



Revision History

CMS Ethernet LAG
Broadcom User Space CMS Ethernet Link Aggregation

Application Note

Revision History

Revision	Change Description
0.1	03/18/2018 First draft
0.2	04/23/2018 PURE181 LAN Ethernet LAG Implemented.
0.3	05/30/2018 PURE181 WAN Ethernet LAG Implemented.

Broadcom, the pulse logo, Connecting everything, Avago Technologies, Avago, and the A logo are among the trademarks of Broadcom and/or its affiliates in the United States, certain other countries and/or the EU.

Copyright © 2018 by Broadcom. All Rights Reserved.

The term “Broadcom” refers to Broadcom Limited and/or its subsidiaries. For more information, please visit www.broadcom.com.

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

1 Introduction

Broadcom CPE Ethernet Link Aggregation Group (LAG) configuration is based on the existing released CPE Linux Kernel which is referred in the menuconfig - LAG as “Interface Bonding” and the capability is provided through Kernel Bonding Driver. This application note describes the supported configuration to the existing released the Linux Bonding Driver.

2 Related Documents

The references in this section may be used in conjunction with this document.

Document (or Item) Name	Number	Source
[1] <i>IEEE 802.3AD/LACP</i>	–	http://www.ieee802.org/3/hssg/public/apr07/frazier_01_0407.pdf
[2] <i>Broadcom provided CPE kernel bonding</i>	–	commEngine\docs\customerDocs\Kernel_Bonding_Driver_AppNote.pdf

3 How to build an Ethernet LAG enabled image

Since Ethernet Bonding interface is defined in the latest TR181 data model, it is supported when:

- 1). Linux Bonding Driver is selected in the profile -- see Fig. 1 below from “make menuconfig PROFILE=962118GW”
- 2). It is a PURE TR181 data model

To create an Ethernet LAG enabled PROFILE by doing the following:

```
release/maketargets 962118GW_PURE181
```

- 2). make PROFILE=962118GW_PURE181

- 3). Once the build done, images will be under: targets\962118GW_PURE181 directory.
-

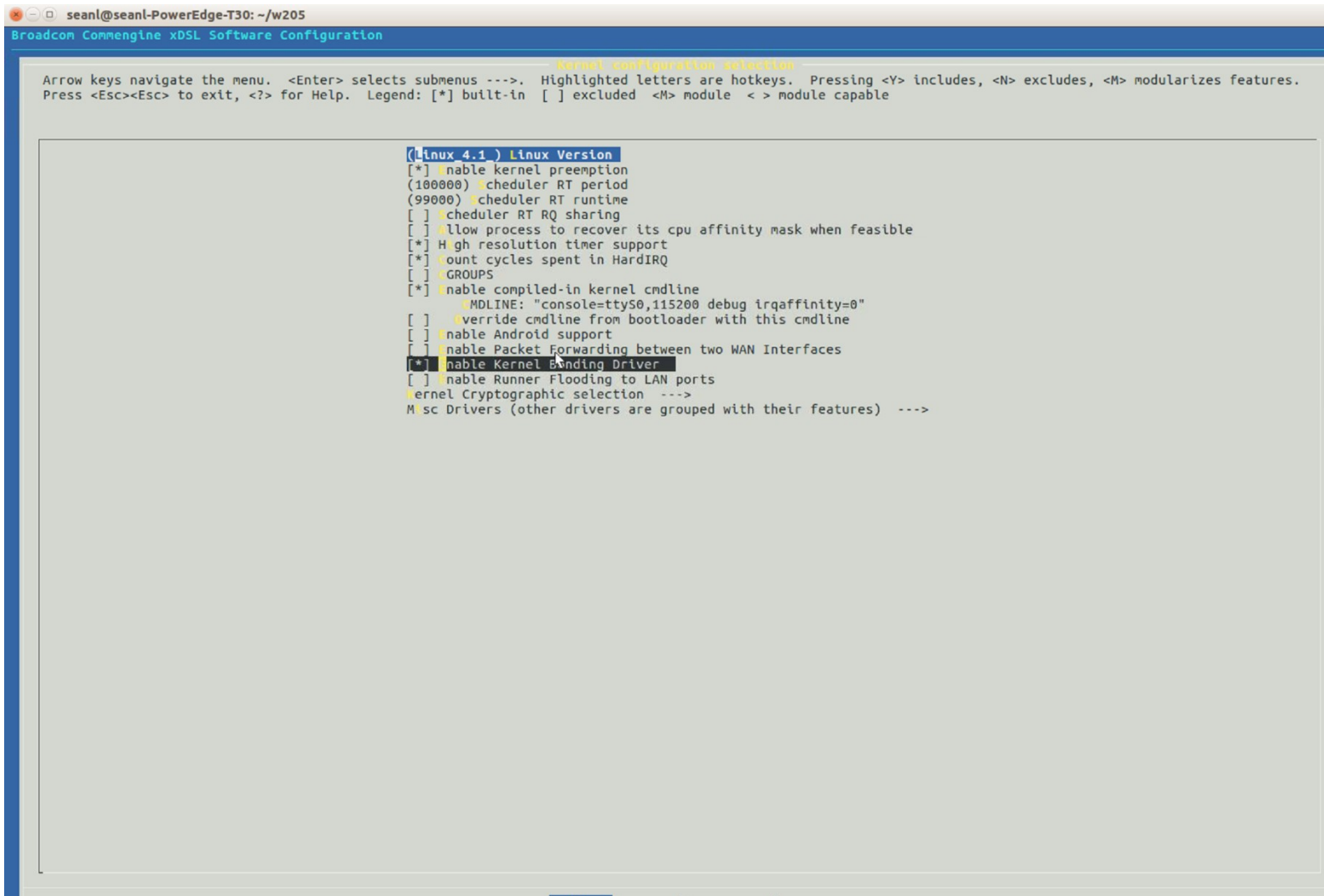


Figure 1.

Once in the Ethernet section, the following selection, the “Eth LAG - Ethernet Link Aggregation” is on the third items and use space to enable/disable this feature.

4 Data Model and Supported Features

Bonding driver could be configured in several modes to do load balancing (packet transmission):

Available Modes are:

Round-Robin (RR)

Transmit network packets in sequential order from the first available network interface (NIC) slave through the last. This mode provides load balancing and fault tolerance.

802.3AD (LACP) Dynamic link aggregation

Creates aggregation groups that share the same speed and duplex settings. Utilizes all slave network interfaces in the active aggregator group according to the 802.3ad specification. This mode is similar to the XOR mode above and supports the same balancing policies. The link is set up dynamically between two LACP-supporting peers.

XOR (balance-xor)

Transmit network packets based on a hash of the packet's source and destination. The default algorithm only considers MAC addresses (layer2). Newer versions allow selection of additional policies based on IP addresses (layer2+3) and TCP/UDP port numbers (layer3+4). This selects the same NIC slave for each destination MAC address, IP address, or IP address and port combination, respectively. This mode provides load balancing and fault tolerance.

802.3AD/LACP Configuration:

Mode (Active or Passive): In Active mode, ports will start to transmit LACPDUs as soon as link comes up. In Passive mode, the port won't send LACPDUs until it receives from the partner which is the driver default mode. In practice, Active mode is the most useful mode and should be configured in Linux Bonding driver by setting “all_slaves_active=1”

Aggregation Selection Logic: In general, LAG must be performed using same type of ports (speed, duplex). Aggregation Selection Logic means if there are multiple ports with different characteristics (speed, duplex), what should be the basis to aggregate the ports. Linux Bonding driver supports three link aggregation criteria: 0=stable (default), 1=bandwidth, 2=count.

Link Monitoring: In order to detect the link failures, speed and duplex; Bonding driver must be able to monitor the link using standard ETHTOOL or MII IOCTLS. Bonding driver must be configured with desired link monitoring interval using “miimon” parameter otherwise the default 100ms is used. Bonding driver could be configured with following transmit policies to use (if applicable for selected mode of operations):

layer2 XOR (MACDA + MACSA + EtherType)

layer 2+3 (IP ^ MAC)

layer 3+4 (IP ^ (TCP || UDP))

encap 2+3 (IP ^ MAC) relies on skb_flow_dissect to obtain the header fields.

encap 3+4 (IP ^ (TCP || UDP)) relies on skb_flow_dissect to obtain the header field

Following are supported in TR181-2-12 Ethernet LAG standard parameters with a few Broadcom defined parameters

Device.Ethernet.LAG. {i}.	object	W	<p>Ethernet Link Aggregation Group (LAG) table (a stackable interface object as described in [Section 4.2/TR-181i2]). Table entries model the Link Aggregation Sub-Layer as defined in [802.1AX] and [802.3ad]. It is expected that a <i>LAG</i> interface can only be stacked above <i>Ethernet.Interface</i> interfaces. The CPE can reject creation of additional LAG instances if this would exceed its capabilities.</p> <p>At most one entry in this table can exist with a given value for <i>Alias</i>, or with a given value for <i>Name</i>, or with a given value for <i>MACAddress</i>. On creation of a new table entry, the Agent MUST choose initial values for <i>Alias</i>, <i>Name</i> and <i>MACAddress</i> such that the new entry does not conflict with any existing entries. The non-functional key parameters <i>Alias</i> and <i>Name</i> are immutable and therefore MUST NOT change once they've been assigned.</p>	-	2.12
Enable	boolean	W	<p>Enables or disables the interface.</p> <p>This parameter is based on <i>ifAdminStatus</i> from [RFC2863].</p>	-	2.12

Status	string	-	<p>The current operational state of the interface (see [Section 4.2.2/TR-181i2]). Enumeration of:</p> <ul style="list-style-type: none"> • <i>Up</i> • <i>Down</i> • <i>Unknown</i> • <i>Dormant</i> • <i>NotPresent</i> • <i>LowerLayerDown</i> • <i>Error</i> (OPTIONAL) <p>When Enable is <i>false</i> then <i>Status</i> SHOULD normally be <i>Down</i> (or <i>NotPresent</i> or <i>Error</i> if there is a fault condition on the interface).</p> <p>When Enable is changed to <i>true</i> then <i>Status</i> SHOULD change to <i>Up</i> if and only if the interface is able to transmit and receive network traffic; it SHOULD change to <i>Dormant</i> if and only if the interface is operable but is waiting for external actions before it can transmit and receive network traffic (and subsequently change to <i>Up</i> if still operable when the expected actions have completed); it SHOULD change to <i>LowerLayerDown</i> if and only if the interface is prevented from entering the <i>Up</i> state because one or more of the interfaces beneath it is down; it SHOULD remain in the <i>Error</i> state if there is an error or other fault condition detected on the interface; it SHOULD remain in the <i>NotPresent</i> state if the interface has missing (typically hardware) components; it SHOULD change to <i>Unknown</i> if the state of the interface can not be determined for some reason.</p> <p>This parameter is based on <i>ifOperStatus</i> from [RFC2863].</p>	-	2.1 2
Alias	string(64)	W	<p>A non-volatile unique key used to reference this instance. Alias provides a mechanism for a Controller to label this instance for future reference.</p> <p>The following mandatory constraints MUST be enforced:</p> <ul style="list-style-type: none"> • The value MUST NOT be empty. • The value MUST start with a letter. • If the value is not assigned by the Controller at creation time, the Agent MUST assign a value with an "cpe-" prefix. <p>The value MUST NOT change once it's been assigned.</p>	-	2.1 2
Name	string(64)	-	The textual name of the LAG interface as assigned by the CPE.	-	2.1 2
LastChange	unsignedInt	-	The accumulated time in <i>seconds</i> since the interface entered its current operational state.	-	2.1 2
LowerLayers	string(1024)	W	<p>Comma-separated list (maximum list length 1024) of strings. Each list item MUST be the Path Name of an interface object that is stacked immediately below this interface object. If the referenced object is deleted, the corresponding item MUST be removed from the list. . See [Section 4.2.1/TR-181i2].</p> <p><i>LowerLayers</i> must reference to Device.Ethernet.Interface instances where Link Aggregation Group is configured by the CPE.</p> <p>For example, "Device.Ethernet.Interface.1, Device.Ethernet.Interface.2"</p>	-	2.1 2
MACAddress	string(17)	W	[MACAddress] MAC address of the Link Aggregation Interface.	-	2.1 2
X_BROADCOM_COM_LastChange	unsignedInt	R	A timestamp, in seconds, of the last status change. Used to calculate the LastChange value.		

X_BROADCOM_COM_Upstream	boolean	W	User selects this EthLAG to be used as WAN or LAN. Upstream TRUE means WAN interface. This param mirrors the upstream param in the lowest layer of the interface stack.	
X_BROADCOM_COM_EthIfName1	string	W	First Ethernet Interface Name.	
X_BROADCOM_COM_EthIfName2	string	W	Second Ethernet Interface Name.	
X_BROADCOM_COM_Mode	string(64)	W	<p>Specifies one of the bonding policies. Possible values (based on bonding driver configuration) are:</p> <p>balance-rr</p> <p>Round-robin policy: Transmit packets in sequential order from the first available slave through the last.</p> <p>This mode provides load balancing and fault tolerance.</p> <p>balance-xor</p> <p>XOR policy: Transmit based on the selected transmit hash policy.</p> <p>The default policy is a simple</p> <p>[(source MAC address XOR'd with destination MAC address XOR packet type ID) modulo slave count].</p> <p>Alternate transmit policies may be selected via the xmitHashPolicy option, described below.</p> <p>This mode provides load balancing and fault tolerance.</p> <p>802.3ad</p> <p>IEEE 802.3ad Dynamic link aggregation. Creates aggregation groups that share the same speed and duplex settings.</p> <p>Utilizes all slaves in the active aggregator according to the 802.3ad specification.</p> <p>Slave selection for outgoing traffic is done according to the transmit hash policy,</p> <p>which may be changed from the default simple XOR policy via the xmitHashPolicy option</p>	

X_BROADCOM_COM_XmitHashPolicy	string(64)	W	<p>Selects the transmit hash policy to use for slave selection in balance-xor, 802.3ad modes. Possible values are:</p> <p>layer2 Uses XOR of hardware MAC addresses and packet type ID field to generate the hash. The formula is hash = source MAC XOR destination MAC XOR packet type ID slave number = hash modulo slave count. This algorithm will place all traffic to a particular network peer on the same slave. This algorithm is 802.3ad compliant.</p> <p>layer2+3 This policy uses a combination of layer2 and layer3 protocol information to generate the hash. Uses XOR of hardware MAC addresses and IP addresses to generate the hash. The formula is hash = source MAC XOR destination MAC XOR packet type ID hash = hash XOR source IP XOR destination IP hash = hash XOR (hash RSHIFT 16) hash = hash XOR (hash RSHIFT 8) And then hash is reduced modulo slave count. If the protocol is IPv6 then the source and destination addresses are first hashed using ipv6_addr_hash. This algorithm will place all traffic to a particular network peer on the same slave. For non-IP traffic, the formula is the same as for the layer2 transmit hash policy. This policy is intended to provide a more balanced distribution of traffic than layer2 alone, especially in environments where a layer3 gateway device is required to reach most destinations. This algorithm is 802.3ad compliant.</p> <p>layer3+4 This policy uses upper layer protocol information, when available, to generate the hash. This allows for traffic to a particular network peer to span multiple slaves, although a single connection will not span multiple slaves. The formula for unfragmented TCP and UDP packets is hash = source port, destination port (as in the header) hash = hash XOR source IP XOR destination IP hash = hash XOR (hash RSHIFT 16) hash = hash XOR (hash RSHIFT 8) And then hash is reduced modulo slave count. If the protocol is IPv6 then the source and destination addresses are first hashed using ipv6_addr_hash. For fragmented TCP or UDP packets and all other IPv4 and IPv6 protocol traffic, the source and destination port information is omitted. For non-IP traffic, the formula is the same as for the layer2 transmit hash policy. This algorithm is not fully 802.3ad compliant. A single TCP or UDP conversation containing both fragmented and unfragmented packets will see packets striped across two interfaces. This may result in out of order delivery. Most traffic types will not meet this criteria, as TCP rarely fragments traffic, and most UDP traffic is not involved in extended conversations. Other implementations of 802.3ad may or may not tolerate this noncompliance.</p> <p>encap2+3 This policy uses the same formula as layer2+3 but it relies on skb_flow_dissect to obtain the header fields which might result in the use of inner headers if an encapsulation protocol is used. For example this will improve the performance for tunnel users because the packets will be distributed according to the encapsulated flows.</p> <p>encap3+4 This policy uses the same formula as layer3+4 but it relies on skb_flow_dissect to obtain the header fields which might result in the use of inner headers if an encapsulation protocol is used. For example this will improve the performance for tunnel users because the packets will be distributed according to the encapsulated flows.</p>		
-------------------------------	------------	---	--	--	--

X_BROADCOM_COM_LacpRate	string	W	<p>Option specifying the rate in which link partner is asked to transmit LACPDU packets when Mode is 802.3ad.</p> <p>Slow Request partner to transmit LACPDU every 30 seconds</p> <p>Fast Request partner to transmit LACPDU every 1 second</p>		
X_BROADCOM_COM_SelectionLogic	string	W	<p>Active Aggregator Selection Logic when Mode is 802.3ad.</p> <p>Stable</p> <p>The active aggregator is chosen by largest aggregate bandwidth. Reselection of the active aggregator occurs only when all slaves of the active aggregator are down or the active aggregator has no slaves.</p> <p>Bandwidth</p> <p>The active aggregator is chosen by largest aggregate bandwidth. Reselection occurs if:</p> <ul style="list-style-type: none"> - A slave is added to or removed from the bond - Any slave's link state changes - Any slave's 802.3ad association state changes - The bond's administrative state changes to up <p>Count</p> <p>The active aggregator is chosen by the largest number of ports (slaves). Reselection occurs as described under the "bandwidth" setting, above. The bandwidth and count selection policies permit failover of 802.3ad aggregations when partial failure of the active aggregator occurs. This keeps the aggregator with the highest availability (either in bandwidth or in number of ports) active at all times.</p>		

X_BROADCOM_COM_Miimon	unsignedInt	W	Specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each slave is inspected for link failures. A value of zero disables MII link monitoring. A value of 100 is a good starting point. The use <code>_carrier</code> option, below, affects how the link state is determined. See the High Availability section for additional information. The default value is 0		
Device.Ethernet.LAG.{i}.Stats.	object	-	Throughput statistics for this interface. The CPE MUST reset the interface's Stats parameters (unless otherwise stated in individual object or parameter descriptions) either when the interface becomes operationally down due to a previous administrative down (i.e. the interface's Status parameter transitions to a down state after the interface is disabled) or when the interface becomes administratively up (i.e. the interface's Enable parameter transitions from <i>false</i> to <i>true</i>). Administrative and operational interface status is discussed in [Section 4.2.2/ TR-181i2].	-	2.1 2
BytesSent	unsignedLong	-	[StatsCounter64] The total number of bytes transmitted out of the interface, including framing characters.	-	2.1 2
BytesReceived	unsignedLong	-	[StatsCounter64] The total number of bytes received on the interface, including framing characters.	-	2.1 2
PacketsSent	unsignedLong	-	[StatsCounter64] The total number of packets transmitted out of the interface.	-	2.1 2
PacketsReceived	unsignedLong	-	[StatsCounter64] The total number of packets received on the interface.	-	2.1 2

ErrorsSent	unsignedInt	-	[StatsCounter32] The total number of outbound packets that could not be transmitted because of errors.	-	2.12
ErrorsReceived	unsignedInt	-	[StatsCounter32] The total number of inbound packets that contained errors preventing them from being delivered to a higher-layer protocol.	-	2.12
UnicastPacketsSent	unsignedLong	-	[StatsCounter64] The total number of packets requested for transmission which were not addressed to a multicast or broadcast address at this layer, including those that were discarded or not sent.	-	2.12
UnicastPacketsReceived	unsignedLong	-	[StatsCounter64] The total number of received packets, delivered by this layer to a higher layer, which were not addressed to a multicast or broadcast address at this layer.	-	2.12
DiscardPacketsSent	unsignedInt	-	[StatsCounter32] The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.	-	2.12
DiscardPacketsReceived	unsignedInt	-	[StatsCounter32] The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being delivered. One possible reason for discarding such a packet could be to free up buffer space.	-	2.12
MulticastPacketsSent	unsignedLong	-	[StatsCounter64] The total number of packets that higher-level protocols requested for transmission and which were addressed to a multicast address at this layer, including those that were discarded or not sent.	-	2.12
MulticastPacketsReceived	unsignedLong	-	[StatsCounter64] The total number of received packets, delivered by this layer to a higher layer, which were addressed to a multicast address at this layer.	-	2.12
BroadcastPacketsSent	unsignedLong	-	[StatsCounter64] The total number of packets that higher-level protocols requested for transmission and which were addressed to a broadcast address at this layer, including those that were discarded or not sent.	-	2.12
BroadcastPacketsReceived	unsignedLong	-	[StatsCounter64] The total number of received packets, delivered by this layer to a higher layer, which were addressed to a broadcast address at this layer.	-	2.12
UnknownProtoPacketsReceived	unsignedInt	-	[StatsCounter32] The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol.	-	2.12

5 CMS Ethernet LAG WebUI Configuration

To configure a WAN service based on Ethernet LAG interface, one of the Ethernet port has to be enabled as WAN port (eth0 in most of the boards). Select “Advanced Setup/ETH Interface” and select eth0:

Device Info

Advanced Setup

Layer2 Interface

ATM Interface

PTM Interface

ETH Interface

ETH LAG

WAN Service

LAN

VPN

NAT

Security

Parental Control

Quality of Service

Routing

DNS

DSL

UPnP

DNS Proxy

Print Server

DLNA

Storage Service

Interface Grouping

IP Tunnel

IPSec

Certificate

Power Management

Batteries

Multicast

Diagnostics

Management

ETH WAN Interface Configuration

Choose Add, or Remove to configure ETH WAN interfaces.
Allow one ETH as layer 2 wan interface.

Interface/(Name)	Connection Mode	Remove
eth0/eth0	VlanMuxMode	<input type="checkbox"/>

Remove

Now eth0 WAN port is ready to be one of the WAN Ethernet LAG interface. To create WAN ETH LAG, select “Advanced Setup/ETH LAG” and Add button:

Device Info**Advanced Setup****Layer2 Interface****ETH LAG****WAN Service****LAN****VPN****NAT****Security****Parental Control****Quality of Service****Routing****DNS****UPnP****DNS Proxy****Print Server****DLNA****Storage Service****Interface Grouping****IP Tunnel****IPSec****Certificate****Power Management****Multicast****Diagnostics****Management****Ethernet LAG Interface Setup**

Choose Add, Remove a LAG Interface over 2 selected Ethernet ports.

LAG interface name	Ethernet port 1	Ethernet port 2	Mode	Lacp Rate	Xmit Hash Policy	MiiMon	Remove
--------------------	-----------------	-----------------	------	-----------	------------------	--------	--------

There will be two lag interface bond0 and bond1 available to select and once Ethernet ports are bonded and enabled, the bond interface can be used as WAN and LAN interface. When Add button is pressed, the following web page shows:

Device Info
Advanced Setup
Layer2 Interface
ETH LAG
WAN Service
LAN
VPN
NAT
Security
Parental Control
Quality of Service
Routing
DNS
DSL
UPnP
DNS Proxy
Print Server
DLNA
Storage Service
Interface Grouping
IP Tunnel
IPSec
Certificate
Power Management
Batteries
Multicast
Diagnostics
Management

Ethernet LAG Settings

Select WAN or LAN Ethernet LAG Interface

Select First Ethernet Interface (WAN port if existed)

Select Second Ethernet Interface

Select Ethernet LAG Mode

Select LAG Xmit Hash Policy

Select LAG LACP Rate

Select LAG MII Link Monitoring in ms

If Apply/Save button is pressed with the above parameters, the WAN bond0 interface is created with eth0 (WAN Ethernet port) and eth1 with the following:




- Device Info
- Advanced Setup
 - Layer2 Interface
 - ETH LAG
 - WAN Service
 - LAN
 - VPN
 - NAT
 - Security
 - Parental Control
 - Quality of Service
 - Routing
 - DNS
 - DSL
 - UPnP
 - DNS Proxy
 - Print Server
 - DLNA
 - Storage Service
 - Interface Grouping
 - IP Tunnel
 - IPSec
 - Certificate
 - Power Management
 - Batteries
 - Multicast
- Diagnostics
- Management

Ethernet LAG Interface Setup

Choose Add, Remove a LAG Interface over 2 selected Ethernet ports.

LAG Interface	LAG Type	Eth port 1	Eth port 2	Mode	Lacp Rate	Xmit Hash Policy	MiiMon	Remove
bond0	WAN	eth0	eth1	802.3ad	slow	encap3Plus4	100	<input type="checkbox"/>

To configure a WAN Service over bond0, just select “Advanced Setup/WAN Service” to select bond0/bond0 as the layer 2 interface and following the same configuration as if bond0 is same as EthWan (eth0/eth0) or xDsl (PTM/ATM).

 **BROADCOM®**
connecting everything®

Device Info

Advanced Setup

Layer2 Interface

ETH LAG

WAN Service

LAN

VPN

NAT

Security

Parental Control

Quality of Service

Routing

DNS

DSL

UPnP

DNS Proxy

Print Server

DLNA

Storage Service

Interface Grouping

IP Tunnel

IPSec

Certificate

Power Management

Batteries

Multicast

Diagnostics

Management

WAN Service Interface Configuration

Select a layer 2 interface for this service

Note: For ATM interface, the descriptor string is (portId_vpi_vci)
For PTM interface, the descriptor string is (portId_high_low)
Where portId=0 --> DSL Latency PATH0
portId=1 --> DSL Latency PATH1
portId=4 --> DSL Latency PATH0&1
low =0 --> Low PTM Priority not set
low =1 --> Low PTM Priority set
high =0 --> High PTM Priority not set
high =1 --> High PTM Priority set

bond0/bond0 ▼

BackNext

Following is a configured IPoE WAN service based on bond0:

Wide Area Network (WAN) Service Setup

Choose Add, Remove or Edit to configure a WAN service over a selected interface.

Interface	Description	Type	Vlan8021p	VlanMuxId	VlanTpid	Igmp Proxy	Igmp Source	NAT	Firewall	IPv6	Mld Proxy	Mld Source	Remove	Edit
bond0.1	cpe-ipintf-2	IPoE	N/A	N/A	N/A	Disabled	Disabled	Enabled	Disabled	Disabled	Disabled	Disabled	<input type="checkbox"/>	Edit

Add Remove

To test the above WAN Eth LAG, you could have to another board with a WAN service (VIA EthWAN, Cable or xDSL) and configure a Eth LAG LAN bonding interface over LAN ports such as eth1, eth2 and connect two Ethernet ports.

To see the working IPoE WAN service statistics, select “Device Info/Statistics/WAN Service”:

Device Info

Summary

WAN

Statistics

LAN

WAN Service

xTM

xDSL

Route

ARP

DHCP

CPU & Memory

Flow statistics

Advanced Setup

Diagnostics

Management

Statistics -- WAN

Interface	Description	Received								Transmitted							
		Total				Multicast		Unicast		Broadcast		Total				Multicast	
		Bytes	Pkts	Errs	Drops	Bytes	Pkts	Pkts	Pkts	Pkts	Pkts	Bytes	Pkts	Errs	Drops	Bytes	Pkts
bond0.1	cpe-ipintf-2	60549799	50412	0	8	7488	130	50278		4		2689511	27620	0	0	0	0
																27620	0

Reset Statistics