Link Aggregation Control Protocol

Mick Seaman

This is a proposed revision of the protocol presented by Tony Jeffree at the July'98 and September'98 meetings of 802.3ad as described in Rev 1.1 of this note. It simplifies the description of the Desire, Nervous, and periodic transmission capabilities of the Transmit Machine into a Periodic Transmission machine, removing a timer. The information communicated by the protocol is unchanged.

The prior separation of these functions facilitated analysis of mechanisms ensuring predictable protocol operation for links attached to shared media, and the timer supported these. Such links cannot occur if 802.3 standard equipment is used: the scope of the proposed link aggregation standard is limited to full duplex links, and there is no standardized full duplex repeater.

A separate note describes possible protocol extensions, outside the scope of the proposed standard, that could be implemented if there is a concern that such non-standard repeaters might exist, now or in the future.

Terminology has been updated in line with group decisions and some footnotes of purely historic interest removed.

While not reiterating basic concepts, this note attempts a complete description of the protocol, with the exception of flush mechanisms which are orthogonal to LACP itself.

Protocol Participants

The protocol is described from the point of view of individual **physical ports**. Each physical port that may be aggregated with other physical ports is a **participant** in the protocol.

When it is clear that protocol exchanges between participants in separate systems are being discussed (rather than the aggregate behavior of participants in a single system) the term "participants" refers to the local participant, sometimes called the "**actor**" for clarity, and his remote "**partner**".

Protocol Machines

As an aid to understanding and analysis the protocol machine for each participant is partitioned into the following components:

- Receive Machine
- Selection Logic and Machine
- Match Logic
- Mux Control Logic and Machine
- Periodic Transmission Machine
- Transmit Machine

The **Receive Machine** receives protocol information (contained in LACPDUs) from (a) remote partner(s), records the information, and times it out after an expiry period.

The **Selection Logic and Machine** selects the **aggregate port** to be associated with the physical port.

The **Match Logic** determines if the participants have both agreed on the protocol information exchanged to the extent that the physical port can be safely used in an aggregate[1] (possibly as an **individual port**, i.e. as the sole port in that aggregate).

The **Mux Control Logic and Machine** turns the distributor and collector for the physical port on or off as required by protocol information.

The **Periodic Transmission Machine** establishes the desire of the participants to exchange LAC PDUs periodically to maintain an aggregate, and how often periodic transmission should take place based on the timeout used by the receiving participant.

The **Transmit Machine** formats and transmits LACPDUs as required by the Periodic Transmission Machine and by other machines if the partner's view of the actor's state is not current. It imposes maximum transmission rate limitations on LACPDUs.

[1] The distinction between the "selection logic" and the "match logic" is important and difficult to grasp initially. While the match logic will never select a different aggregate port to the selection logic, the selection process is necessary and distinct. As an example, a physical port which has been "unplugged" (disabled) may continue to select the same aggregate port so that disruption to higher protocol layers is minimized, particularly if the plug is put back in later allowing the port to resume its role in the aggregate. However while the port is unplugged it will not be "matched". The distinction between selection and matching also proves useful in modeling the way in which the protocol accomodates system delays in reconfiguring mux, distributor, and collector resources. This should become apparent as the description unfolds.

Receive Machine

The receive machine extracts and records the following information from a received LAC PDU[2]:

- Partner[3]'s System ID
- Partner's Key
- Partner's partner[4] System ID
- Partner's partner Key
- Partner's State
- Partner's View

The partner's **view** is simply what the partner believes to be the **state** of its partner[5]**.** The state information communicated is that additional to the system ids and keys, comprising the following flags[6]:

- Active
- Short timeout
- Aggregate
- Sync
- Collector on
- Distributor on

Each of these is described in detail in the relevant machine, a brief summary follows.

The **Active** flag indicates a participant's intent to transmit periodically to detect and maintain aggregates. If set[7] the flag communicates **active LACP**, if reset **passive LACP**. A passive participant will participate in the protocol if it has an active partner.

The **Short timeout** flag indicates that the participant wishes to receive frequent periodic transmissions, probably because it lacks confidence that its own hardware will indicate a disabled physical link quickly, and will aggressively times out received information. If set the flag communicates **Short timeout**, if reset **Long timeout**[8]**.**

The **Aggregate** flag indicates that the participant will allow the link to be used as part of an aggregate. Otherwise the link is to be used as an **individual link**, i.e. not aggregated with any other; e.g. the participant may know that the link has a **unique key**[9] (at present) and hence will not be aggregated. Signaling individual allows the receiving actor to skip protocol delays that are otherwise invoked to allow all links with the same system id and key combinations to be collected into one aggregate port without successive rapid changes to aggregate ports and accompanying higher layer protocol disruption.[10] If set the flag communicates **possible aggregate**, if reset **individual**.

The **Sync** flag indicates that the transmitting participant's mux component is in sync with the system id and key information transmitted. This accommodates multiplexing hardware[11] that takes time to set up or reconfigure.[12] If set the flag communicate **in sync**, if reset **out of sync**.

The **Collector on** flag indicates that the participant's collector, i.e. the reception component of the mux, is definitely on. If set the flag communicates **collector on**.

The **Distributor on** flag indicates that the participant's distributor is not definitely off. If reset the flag indicates **distributor off**.

[2] Only one (the last received) LAC PDU is recorded.

[3] The "partner" referred to here is the transmitter of the LAC PDU.

[4] The partner's partner is of course the actor, but the partner may not know that yet. Information received may reflect old knowledge.

[5] The partner's view is simply the partner's partner state, i.e. the actor's state if the partner knows about the actor and initial or old information if it does not.

[6] It is proposed that these flags be encoded in a LAC PDU as bits in a transmitted octet. Since only 6 flags are currently defined it is highly desirable that the treatment of the two 'spare' flags be explictly defined to facilitate interoperability and protocol upgradeability should that ever be desired. This definition should include how received flags are reflected into the actor's view.

[7] Strictly speaking we are discussing interpretation of the flag within the protocol entity here, not the encoding of LAC PDUs.

[8] In the previous revision an actor monitored it's partner view so as to use a short timeout only if its partner was aware of that. As Norm Finn pointed out this was an unnecessary complication under the single lost message design assumption.

[9] An alternative approach to explictly signaling "individual" would have been to allow the protocol machine to change the key to a reserved null value with the same semantics. However this blurs the network administrator's original intention (in setting the key value) with operational actions taken by the protocol entity (deciding a link is individual on the basis of its own information rather than having to consult with its partner). Using 'clever' encodings which have this blurring effect add nothing to protocol simplicity, transparency, or upgradability, and we are hardly short of the encoding space for one bit.

[10] Asserting "individual" is a potential exit route for the protocol machine in future scenarios and it is anticipated that it will be useful if an extension to shared media is ever standardized. Another reason not to confuse this functionality with the administrator assigned key.

[11] And software if the multiplexing is under the control of a separately scheduled software process which may communicate with the actor's port based LACP protocol entity through operating system style messaging mechanisms, with the types of delays that typically implies.

[12] While a principal goal of this protocol is ensuring high availability, that does not require that new links or physical ports be added to aggregates rapidly, simply that links or physical ports that have failed be removed in a timely fashion. Since the mechanism prompting addition will usually involve administrator intervention either at an administrative console or simply adding a physical link in a plug and play environment that is fortunate. This observation can be taken advantage of in structuring the protocol design and ensuring that it applies to the widest possible set of existing and new hardware and systems.

Receive Machine States and Timer

The receive machine has two states, other than those implied by the stored data described above. They are:

- Current

  and

- Expired

In the **Current** state the receive machine has current received protocol data, in the **Expired** state it does not. The initial state, on creation of the protocol entity is Expired.

The Receive Machine has a single timer, the **current while** timer. This runs in the Current state, on expiry the machine transitions to the Expired state. The timer is started or restarted on receipt of a valid LAC PDU. Its starting value[13] is either:

- Short timeout

  or

- Long timeout

A administratively settable boolean parameter **actor's timeout**, selects between these[14], and determine the setting of the Short timeout flag in the actor's state transmitted in LACPDUs.

Receive Machine Events

The following events can occur:

- participant created
- participant reinitialized[15]
- received LAC PDU
- physical MAC disabled
- current while timer expiry

The physical MAC disabled event indicates that either or both of the physical MAC transmission or reception for the physical port associated with the actor have become non-operational after being operational.

The received LAC PDU event only occurs if both physical transmission and reception are operational, so far as the actor is aware[16].

Receive Machine Actions

The receive machine can take the following local actions:

- record received PDU information
- clear received PDU information
- start the current while timer

Receive Machine Signals

The receive machine uses the following signals:

- information received
- information expired

to communicate to the other machines comprising the LACP participant as shown in the Receive Machine State table.

Additionally the signal:

- need to transmit

is sent to the Transmit Machine if the Partner's partner System ID, Key, or View differs from the actor's System ID, Key or State.

Receive State Machine

	Expired	Current
create	-	X
reinitialize	-	clearPDU infoExpired Expired
receivedPDU	recordPDU start_current infoReceived Current	recordPDU start_current infoReceived -
pMACdisabled	-	clearPDU infoExpired Expired
current_expiry	-	clearPDU infoExpired Expired

[13] Throughout this description timers are described as down counters which expire when they reach zero.

[14] The values of these timeouts are set by the protocol specification; they are not changed by implementations or network administrators except to select the short or long value.

[15] This is a management event whose purpose is to restore the protocol entity to its initial state gracefully without dropping any loose ends. Typically protocol descriptions omit the specification of basic management operations thus inviting implementation problems. We attempt to avoid this mistake.

[16] This removes the need for the receive machine to explicitly track the physical MAC operational states and for a separate physical MAC enabled event to the machine. Occurrence of a receive event implies physical MAC enabled, and the machine has no interest in the enabling event other than the potential reception of LAC PDUs.

Periodic Transmission Machine

This machine establishes the desire of the participants to exchange LAC PDUs periodically to maintain an aggregate, and how often periodic transmission should occur.

Periodic exchanges[17] will take place if either participant so desires[18]. Transmissions occur at a rate determined and communicated by the receiving participant.

Periodic Transmission Machine States

This machine has three states:

- No Periodic transmission
- Fast Periodic transmission
- Slow Periodic Transmission

A boolean parameter, **actor's activity**, administratively settable to **active LACP** or **passive LACP** influences the state machine and determines the value of the Active flag in the actor's state transmitted in LACPDUs.

A boolean parameter, **actor's timeout**, administratively settable to **short timeout or long timeout** determines the value of the Short Timeout flag in the actor's state transmitted in LACPDUs. This then influences the partner's state of the partner's periodic transmission machine.

Periodic Transmission Machine Timers

The periodic transmission machine has two timers, the **periodic transmission** timer and the **fast while** timer.

The periodic transmission timer stimulates periodic transmissions to ensure that the actor's information is not timed out by a protocol partner, and, if the actor is active, to discover a new partner. In the Fast Periodic state it is restarted on expiry with an initial value of Fast Transmit, and in the Slow Periodic state with an initial value of Slow Transmit.

If the Receive Machine has Current information from a protocol partner the periodic transmission machine state is selected by the received value of the Short timeout flag – Fast Periodic for Short timeout, Slow Periodic for Long timeout.

If the Receive Machine's stored protocol information has Expired (which includes no information having been received since creation, initialization, or enabling the physical MAC), the periodic transmission machine state will be No Periodic if the actor is passive, otherwise (for an active actor) the state will be Fast Periodic if the fast while timer is running, and Slow Periodic otherwise. The timer is started or restarted with an initial value of Fast Timeout when the machine is reinitialized or when the receive machine signals information expired. The fast while timer thus operates to ensure that protocol partners are discovered rapidly while allowing for a slow periodic rate in the steady state to ensure that no new partner is missed[19].

Periodic Transmission Machine Events

The following events can occur:

- physical MAC enabled, participant created or reinitialized, with active actor
- physical MAC enabled, participant created or reinitialized with passive actor
- information received (from the Receive Machine) indicating an partner with a fast timeout[20], with either that partner or the local actor being active
- information received (from the Receive Machine) indicating an partner with a slow timeout, with either that partner or the local actor being active
- information received indicating a passive partner while the actor is passive
- information expired (signaled from the Receive Machine) while the actor is active
- information expired (signaled from the Receive Machine) while the actor is passive
- management sets the actor to active LACP
- management sets the actor to passive LACP[21]
- fast while timer expires
- periodic transmission timer expires

Periodic Transmission Machine Signals

The desire machine uses the following signal to communicate to the transmit machine:

- need to transmit

This signal is invoked as shown in the Periodic Transmission Machine State Machine table.

[17] This machine only governs periodic transmission. If management operations cause both participants to be automatic, there may be exchanges that need to take place to move the physical port gracefully to an individual link. These will occur and LAC PDU transmissions will stop only when the configuration has reached a steady state.

[18] If the protocol were to be extended to shared media, periodic exchanges would take place if **any** participant so desired and at the fastest rate desired by any participant.

[19] Thus an implementation which was only capable of being passive, should such an implementation be standards compliant, could avoid having to use the timer. Likewise if either slow guaranteed discovery of a new partner or continuous operation at fast transmission rates in the absence of any partner is acceptable, the timer could be omitted.

[20] The events of one machine do not necessarily correspond one for one to the signals of another. A given signal may be conditioned by data to result in zero, one, or more events. Likewise signals can be combined into a single event.

[21] In theory this could be improved upon by consulting any current information held by the Receive Machine to see if the partner is active. In practical terms this is far too fussy given how rare the management operation on a live system will be, and the fact there will be little or no disruption even if the Received machine is not consulted. Keep it simple.

Periodic Transmission State Machine

	No_periodic	Fast_periodic	Slow_periodic
physical MAC enabled, create, reinitialize, active actor[22]	need_to_transmit (re)start_fast_while (re)start_periodic(fast) Fast_periodic	need_to_transmit (re)start_fast_while (re)start_periodic(fast) -	need_to_transmit (re)start_fast_while (re)start_periodic(fast) Fast_periodic
physical MAC enabled, create, reinitialize, passive actor	-	No_periodic	No_periodic
rcvd info, fast timeout, active actor or partner	need_to_transmit (re)start_periodic(fast) Fast_periodic	-	need_to_transmit (re)start_periodic(fast) Fast_periodic
rcvd info, slow timeout, active actor or partner	need_to_transmit (re)start_periodic(slow) Slow_periodic	need_to_transmit (re)start_periodic(slow) Slow_periodic	-
rcvd info, passive actor and partner	-	No_periodic	No_periodic
info expired, active actor	need_to_transmit (re)start_fast_while (re)start_periodic(fast) Fast_periodic	need_to_transmit (re)start_fast_while (re)start_periodic(fast) -	need_to_transmit (re)start_fast_while (re)start_periodic(fast) Fast_periodic
info expired, passive actor	-	No_periodic	No_periodic
set actor active	need_to_transmit (re)start_fast_while (re)start_periodic(fast) Fast_periodic	need_to_transmit (re)start_fast_while (re)start_periodic(fast) -	need_to_transmit (re)start_fast_while (re)start_periodic(fast) Fast_periodic
set actor passive	-	No_periodic	No_periodic
fast while timer expires, info has expired	-	Slow_periodic[23]	-
fast while timer expires, info has not expired	-	-	-
periodic transmission timer expires	-	need_to_transmit restart_periodic(fast) -	need_to_transmit restart_periodic(slow) -

[22] The effect of the treatment or initial and reinitialized states is that an active participant will send a few LACPDUs at the fast rate before settling down to steady periodic transmission at the slow rate (absent detecting a protocol partner using fast timeouts). A passive participant will remain quiet until spoken to.

[23] Will only be in this state if the actor is active.

Selection Logic and Machine

The selection logic and machine selects the aggregate port to be associated with the physical port.

It is worthwhile reviewing the protocol modeling objectives that lead us to distinguish between physical and aggregate ports. This note proposes a set of rules for the use of aggregate ports that:

a) do not require additional MAC addresses to those already provided by the physical MACs,

b) provide an element of determinism (history independence) in the assignment of physical ports to aggregate ports, and

c) should match the users intuition in the trivial cases where individual links result[24].

These rules are compatible with an alternative view of link aggregation that does not consider it in terms of physical ports attaching to aggregate ports, but rather as a set of physical ports bonding together.

Note that the rules described are not required by the proposed protocol, which can accommodate greater flexibility in the relationship of aggregate and physical ports. They are introduced here to allow the reader to have a clear and concrete view of the operation of the protocol. Quite separately we believe that it will be necessary to specify such rules, or very similar ones, to ensure acceptance of link aggregation, for the reasons outlined above.

Physical Ports and Aggregate Ports

Physical ports represent a single point of attachment to a transmission medium and the particulars of the access method used to access that medium. Aggregate ports represent a point of attachment of higher layer protocols.

So if you are in the aggregation 'layer' you look down through physical ports to the transmission medium and up through aggregate ports to the users of that medium.

In the familiar case of individual links there is a trivial one to one correspondence between physical and aggregate ports which is so obvious that we do not distinguish them[25].

Selection Rules

Each physical MAC naturally has (comes equipped with) both a physical port and aggregate port.

Aggregation is represented by attachment or association of the physical port with an aggregate port.

Every physical port always has one aggregate port selected at any point in time.

A physical port that is operating as an individual link always selects, and has first claim on its own aggregate port.

A number of physical ports in an aggregate always select the lowest numbered port[26] for their aggregate port. The corresponding physical port may not be in a state that allows data to be transferred on its physical link but it has selected that aggregate port.

The following diagrams illustrate these rules. Figure 1 shows a valid configuration and Figure 2 an invalid one.

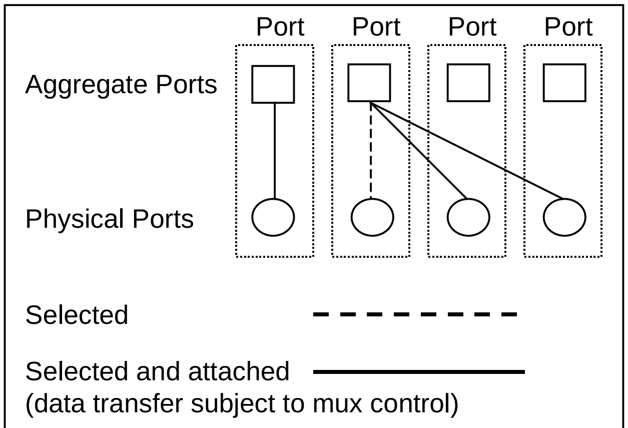

Figure 1

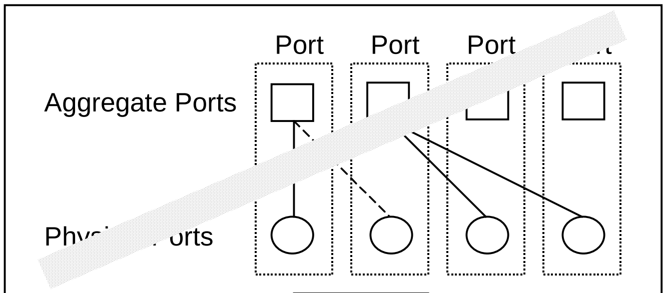

Figure 2

[24] This last point is believed to be very important when introducing the protocol particularly if the objective is to seed the market by universal deployment through a period where customers may be sceptical as to its value because not all attached devices are running the protocol. Counter intuitive behavior of systems running the protocol and conforming to the link aggregation architecture but not providing additional functionality in this period would be a significant negative.

[25] Most network protocols were originally designed to run over a single link layer access point, and as the requirement to provide transparent multiplexing over a number of links as emerged this distinction between 'user down' ports and 'provider up' ports has been widely introduced and is familiar to the designers and user of wide area protocols. Most routers provide a universal abstraction for this concept across the details of the particular multiplexing technology.

[26] An arbitrary rule of course.

Selection Logic

The logic by which an aggregate port is selected operates on the information recorded by the receive process. It determines:

a) the selected partner's system id and key

b) whether an individual link has been selected as opposed to possible participation in an aggregate

c) whether a new partner[27] has been selected, i.e. whether there has been a change in the partner system id or key[28]

and records the selected partner system id, key, and selected as individual (or not) state.[29]

The selected partner system id, selected partner key, and "selected as individual" information is updated when:

- the receive machine signals information received[30]
- the wait while timer used by the selection machine expires
- one or more of the actor's parameters that contribute to the selection process have been changed by management[31].

The selected partner's system id and key are set to that recorded by the receive machine (null if that data had previously expired, i.e. the received machine state is Expired).

Selected as individual is true if the receive machine state is Expired, if one or more of the actor's state, the partner's state, or the partner's view are individual, if the partner's system id and key are the same as the actor's[32], or if both partners are automatic. In summary, if either partner has a suspicion the port should be selected as an individual port it will be.

If selected as individual is true, the aggregate port selected is the port's own aggregate port, otherwise the aggregate port is the lowest numbered port with matching selection parameters (same local system id and key, same partner system id and key, not selected as individual).

If the selection parameters for a given physical port are changed, other ports in the system may have to reselect their aggregate ports[33],

[27] Determining that the partner is new serves two purposes. Firstly it acts as a prompt to the aggregate port selection process along with any change to the selected as individual state. Secondly it may be used to flag to higher layers including those concerned with peer authorization and related matters that the partner has changed. Otherwise link aggregation may fail to preserve the same physical port down up characteristics that exist with purely individual links. The 'new partner' determination is not exported from the selection logic through any new layer management interface but will cause 'down up' behavior at the aggregate port as the selection machine is currently defined. Here it is necessary to declare "this far and no further" so far as physical topology discovery is concerned.

[28] Even in a point to point situation with perfect hardware the partner system id and key may change as the partner may be managed.

[29] It is easier to describe the functionality associated with this process if it is imagined that a separate record is taken of the partner's id, key, and selected individual state. However when receive information is current there is never any difference between the information recorded by the selection logic and that already recorded by the receive machine. There is no implementation requirement for increased store, although the new partner determination has to be done by the receive machine with some impact on conceptual modularity.

[30] Note that the selection parameters are not updated when the receive machine signals information expired, that signal is used to start or restart the wait while timer to give the partner or the physical link to become operational once more.

[31] This is a principal reason why all the received data is recorded, absent this possibility all of it save the outcome of the selection logic and the outcomes for other machines could have been discarded.

[32] This ensures that a link between two ports with the same key on the same system does not attach to an aggregate port and create a new loopback capability unknown to any existing loopback controls.

[33] The search for other ports that may have to select the same aggregate can be narrowed significantly. Unless the local key has changed it can be restricted to those ports with a matching key. Further if the port whose parameters have changed was not the lowest numbered port in its previous selection and it not the lowest numbered in its new selection it will not affect the choice of aggregate port by other physical ports.

Selection Machine

The selection machine attaches the physical port to the selected aggregate port, or to put it another way the selection machine combines that fraction of a mux represented by the physical port with others which taken together comprise the mux for the aggregate port.

After a physical port changes its selection of aggregate port, the selection machine detaches it from the current aggregate port. It then waits a while to allow changed protocol information from the partner system which may be expected to arrive on several physical ports to do so[34].

After the waiting period has elapsed it confirms the continued validity of the received selection information, checks that no other physical port which has selected the same aggregate port is still detaching itself from the prior aggregate[35] or is still waiting, and attaches to the new aggregate.

Selection Machine States and Timer

The selection machine has four states:

- Detached
- Attaching
- Attached
- Detaching

and one timer, the **wait while** timer.

These states are probably better expressed in terms of the running timer and two booleans which express the administrative and operational states of the physical port's attachment to the mux:

- **attach**

  and
- **attached**

If attach is true, the selection machine wants the physical port attached to the currently selected aggregate port. If attach is false, the selection machine wants the physical port detached from whatever aggregate port it is currently attached to. For hardware that responds synchronously and instantaneously, attached will always equal attach.

Selection Machine Events

The following events can occur:

- change aggregate port request from the selection logic[36]
- ready to attach to the selected aggregate, this means that neither this port not any other selecting the same aggregate is currently detaching from another aggregate, or has the wait while timer running
- attached signal from the mux control machine
- detached (attached is false) signal from the mux control machine
- physical MAC enabled
- physical MAC disabled.

The selection logic requests a change of aggregate port when either a different aggregate port has been selected or there has been a change of partner[37].

The wait while timer is stopped when the physical MAC (both transmission and reception) is disabled. This allows the selection machine to hold onto prior aggregate selection details when a physical port is disconnected. The timer is started with its initial value when the physical MAC is enabled thus allowing a period for the participants to reestablish the port as part of a former aggregate.

Selection Machine Actions

The selection machine can take the following local actions:

- start or restart the wait while timer
- stop the wait while timer

Selection Machine Signals

The selection machine uses the following signals to communicate to the mux control machine, which in turn controls the multiplexing hardware:[38]

- attach to a specified aggregate port
- detach from the current aggregate port

[34] Thus minimizing thrashing of the higher layers which is important since port up events may consume considerable numbers of buffers for initial protocol use.

[35] The machine thus accomodates hardware delays, though there is no requirement to delay artificially. Although a description at this level of detail is necessary to explain what the higher level protocol user of the aggregate port may see, and to validate the protocol's fitness for deployment across a wide range of hardware, this detail is not communicated in the protocol. This allows both "instantaneous" and more convoluted implementations to be accomodated by the protocol.

[36] If the port has been selected as individual then the wait while timer can be stopped (if the current state is Detached) and the port can proceed directly to Attaching if no other physical port is currently attached to its aggregate port.

[37] A state machine rule (mentioned above) holds new additions to an aggregate port until any ports to be aggregated to that port who are leaving their current aggregate port do leave. This rule operates here to ensure that a change of partner does bring the aggregate port down even if there are hardware switching delays which might allow new additions to the port just after a change of partner to keep the port up.

[38] The selection machine state transition tables operate so as not to send attach or detach signals when the mux control machine and the underlying hardware have not completed the prior attach or detach instruction. Whether or not an implementation mimics this polite behavior is entirely up to the implementor, conformance is, as always, assesed purely on the basis of observed external behavior.

Selection Machine State Table

States	Detached	Attaching	Attached	Detaching
State variables		attach	attach attached	attached
change[39]	start_wait[40] -	-[41]	detach Detaching	-
ready	attach Attaching	-	-	X
attached	X	Attached	X	X
detached	X	X	X	start_wait[42] Detached
pMACenabled	start_wait	start_wait	start_wait	start_wait
pMACdisabled	stop_wait	stop_wait	stop_wait	stop_wait

[39] Change in selection parameters as updated following receive information, information expiry, change of state of the selection machine, or change of the actor's manageable parameters.

[40] The wait while timer will only be started if the change is to a possible aggregate, if the port is selected as an individual port at any time the timer will be stopped.

[41] Change will be checked for on transition to Attached state.

[42] Only if the port is selected as part of a possible aggregate, not if it selected as an individual port.

Match Logic

The match logic determines if the participants have both agreed on the protocol information exchanged to the extent that the physical port can be safely used in an aggregate. The match logic comprises a simple set of predicates over the data recorded by the receive and selection machine:

The data is "**matched**" if the physical MAC is enabled and:

1. The actor has **no partner**, i.e. the received machine is in the Expired state and the selection machine has recorded a null system id as the selected partner id.[43]

or

2. The actor has a **matched individual**, i.e. there is current information from a partner and either:

   a) the received partner state signals individual

   or

   b) the actor's own state is individual and the received partner's view is individual

   or

   c) there is current information from a partner but the partner's id and key are the same as the actor's id and key[44].

or

3. The actor has a **matched aggregate**, i.e. there is current information from a partner and the partner's partner id and partner's partner key match those of the actor, and the selection logic has not identified the selected aggregate port as individual[45].

Mux Control and Logic

The mux control and logic turn the distributor and collector on or off as required by the selection machine and protocol information.

Leaving the aside the administrative and operational boolean states which can be used to model switching on and off the collector and distributor, the mux has two states:

- In_sync
- Out_of_sync

The mux is "in sync" if the selection machine has Attached[46] the physical port to the correct aggregate and the match logic indicates "matched".

Given the wide range of possible implementation considerations and constraints, operation of the mux is best specified in terms of the goals for collector and distributor operational states given the above states and received information[47].

If the mux is Out_of_sync or the partner's state is signaling out of sync, then both collector and distributor should be turned off[48].

If the mux is In_sync and the partner's state is in sync, then the collector should be turned on.

If the mux is In_sync, the partner's state is in sync, and the partner's collector is turned on then the distributor should be turned on.

If the mux hardware is **coupled**, i.e. forces the distributor to turn on when the collector is turned on then the above rules also apply.

If the mux hardware is **independent**, i.e. not coupled, then if the partner's collector is turned off, the distributor should be turned off[49].

[43] Which means that either a partner has never been seen on this port since it was last completely reinitialized, or the physical MAC has been enabled for long enough for the wait while timer to expire since the last partner record expired.

[44] This allows a link from one port on a system to another port on that system with the same key to be brought up without causing a new type of loopback or reflective port to be instantiated at the aggregate port level. The specified behavior follows what would occur had LACP not been present. Perhaps a test for the transmitted port identifier (not mentioned in protocol parameters above) should be added to prevent any loopback at the aggregate port level.

[45] "the selection logic has not identified" means there is not a whiff of individualism in the received data, the match logic can equally be based directly on that data by copying the rules for suspecting a port as individual from the selection machine.

[46] Administrative and operational variables **attach** and **attached** are both true.

[47] Complete absence of received information, i.e. the no partner case is treated as if a partner was in sync and had both collector and distributor enabled.

[48] And the Sync flag in the actor's state will be transmitted as **out of sync**.

[49] While graceful removal of a link from an aggregate is not currently specified this behavior supports managing that graceful removal from one end of the aggregate without having to invoke higher layer coordination.

Transmit Machine

The transmit machine maintains the following information:

- Actor's System ID[50]
- Actor's Key
- Actor's State[51]
- Actor's View

The actor's state and view include the Active, Short timeout, Aggregate, Sync, Collector on, and Distributor on flags as described for the receive machine.

Need To Transmit

The Transmit Machine transmits a properly formatted LACPDU within a Fast transmit time following the signaling of a need to transmit from another protocol machine.

Hold Timer and Count

The transmit machine limits the maximum transmission rate of the protocol participant to no more than 3 LACPDUs[52] in a Fast transmit interval. If a need to transmit signal occurs when such limiting is in force the transmission is delayed.

[50] The actor's system ID and key are consulted by the match logic. They could have been stored there.

[51] The actor's state and view are updated by other machines. There is no practical difference between describing this information as being collected from those machines just prior to transmission (as at the July'98 meeting) and being held by the transmission machine and updated by the other machines just when changes happen. The latter approach, used here, is perhaps a little less modular since the transmission machine needs to determine what is a significant change that would prompt transmission. However it does have other modeling benefits.

[52] Sufficient to establish an aggregate in the worst case when initial LACPDUs from the participants cross.