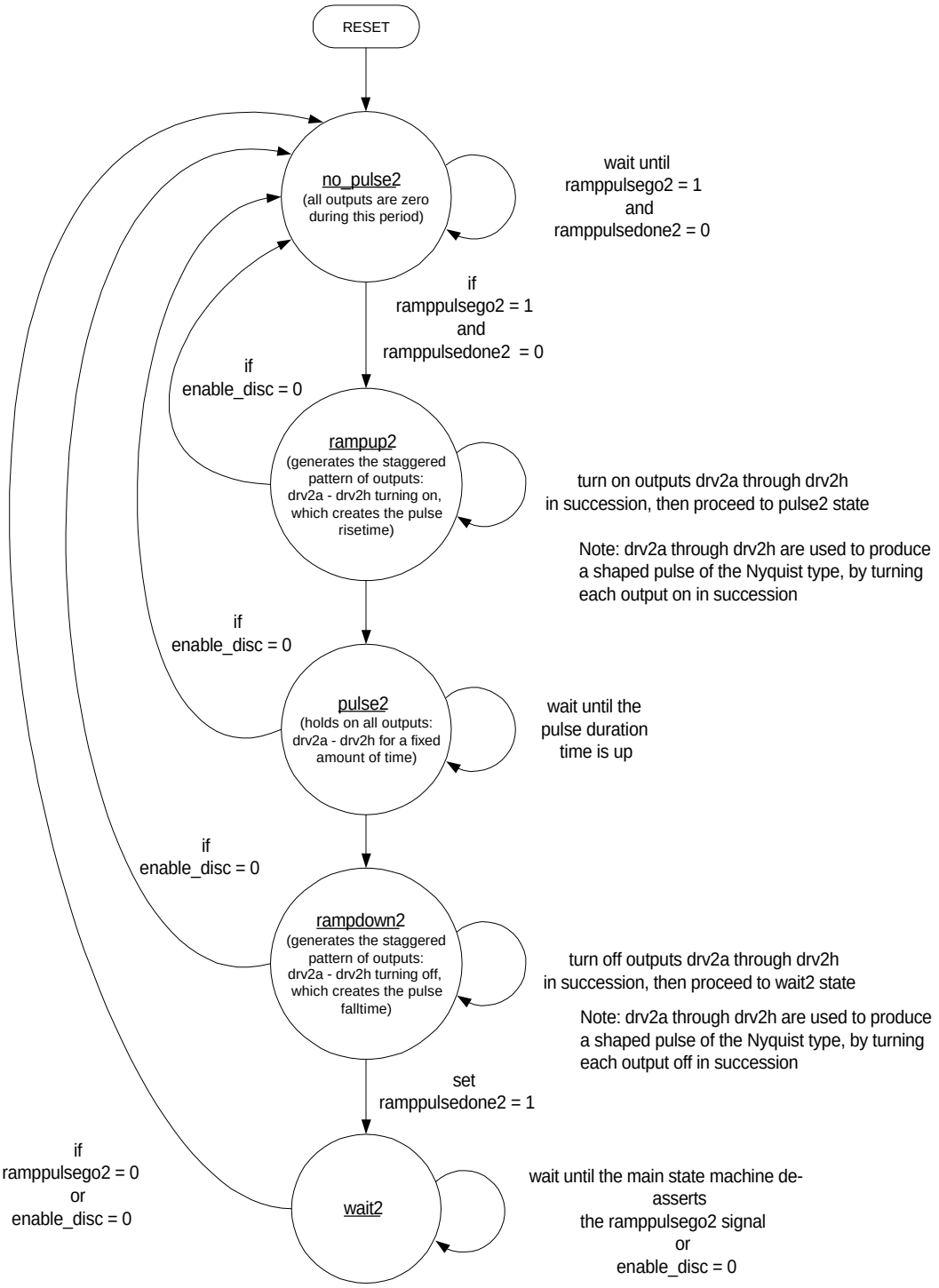


**Figure 9, Pulse #2 Finite State Machine, containing synchronous detection**

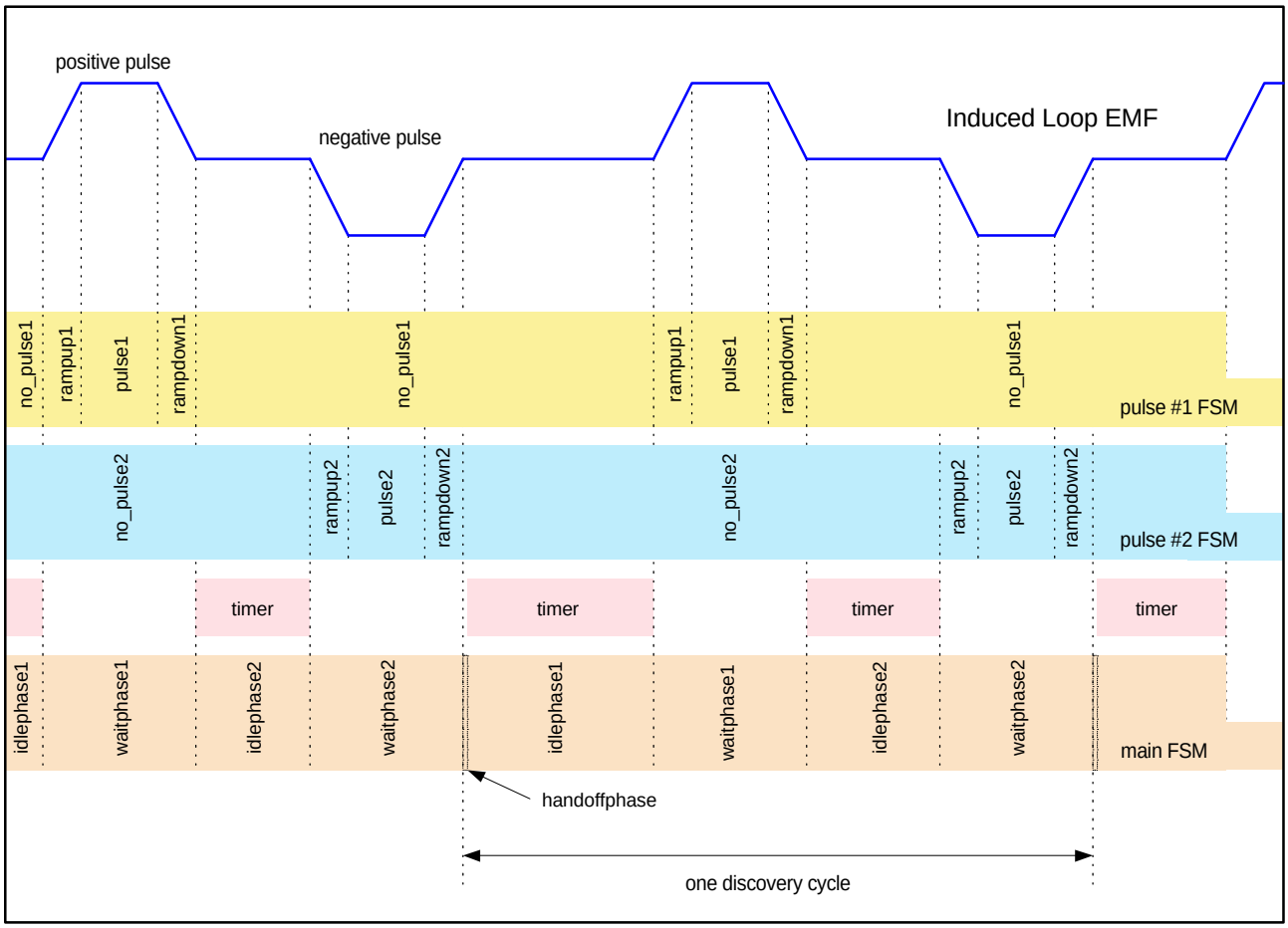


**Figure 10, Discovery Process, Finite State Machine Timing**

- Composite State Machine Timing

Figure 10, above, shows how the three finite state machines (top level discovery FSM, pulse #1 FSM, and pulse #2 FSM), and the one general purpose timer work as a team. The top level discovery, also called the main state machine, is the master; it calls the general purpose timer, the pulse #1 FSM, and the pulse #2, one at a time, and in the correct sequence. Note that the "handoffphase" state occurs at the end of each discovery frame in the top level, or main state machine. The "handoffphase" has a very short duration; only one or two clock cycles.

- Synchronous Detection

The synchronous method of detection is used to test for the existence on the identity network at the far end of the link. During the positive and negative pulse states, which are part of the pulse #1 and pulse #2 finite state machines described above, a template for a detected response is formed. Refer to Figure 11, below, to see where the synchronous detection points are located compared with the transmitted discovery pulses. There are 8 synchronous detection points for each discovery frame, 4 within the positive pulse (pulse #1 FSM), and 4 within the negative pulse (pulse #2 FSM). At each of these 8 points, a separate logical register is assigned to detect the loop response to the induced pulse. These registers are named as follows: during the positive pulse, the register names are: in_t1, in_t2,

in_t3, and in_t4. The names are ordered in time, with the register in_t1 capturing the first response in the pulse. During the negative pulse, the registers are named, also in chronological order: in_t5, in_t6, in_t7, and in_t8.

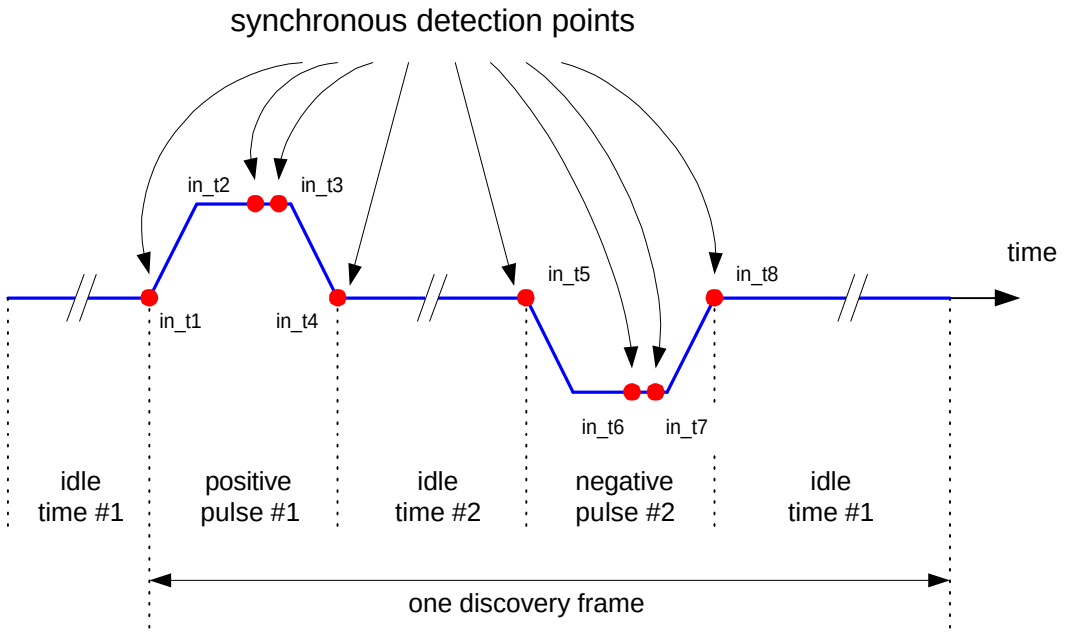


**Figure 11, synchronous detection timing**

If loop current is flowing in the physical layer through the identity network in response to the discovery pulse, then the identity diode is conducting, and the received data should be detected as follows. First, a low current (a "0") at the start of the pulse during the ramp up state, followed by a high current (a "1") at two points near the end of the sustained level state. Finally, a low current (a "0") at the end of the pulse during the ramp down state. The synchronous detected data is therefore (0,1,1,0) in this case, see Figure 12.

If little or no loop current is flowing in the physical layer through the identity network in response to the discovery pulse, then the identity diode is not conducting, and the received data should be detected as follows. First a low current (a "0") at the start of the pulse during the ramp up state, followed by a low current (a "0") at two points near the end of the sustained level state. Finally, a low current (a "0") at the end of the pulse during the ramp down state. The synchronous detected data is therefore (0,0,0,0) in this case, see Figure 12.

During a single discovery frame, a proper identity network is considered "detected" only if current flows during one, and only one, of the two discovery pulses: in other words, either in response to the positive pulse and not the negative pulse, or in response to the negative pulse and not the positive pulse. Only in either, but not both, of these two cases is the discovery frame considered a success.

If an open circuit is present in the physical layer, then there will be a "non-conducting" response for both the positive or the negative pulses, and the synchronous detected data would therefore be (0,0,0,0) for both pulses.

If a short circuit, or low impedance is present in the physical layer, then there will be a "conducting" response to both the positive and negative pulses, and the synchronous detected data is therefore (0,1,1,0) for both pulses.

So, we can distinguish between a proper identity network response, a short circuit, and an open circuit. Refer to Figure 12, which illustrates the correct response of the identity network to a discovery frame for the case where the diode is conducting current during the positive pulse, and not conducting current during the negative pulse. Also shown, in Figure 12, are the results of synchronous detection: (0,1,1,0) and (0,0,0,0).

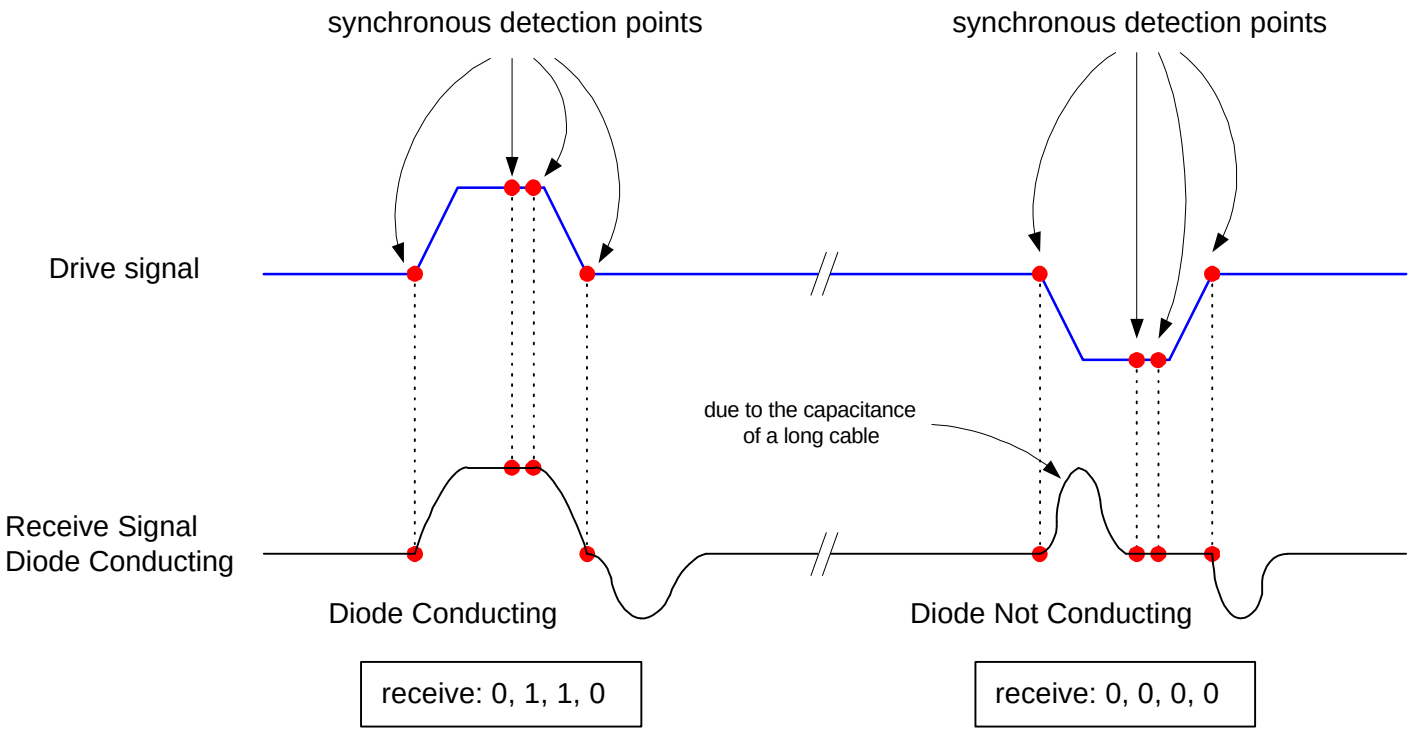


**Figure 12, example of the expected response of the identity network**

Therefore, through the use of synchronous detection, the desired response characteristics are found when the load is the identity network. Also, both open circuits and shorts circuits are each separately detected, and that knowledge prevents successful discovery. A real advantage in network management is that this discovery method can not only find problems with the connected link, but also provide status as to whether the problem is caused by a short circuit, or an open circuit.

- Discovery Pulse Shaping

The discovery pulses are generated in such a way as to produce the minimum harmonic content using pulse shaping techniques. These types of pulses are generally know as being a part of the Nyquist family of pulses [5]: they feature a minimum of required bandwidth in the frequency domain in order to propagate. These signals are used, in this case, so that the discovery process, which utilizes these pulses, can be made independent and orthogonal to the delivery of data on the link. The bandwidth needed for the discovery pulses is less than 100 kHz, which is much lower than the low end of the frequency band used by 10Base-T, or 100Base-T on the UTP cable. Other benefits include minimizing radiated energy, EMC, and maximizing immunity to outside interference.

Refer to Figure 13 which shows scope waveforms captured on the prototype unit running the discovery process. Note that the rise and fall times are in the range of 2 µs to 3 µs.

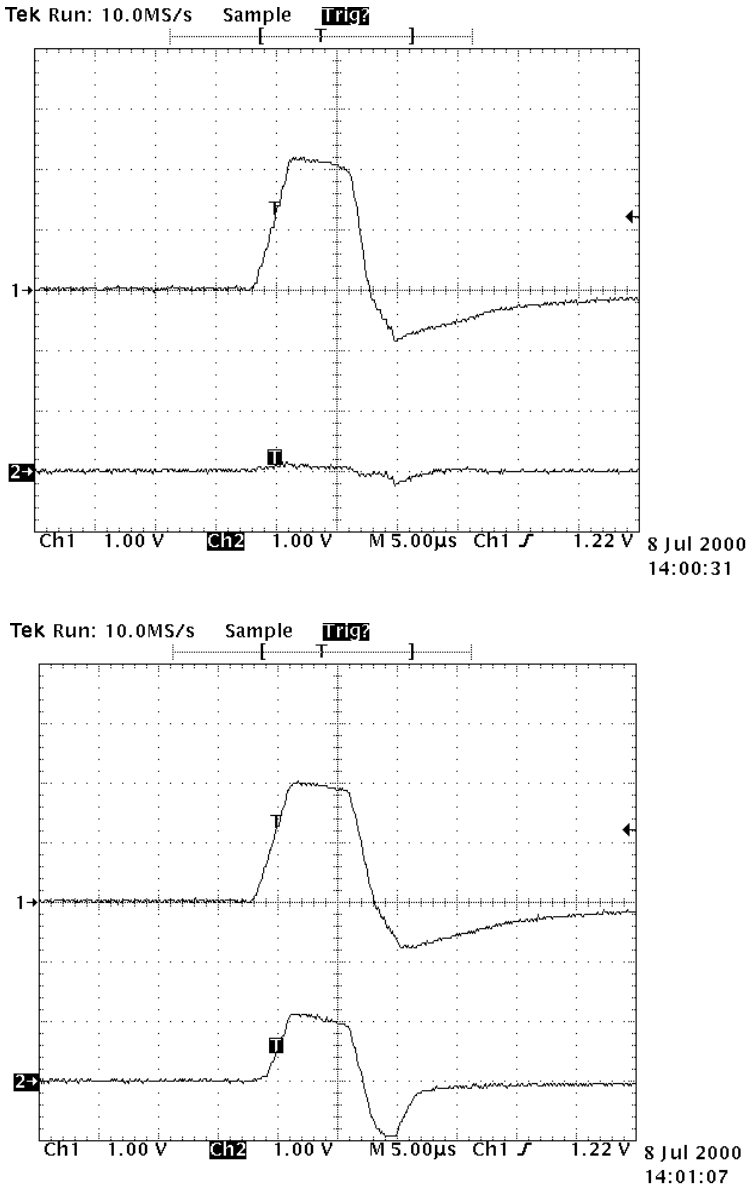


**Figure 13, pulse shaping demonstrated on the prototype unit,**
**Top figure:** ***identity network (diode) not conducting,*** **transmit pulse (upper trace), receive pulse (lower trace)**
**Bottom figure:** ***identity network (diode) conducting,*** **transmit pulse (upper trace), receive pulse (lower trace)**

- Pseudo Random Time Generation

The pseudo random idle times, used in the discovery process, are generated using a general purpose timer, which counts up from zero to the desired count. For the generation of the pseudo random times, the timer value is obtained from a register of the pseudo random number generator. Basically, the generator consists of an array of linear shift registers, as described in reference [4]. The registers are initially loaded with all 1's. The bits 9 and 11 in the linear array are sent through an exclusive OR gate to provide the zeroth bit at the input of the shift registers. The portion of the Verilog [(14)] code that generates the pseudo random numbers, was taken from the TP/PMD spec listed in reference [6].

The 12 bit register, named "ds", contains the present pseudo random number, which is then captured as a snapshot, by the general purpose timer, using the register called "rand_idletime[11:0]". The rand_idletime[11:0] register is the same register which is shown in the description of the discovery process FSM, above. The pseudo random code generator, always runs continuously, but the snapshot is captured into the "rand_idletime" register only when the general purpose timer is started during the idle periods of the discovery process FSM. A new random number is captured for each idle time.

Refer to Figure 14 to view a scope waveform of the prototype unit running the discovery frames. The pseudo random duration of the idle periods is easily seen here. Note that the highest amplitude pulses are the transmit pulses, and the lower amplitude pulses are the receive pulses.

Figure 15 shows an entire set of successful discovery frames, 256 consecutive frames in all.

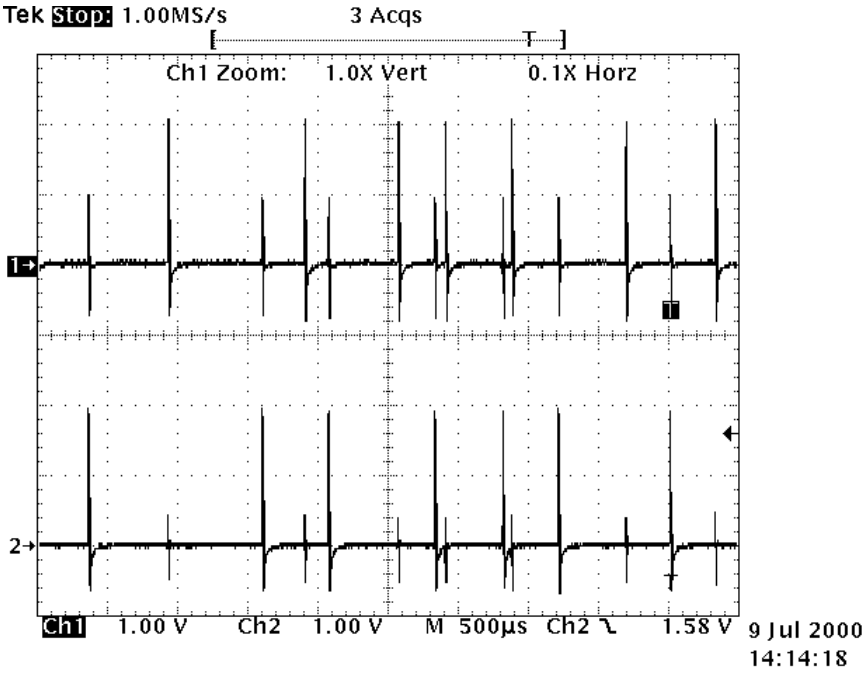


**Figure 14, pseudo random time between pulses is demonstrated on the prototype**
Note that the higher amplitude pulses are the transmit pulses,
and the lower amplitude pulses are the receive pulses

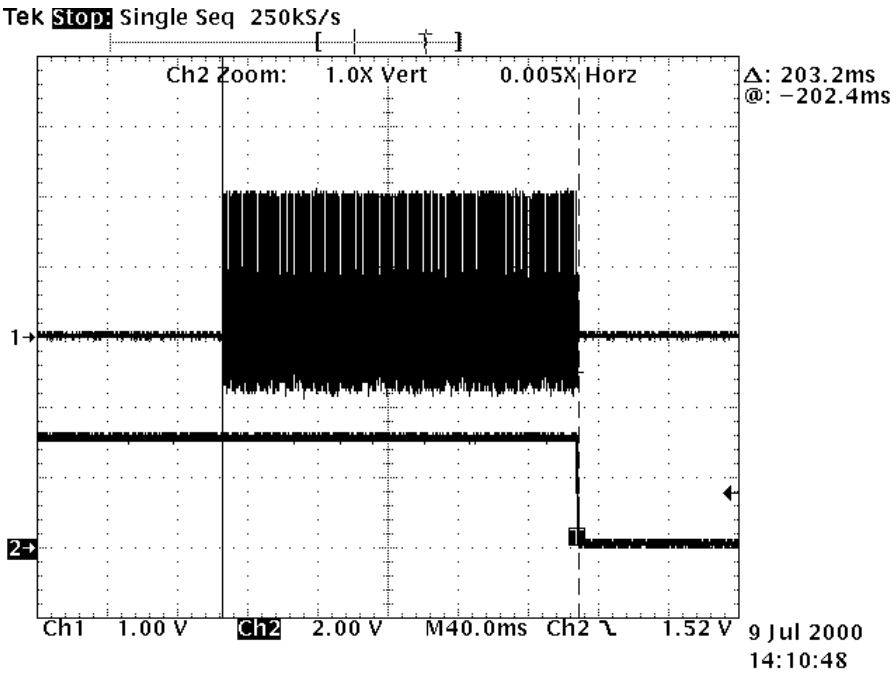


**Figure 15, a successful set of 256 consecutive discovery frames (top trace), and the command signal to turn on the power, active low (bottom trace) demonstrated on the prototype**

**The Power Process:**

Following the successful discovery process, the power for the link can be energized. This is the responsibility of the power process. Network management may optionally be used to enable this process to occur. If enabled, the power process is expected to safely deliver power to the PD, and to continuously monitor for any fault conditions.

Referring to Figure 16, below, once discovery is successful, the power process FSM goes into the “poweroninit” state, and the power supply in the PSE is energized. In order to delay a fixed amount of time, the general purpose timer is called by setting the bit "faultdelaystart" to true. The timer generates a fixed time delay, about 300 ms. This timer is used in order to let the power levels establish themselves in the physical layer. When the timer has finished, it sets the bit "timerdone" to true. Subsequently, the “poweroninit” state sets the bit "faultdelaystart" to false, then proceeds to the "poweron" state.

The “poweron” state keeps the power energized until there is a cause to turn it off. Any one of the following cases will cause the power to turn off, otherwise the power remains on indefinitely:

- a power supply fault such as “over current”, or “over voltage”
- an disconnected link, when the cable is unplugged
- a loss of the power/discovery enable, or "enable_disc" from network management

The first two cases, above, cause the state machine to go to the “fault” state, where a timer is started to insure that the power remains off for a minimum amount of time. In the case of a loss of the power/discovery enable from the network management, the state machine jumps to the “start” state, which is the safe and default state. For the power to turn on again at this point, the full discovery process must run again.

In the "fault" state, the power has been turned off, and it is desired to generate a minimum length of time during which the power will be off. Therefore, the general purpose timer is employed again to generate a delay of 300 ms. The delay is initiated as the "fault" state sets the bit "faultdelaystart" to true. The timer then generates a fixed time delay of about 300 ms. When the timer has finished, it sets the bit "timerdone" to true. Subsequently, the “fault” state sets the bit "faultdelaystart" to false, then proceeds to the "start" state, and we are back at the beginning of all of the processes.

Refer to Figure 16, below.

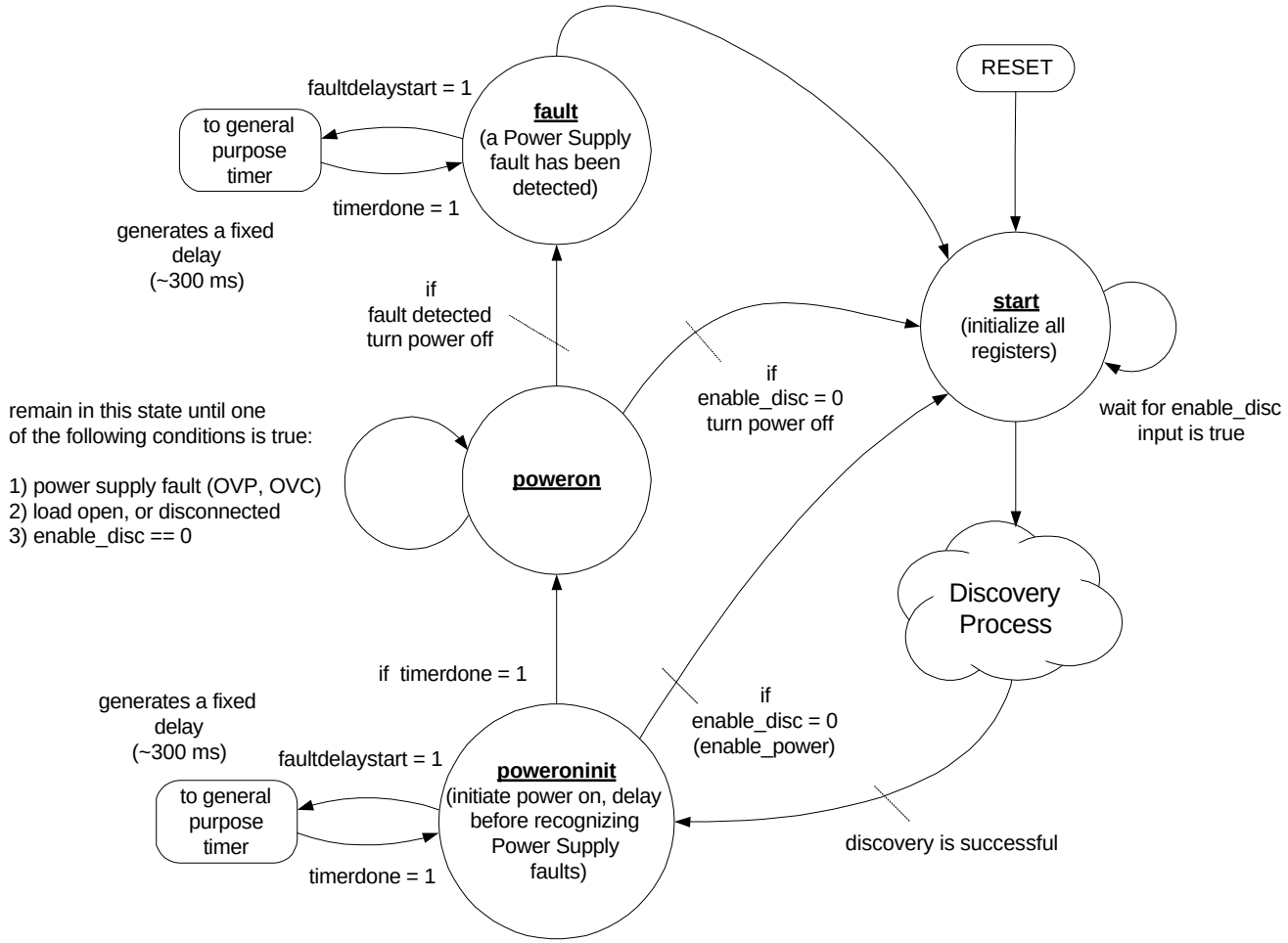


**Figure 16, Power Process Finite State Machine**

**Network Management Interface:**

Here is a likely interface between the discovery and power processes, and the network management function. These would be used for a managed LAN system, and omitted for an unmanaged LAN system. In an unmanaged LAN system, the processes would be configured to start automatically.

Inputs to Network Management:

- PD is discovered
- the power is energized
- power fault: "over voltage", or "over current"
- power supply load "under current" or disconnected
- current state of FSM's

Outputs from Network Management:

- enable discovery process
- enable power process

(or enable both processes)

**Conclusion:**

The discovery and power processes contain protocols that have been described herein. These protocols are correct, in the sense that they make progress only when it is physically safe and also desirable to do so, and also because there are no states which wait for a response that are not at the same time stable operating states. In other words, the discovery and power protocols are both safe and live. For example, the "start" state is a physically safe state, since nothing is happening. Only after being enabled by network management can progress be made into the discovery process. Otherwise, the "start" state will persist indefinitely. The discovery protocol is “safe” because it always makes progress when the conditions are proper; when it detects the PD identity network. If it does not detect the identity network, then the discovery protocol does not make progress, which is exactly what is desirable, and physically safe. In this case, the discovery process repeats indefinitely, or until disabled by network management. The discovery protocol is live because it generates it's own timing, and never waits indefinitely for any input. The discovery process is both safe and live, and therefore “correct”.

The power delivery process is physically safe, since it is reachable only through the discovery process, and must be continuously enabled by network management. The "power" state itself is a stable state under normal conditions; it is the whole reason to supply power over the UTP cable. The power process is really quite simple; it normally persists in the “poweron” state until there is cause to make a change, for example in response to a fault condition. When such a fault occurs, the power process makes the appropriate progress by turning off the power and going back to the “start” state. The power process is therefore correct, as is the top level set of processes: start, discovery, and power.

**Printed Media References:**


1) IEEE Std 802.3 1998 Edition, for LAN/MAN Networks, 28 September 1998

2) Reference Data for Radio Engineers, Howard Sams, 4th printing, 1981, ISBN 0-672-21218-8, parts 35-20 to 35-21, Figure 10 and 11, Interoffice signaling

3) Donald G. Fink, Donald Christiansen, Electronic Engineers' Handbook, 3rd Edition, 1989, ISBN 0-07-020982-0, part 22-82 to 22-84, DC power transfer in telephone systems

4) Daniel Zwillinger, CRC Standard Mathematical Tables and Formulae, 1996, ISBN 0-8493-2479-3, pg 594, pseudo-random number generators using linear shift registers.

5) John G. Proakis, Masoud Salehi , Communication Systems Engineering, 1994, ISBN 0-13-158932-6, section 8.2, pg 547 to 554, signal design for bandlimited channels

6) ANSI X3.263:199X T12/95- TP-PMD/_____ Rev 2.2, March 1, 1995 American National Standard for Information Systems – Fibre Distributed Data Interface (FDDI) – Part: Token ring twisted pair physical layer medium dependent (TP-PMD)


**Web Page References:**


7) IEEE802.3af DTE power via MDI task force main site:
http://www.ieee802.org/3/af/

8) IEEE802.3af DTE Power via MDI, Call for Interest, Nick Stapleton, July 1999:
http://www.iee802.org/3/af/public/july99/stapleton_1_0799.pdf

9) Discovery Matrix, November 1999:
http://www.ieee802.org/3/af/public/nov99/discovery_1199.pdf

10) Power Matrix, November 1999:
http://www.ieee802.org/3/af/public/nov99/power_1199.pdf

11) Idle Pair Discovery Process, Robert Muir, IEEE802.3af, Mar 2000:
http://www.ieee802.org/3/af/public/mar00/muir_1_0300.pdf

12) Common Mode and Differential Mode Discovery Techniques,
Rick Brooks, IEEE802.3af, May 2000:
http://www.ieee802.org/3/af/public/may00/brooks_1_0500.pdf

13) Discovery Protocols and Prototypes,
Rick Brooks, IEEE802.3af, July 2000:
http://www.ieee802.org/3/af/public/jul00/brooks_1_0700.pdf

14) DTE Power Discovery PLD Verilog,
Rick Brooks, IEEE802.3af, July 2000:
http://www.ieee802.org/3/af/public/jul00/brooks_2_0700.pdf

15) DTE Power Discovery Schematics,
Rick Brooks, IEEE802.3af, July 2000:
http://www.ieee802.org/3/af/public/jul00/brooks_3_0700.pdf

16) IEEE802.3 DTE Power via MDI, Detection and Signature Protocol,
Richard Glaser, Robert Leonowich, et al., IEEE802.3af, May 2000:
http://www.ieee802.org/3/af/public/may00/glaser_1_0500.pdf

17) Resistive Signature and Discovery Update,
Robert Leonowich, IEEE802.3af, July 2000:
http://www.ieee802.org/3/af/public/jul00/leonowich_1_0700.pdf

18) IEEE802.3af DTE Power via MDI Enhanced I/V Detection Method,
Amir Lehr, IEEE802.3af, July 2000:
http://www.ieee802.org/3/af/public/jul00/lehr_0700.pdf