636x xDSL/XTM Bonding

Information Note

Rev 0.8, November 19, 2010

Revision Control

Version	Date	Reason	Author
0.1	May 1, 2009	Initial Version	Srini Seshadri
0.2	June 3, 2009	Updated with build & test section	Srini Seshadri
0.3	Jan 27, 2010	ATM & PTM Bonding updated	Srini Seshadri
0.4	Feb 21, 2010	ATM & PTM bonding versions/xdsl phy information/test platform update/cms support description/sections updated.	Srini Seshadri
0.5	Feb 25, 2010	Limitations/Restrictions/adsl ptm bonding mode descriptions updated. Description about bonding and preemption	Srini Seshadri
0.6	Sep 3, 2010	Descriptions, all the sections updated, as per features/fixes/upgrades available in 4.06L03/4.04L02Plus Patch fixes. Performance section updated, ptm bonding QoS support details added.	Srini Seshadri
0.7	Oct 20, 2010	Bonding, SMP support & Optimizations	Srini Seshadri
0.8	Nov 19, 2010	Auto switching between Bonding & Non-bonding modes	Srini Seshadri

1 Table of Contents

REVISION CONTROL 2

1 TABLE OF CONTENTS 3

2 ADDRESSED REQUIREMENTS 4

3 OVERVIEW 6

4 REFERENCES 7

5 PTM BONDING 8

5.1 IMPLEMENTATION 8

Rx Algorithm Minimal Outline 8

6 IP QOS & BONDING 11

SW Scheduling/Arbitration Releases 11

SW solution impact 11

7 PREEMPTION & BONDING 13

8 ATM BONDING 15

9 SMP (SYMMETRIC MULTI PROCESSING) & PREEMPTION SUPPORT FOR BONDING 16

10 BONDING & NON-BONDING MODE SWITCH 17

11 GENERAL LIMITATIONS 18

12 BUILD & TEST 19

13 PERFORMANCE 21

14 ENHANCEMENTS 22

2 Addressed Requirements

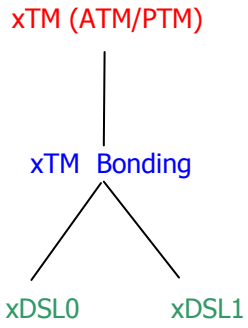
Bonding support 636x platform addresses PTM bonding (G.998.2 and Section 61 of 802.3ah-2004-(w-PTM6465).pdf) as well as ATM bonding (G.998.1) scheme requirements. Some of these requirements are derived from HW support/limitations in the system.

1. Only one 636x chip will be used for bonding.
2. PTM bonding & ATM Bonding solutions are available now as of 4.04L02/4.06L03 xDSL CPE SW releases.
3. It is highly recommended to upgrade to the latest version 4.06L03 for a much stable and functional version of bonding. However, there are patches available for 4.04L02 for some of the extended features of bonding with more robustness.
4. Maximum number of bonding groups supported is 2 if dual latency is selected, otherwise 1. For PTM bonding, there can be only one group due to HW limitation. So we will either support fast path group/interleaved path group. For ATM Bonding, current support tested & verified is single-latency bonding. Dual latency is not supported by any of BRCM CO platform so the CPE solution cannot be verified currently.
5. TR-159 – bonding management framework with MIB is currently supported in the datapath. Control plane routines can easily be added to implement it fully.
6. A bonding group can have atmost 2 members. More than 2 members is applicable only for ATM bonding due to HW support, and if market demands it, >2 member bonding can be supported for ATM. Current support is only 2 channels per bonded group.
7. Bonding control protocols (BACP for PTM bonding) will be added at a later stage. PTM bonding is provisioned statically and ATM bonding is provisioned dynamically through g.998.1 standard. ATM bonding thro ASM messages are already supported.
8. PTM bonding is based on HW Tx assist and SW based Receive functionalities.
9. ATM bonding is based on HW support in data path & SW support in control path. (XTM runs ATM bonding control protocol (through ASM messages) for the purpose of provisioning bonding.
10. It should be possible to enable/disable bonding on the fly. When disabled, the device will work in a normal fashion as is the case now. Currently it is not supported. Reboot is required for the bonding/normal control switching. This will be addressed in 4.10L01 release.
11. Effort is made to make the bonding/non-bonding look similar from provisioning and configuration perspective.
12. Different flavors of bonding (atm/ptm) are supported on the fly with no reboot.

13. Differential rates/delay variances of the order of atleast **4:1 & 2 ms** respectively, in both the directions.
14. Quickest recovery in both the directions of traffic under line status change conditions (either one or both the lines) basically less amount of loss of data should be seen under error conditions.
15. IP QoS provisioning for non-bonding/bonding.
16. VDSL PTM performance requirements for Quad-band. (40US/80DS) for all packet sizes.
17. ADSL ATM/PTM performance requirements for the line aggregate rates. (2US/54DS) for all packet sizes.
18. VDSL ATM mode for bonding will not be our focus unless the market case necessitates otherwise.

3 Overview

Bonding is to be logically layered as follows.



Bonding is implemented as an XTM enhanced functionality. Decision behind this approach are due to the following.

1. From system perspective, bonding functionality is transparent to the layers above xTM. This gives an advantage of non-widespread implementation (which would otherwise take larger amount of time to implement and verify) as well as ability to use flow cache mechanisms for the normal/bonded data paths thus achieving greater performances.
2. Userspace layers are aware of bonding to the extent to support TR-159 and to be aware of WAN side constituting more than 1-link with XTM/XTM bonded devices. Rest of the software including linux is not aware of xDSL bonding which gives us less dependency over the other system SW.
3. XTM layer deals closely with xDSL links and the transport modes/connections/flows/interface to SAR hardware layer which are necessary for XTM bonding layer as well so it it logical to make use of the existing infrastructure provided in XTM which will be seemless.
4. More importantly, ATM/PTM transport modes will either run over single port/bonded dual ports, transparently. Transitioning from normal mode to bonded mode and vice versa should be possible in future.
5. TR-159 specifies this architecture as well as the other ITU-T G.998 standards.
6. Bonding layer deals mostly with fragments/packets/cells with SIDs (Sequence IDs) and the higher layers deal with packets (ENET/AAL5).
7. Less dependency on the system gives time-to-market advantage for the solution.

4 References

For bonding information, refer to, (this section links is internal to BRCM and is not accessible to customers.) This sections has standards documents for xDSL bonding from which the BRCM solution is implemented.

\\brcm-irv\dfs\projects\comengine\sw-docs-depot\Bonding\

As well as,

6368 register data base file & data sheets. (The data sheet customer version present in docSafe as well)

For IPV6/IPV2 image, latest versions should be used. We recommend using version(s) 10_5_9 or latest.

The image is, *8lco/*12lco binary files for IPV6.

For IPV2 image, "*tco" image should be used.

Ipv2 supports atm bonding over ADSL.

IPV6 supports ptm bonding over xDSL.

Performance (.xls) sheets will be available shortly in the same links here.

5 PTM Bonding

Most of the implementation is in the XTM Configuration/Network drivers and the userspace control to manage bonding in the system.

5.1 Implementation

6368 uses single latency bonding/dual latency bonding. For single latency, ports 0/1 are to be used and for dual latency ports 0/2 for path0 and ports 1/3 for path1 to be used.
Implementation uses this premise of the HW support in the system in the SW as well.

Normal path is preserved almost "as is" and the bonded path is done based on the configuration that XTM driver gets during its initialization. PXTM_INITIALIZATION_PARMS.
This configuration is used in the XTM configuration driver and network drivers to do the tasks specific to bonding layer. The tasks that are noteworthy are,

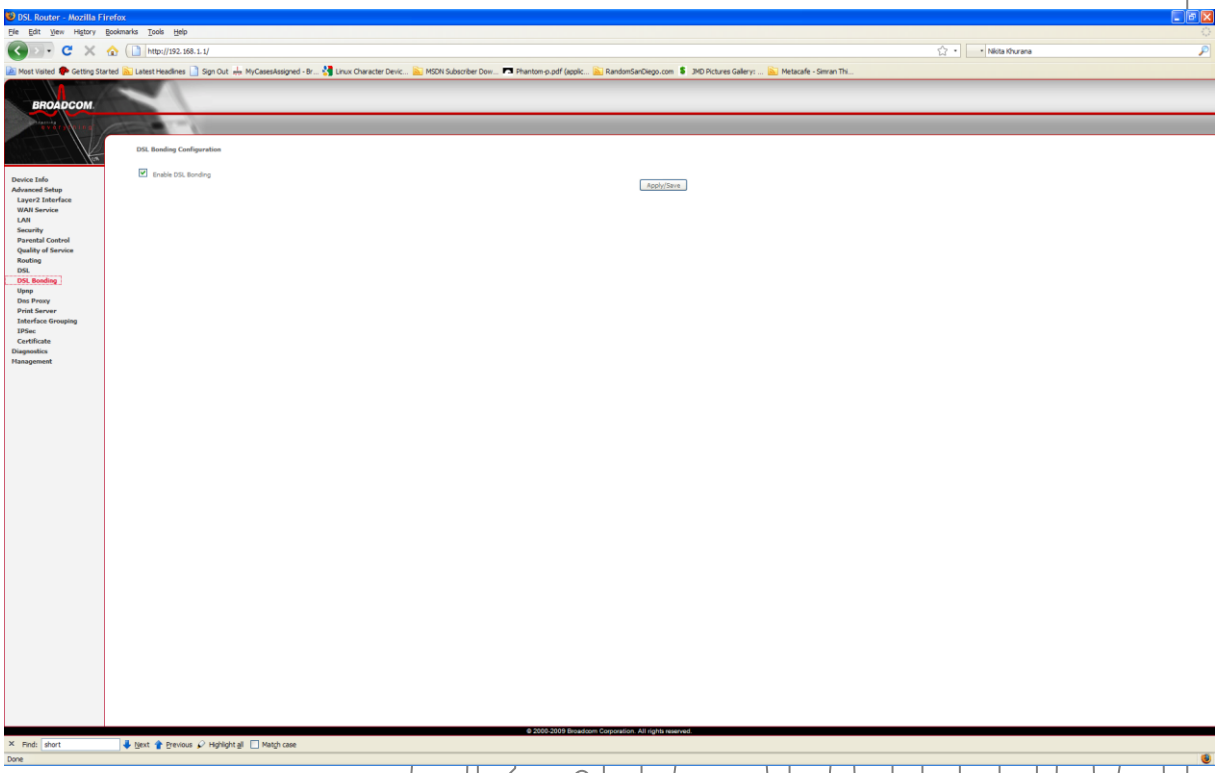
1. When management actions take place on a given port (say a connection is created on port 0), its bonded port (port1 for single latency, port 2 for dual latency) is associated with the all the resources, such as interfaces, connections, shapers inside the XTM layer and vice versa for when the connection is removed (the bonded member is disassociated in this case). This happens automatically for bonding without the knowledge of the user layer. This is applicable for stats/status as well.
2. HW gets configured to be in bonded mode (tx side only for PTM & both the tx/rx sides for ATM) and HW tx bondig is assist based which involves SW intervention per packet flow which is done part of the XTM network driver Hard transmit routine. This implements the TX assist functionality for PTM bonding as per the data sheet section 8, \\brcm-irv\dfs\projects\bsehwdsl\projects\BCM6368\data_sheet\DSXX6CT\636XCT-DSXX6-RI_nr.pdf (customers should refer to version of the 6368 data sheet in their docSafe accounts for the HW support for information)
3. Rx side is fully implemented in SW as part of bcmdrivers/opensource/net/xtmrt/impl1/bcmxtmrtbond.c and the implementation is as per 802.3 ah standard spec as far as reassembly of the fragments & further processing goes, which is outlined here.

Rx Algorithm Minimal Outline

a) Determine the nextFragmentSequenceNumber via the algorithm.
b) If the nextFragmentSequenceNumber is equal to the expectedFragmentSequenceNumber, process
that fragment and continue to step c). If (nextFragmentSequenceNumber is less than

expectedFragmentSequenceNumber) or (all the active port queues are non-empty and nextFragmentSequenceNumber ¯ expectedFragmentSequenceNumber) or (any port queue has been non-empty for maxDifferentialDelay bit times without any fragment being processed) follow
the fragment sequence error handling rules, before returning to normal
fragment processing.
c) Accept the fragment into the fragment buffer. If (accepting the fragment into the fragment buffer
causes a frame length overflow) or (the fragment is an unexpected start of packet) or (the fragment is
an unexpected end of packet) or (the fragment has the StartOfPacket bit deasserted when the start of
a new packet is expected) then follow the error handling procedures. Else if
that fragment is an end-of-packet, pass the packet to the linux/flow cache.
d) Increment (modulo 2^14) the expectedFragmentSequenceNumber.
e) Repeat processing.

4. Interface with xDSL driver for link status/management/configuration actions back and forth for all the bonded links.
5. PTM Bonding Tx side implements weighted round robin scheduling for fragments on the available ports. Different rates/delays among xDSL lines are supported here.
6. For userspace, WebUI control for enable/disable bonding system-wide is introduced (in the Advanced Setup/DSL bonding page) and the default action is to have this disabled. This will be combined with PHY handshake results to determine the course of operation being normal mode/bonded mode later in the future which will be more robust. It is imperative to reboot the system to effect this configuration, as it is not foced currently.
7. Also for userspace, CMS supports PTM bonding only with 4.04L01. From 4.04L02 onwards, we are adding atm bonding support to CMS. Idea is to create WAN devices in the data model for atm bonding/ptm bonding modes and associate them with atm/ptm devices which are primary. In normal mode, it would either be atm/ptm type and in bonded mode, it will be atm-atm bonded/ptm-ptm bonded devices. Now CMS can support wan device atm/ptm/atmbonding/ptm bonding on xDSL side.



8. Userspace has SUPPORT_DSL_BONDING flag used in the code to do the tasks specific to bonding. Driver/kernel space uses this flag only to build xDSL image differently as there are different codes for normal/bonding mode for xDSL PHY. Otherwise, no flag of any sort is used in XTM/anywhere in kernel space.

6 IP QoS & Bonding

Currently 6368 CPE supports IP QoS in ATM non-bonded/bonded and PTM non-bonded modes. IP QoS can be supported using strict priority and Weighted Fair Queueing scheduling disciplines in standalone mode or mixed-discipline mode(s). These SP/WFQ disciplines are done in 6368 HW for all the above modes. User should be able to define multiple queues for all these modes and define a IP QOS classification/modification/output behavior for the queues using the CPE Web UI.

For PTM bonding mode, the current 6368 HW supports a different kind of scheduling/arbitration based on ptm priorities/latencies and not based on strict priority/WFQ disciplines. So, in PTM bonding mode, there can be only 4 Tx Queues in HW each catering to PTM Low-fast/PTM High-fast/PTM Low interleaved/PTM High-interleaved. There is round-robin arbitration across these 4 queues in this mode and they are individual flows not related to each other. In fact they have different sequencing on all of them. This allows to have multiple groups of PTM bonding based on the PTM priority/line latency characteristics.

IP QoS management relies on SP/WFQ disciplines across the Tx Queues among the related flows, such as it should be possible to map IP Priorities/fields to a given tx queue and activate the SP/WFQ/Mixed mode disciplines among them. For instance, all of these TX queues may map onto one single PTM flow (such as PTM low-fast or some other mode) and it should be possible to configure arbitration among these to provide the necessary QoS within a PTM flow among many queues.

As said earlier, the above IP QoS behavior is possible with all the modes except PTM bonded mode and BCM963268 addresses this functionality in HW for PTM bonded mode so that IP QoS can be entirely supported in HW for BCM96328.

For BCM6368 solutions, which require the above-mentioned IP QoS support in PTM bonded mode within a bonded group, the arbitration/scheduling disciplines have to be supported in SW in coordination
with the existing HW disciplines.

SW Scheduling/Arbitration Releases

The implementation is in the XTM kernel drivers and take effect for the PTM bonded mode. When the system flows are in modes other than PTM bonded mode, the HW support will be used. Current plan is to carry this out in 2 phases, the SP scheduling/arbitration is already available currently with 4.06L03/4.04L02 Plus patches. This will be followed by another delivery by 8/30 (including tests) with WFQ and Mixed mode Scheduling/arbitration for PTM bonded mode.

SW solution impact

As you see the impact will be on the flows/tests that are carried out on PTM bonded mode and will

not have any impact on the other modes. It should also be pointed out that all these modes are backward compatible with the existing codebase and will have minimal impact to the existing flows.
The impact in PTM bonded mode is only on multiple US flows and no impact on DS flows whatsoever.
The new XTM drivers will be compatible with the future releases of the BCM CPE SW as well. It is to be noted that currently all the services run on one flow of the PTM (PTM low/high – 1 latency) with 1 Tx Queue. This configuration will have very low impact with the new SW solution as opposed to having multiple PTM flows with multiple Tx queues on US side.
Net result is that for the user/configuration manager to use the existing IP QoS support through the WebUI available for the PTM bonded mode as well in a transparent manner so all the modes will look alike the same for the user as far as the management of services/IP QoS etc.,

7 Preemption & Bonding

The preemption is in HW and it would be automatic given there are 2 flows (high, low) in US be in bonding/normal mode.

But in bonding, every packet gets converted to fragments which then converted to codewords. Each fragment has got a sequence id with it and it is presumed that all the fragments come in sequence on a given line. No out of ordering on a given line. In preemption case, an high priority packet-fragment-codeword(s) can get interspersed between a given low prio packet-fragment-codewords on a given line. This creates a possibility that out-of ordering sequenced fragments can happen on a given line. Our implementation on Rx side do not have problem with this. But, a CO implementation on their Rx side (our upstream) may have an issue with this.
Even if this is ok, These preemptive codewords can travel on dual lines which makes it possible that preemption may/may not have an effect, as they may get delayed due to delay differentials between the lines. This is not defined in bonding spec clearly.

We may get around this by forcing all the fragments of a preemptive packet on a single line only, these are all implementation specific. Currently we do not have heuristics to send flows on only one line and it is always both the lines be used based on their available rates. When we do this, we have to take care of the fact about each flow rate. We can do it if it is required but will be our own implementation and as there is no standard.

Wrt the tests for this feature, the following configuration can be used i.e. testing B0/L0 + B0/L1 (1 channel on two lines) or B0/L0 and B1/L0 (2 channels [normal + preemption]) differ ?

```
Normal    queue -[bonding0]-line 0-[bonding0]->
Preempted queue -[bonding1]-line 1-[bonding1]->
```

Bonding agent 0 and 1 in this case are segmenting/reordering to/from a single line but you can test your configuration with preemption (we still miss the 4 queues config B0/L0, B1/L0, B0/L1, B1/L1 but we atleast tested B0/L0, B0/L1 (std bonding) and B0/L0, B1/L1 (preemption with bonding))

Current implementation has a switch in the bcmxt/implx/Makefile that can be turned on/off to enable the feature preemption with bonding. Default is that the switch (PTMBOND_US_PRII_TRAFFIC_SPLIT) is disabled.

8 ATM Bonding

Configuration is to have the bonding enabled globally and create the ATM interfaces as usual. This bonding runs a control protocol from xtmcfg driver (called ASMs, autonomous state messages) to manage bonded pairs. Currently only single latency is supported. HW forwarding through CMF is not used currently and HW bonding through SAR in both the directions are used for data path and we achieve aggregate line rates over ADSL dual links for all packet sizes.

It is necessary to create ATM interface in the usual manner for the bonding control protocol to work. Internal implementation for configuration is similar to that of PTM bonding section.

Entire data path is through SAR HW. XTM cfg configuration driver has been extensively modified to accommodate the ASM protocol for atm bonding. XTM network driver is least modified.

All the ATM functionalities are supported with the following limitations as of now:

1. CMF forwarding for ATM bonding is possible but will be part of the next phase. Even without CMF HW forwarding, we achieve line rates.
2. Dual latency bonding is possible with a compatible CO that supports it. Support exists but testing has not been done for dual latency bonding.
3. Multiple VC testing has not been done but we don't anticipate any issues in this domain.
4. Bonding fallback – The system will work in one/both the lines depending on the availability.

9 SMP (Symmetric Multi Processing) & Preemption Support for Bonding

From 4.09L02 release onwards, one of the goals is to ensure xDSL bonding mode(s) of operations are Linux kernel SMP & Preemption modes safe. With SMP, preemption enabled, tests were run to ensure the atm/ptm bonding functionalities run fine.

SMP facilitate dual core options and system processes can make use of this facility to be able to run in any available core. PTM bonding flow mostly resides in software and it consists of 3 different stages of processing. Preprocessing to queueing in 1 stage and SW scheduling in 2 stages, both in upstream and another stage in downstream direction. In upstream, it can clearly take advantage of the dual core options with SMP. The SW Scheduler component in bonding upstream side is offloaded to run in core 1 while the rest of the components (upstream preprocessing & downstream) run with core 0 with the other data path entities.

SW Scheduler runs close with HW SAR and implements scheduling/shaping functionalities and uses HW Timer Interrupt (Currently Timer 1 , Power management will use Timer 0 & Timer 2 supposedly) to ensure QoS characteristics.

With this approach, tests were run to see the benefit and utilizing both the cores for bonding achieves at least 40% more performance on an aggregate level with simple optimizations. In other words, SMP dual options fetch 40% more throughput than non-SMP single core option.

10 Bonding & Non-bonding Mode Switch

From 4.10L01 release onwards, the system will auto-switch between bonding and non-bonding modes depending on the CO configuration without any need for manual intervention. In previous releases, an alarm message was displayed on the CPE console to alert the user of any mis-configuration between CPE & CO (say CPE is in bonding mode and CO is in non-bonding mode or vice versa) and the user is required to take actions accordingly by configuring the mode status through WebUI and restarting the modem.

From 4.10L01 onwards, the system alerts of this condition and automatically switch to the CO mode internally and this is achieved by restart of the system eventually. This provides a greater benefit in the field that the CPE is able to auto-adopt according to the CO configurations and there is no need for any manaual interventions.

The restart trade-off is okay as when this happens (which is infrequent), it is typically at the time of service provisioning and the procedure is robust and quick that it is not noticeable greatly.

11 General Limitations

1. Only one bonding scheme will be supported at a time during run time. (Either ATM/PTM) Detection is automatic internally based on the xDSL handshake. Basically, no concurrent bonding schemes at CPE will be operational. (i.e atm bonding on certain ports and ptm bonding on certain ports)
2. PTM bonding supports only single latency path.
3. A bonding group can at most have 2 members.
4. Static provisioning of ptm bonding is supported. Dynamic provisioning using BACP will be considered in later times based on the market needs.
5. CMF based HW forwarding will not be applicable to PTM bonding and will be later considered for ATM bonding. (CM functionality in upstream is possible for PTM bonding and will be supported in later releases)
6. Bonding status (rates/lines) can be seen from the console logs. WebUI is also updated to show the xDSL statuses, as well as aggregate rate available.
7. 64 bytes/pkt US rate is limited to a maximum 20 Mbps rate when there is a line rate downstream traffic.
8. Barring the above limitation, there is also this one. Small size packet (64-127 bytes/pkt) performance for PTM bonding is limited to 20-3- Mbps US/80 Mbps DS. Bigger packet sizes can go up to 40 Mbps US/100 Mbps DS for data traffic rates.
9. PTM Preemption & Bonding is not defined in the spec clearly and in wide use as well. PTM preemption may not be usable with bonding as described in earlier sections.
10. VDSL ATM Bonding mode has not tested as we think the DSLAMs which support vdsl &bonding will also support ptm mode of operation, and hence PTM can be used which inherently has less overheads. However, we support ATM bonding for ADSL mode(s) due to legacy DSLAMs that are present.

12 Build & Test

- Profile 96368BGW (Bonding Gateway) is to build image that is also capable of bonding only from the 4.04L.02 production release onwards. The same image can also be used for normal mode with the global configuration through WebUI. If xDSL bonding capability is not needed at all, then this profile should not be used.
- The phy code is different for normal-multimode, vdsl-bonding mode, adsl-bonding modes of operations. The release code has all the above modes of PHY code in adsl directory. Users will need to select the option in DSL/XTM section of the profile through "make menuconfig" command. Default mode from the profile 96368BGW is to build vdsl-ptm bonding mode.
    1. For VDSL PTM bonding mode, set the DSL/XTM drivers option of the profile 96368BGW to multimode (default) / VDSL and clean build the image.
    2. For ADSL ATM/PTM bonding modes, set the DSL/XTM drivers option of the profile 96368BGW to ADSL and clean build the image.
- Test platform constitutes 96368MB2G CPE reference board and an IPV6/IPV2 DSLAM CO.
- For testing, it is recommended that CO version 10_6_15v8lco/10_6_15v12lc0 firmware as well as GUI10_6_2 or greater to be used which is relatively stable.
- CO Configuration (through GUI) (for IPV6)
    1. Select the front end to be BCm6526_ISL1556_IPV6
    2. Send Afe connect message
    3. Configure Bonding group parameters,
        - Lines (0,1)
        - Bonding group id – 0
        - Main line index – 0
        - Flags – 1 (for ptm bonding), flags – 0 (for atm bonding). For IPV6, the IWF (internetworking forward path) also needs to be configured from hmicli. (refer to Ipv6 instructions/manual)
    4. Configure the lines to 8a,8b,8c,8d profile only (bonding is supported only with this profiles) and no need to limit the rates.
    5. Start the lines for operation (Default is VDSL PTM here) Other modes also be started for other types of bonding.
    - Set all non vlan tag traffic be forwarded between fe and ge0 port
      Run commands by console port of IPDv6
      1. vlan add 1 pbm=ge1,fe ubm=fe
      2. pvlan set fe 1

After the commands, all the traffic between GPHY_0 and xdsl will all be non-tagged.

- CO Configuration (through GUI) (for IPV2)
    1. Select the front end to be BCM reference design v2.
    2. Configure Bonding group parameters,
        - Lines (0,1)
        - Bonding group id – 1
        - Main line index – 0
        - Flags – 0 (for atm bonding)
    3. Start the lines for operation (Default is ADSL ATM here)
    4. Configure the ATM AAL5 vc and forwarding table information through external lcaconfig utility.
- CPE Configuration
    1. No special configuration for normal mode
    2. For first use, you may need to **wipe out the flash** for running the image built through 96368BGW profile.
    3. Configure DSL bonding to enabled for bonding mode & **reboot**. This control is available under "Advanced Setup" page of modem WebUI.
    4. Configure rest of the transport (atm, ptm, services, bridge/MER/PPPOE etc.,)
    5. Lines will come up automatically displaying the status on the console, service will be up and ready for packet (AAL5/PTM) transport. (in bonded mode)
    6. CPE side debugging is possible through ifconfig as well as **/proc** pages in **/proc/driver/xtm** directory commands. More debugging is available PTM_BONDING_DEBUG flag in the XTMRT make file, which is disabled by default.
    7. ATM Bonding ASM protocol implementation is present in bcmdrivers/broadcom/char/xtmcfg/impl1/xtmasmhandler.cpp for reference.
    8. For ADSL-PTM normal/bonding modes, the following actions need to be done (at CO & ADSL bonding CPE, as for adsl mode, default is always ATM).
        - Disable VDSL in CO line configuration / Test tab / Disable ITU-G993-2 initialisation -- 1/both lines
        - Enable ADSL PTM TPS TC and prefer PTM in CO line configuration / Traffic -- 1/both lines
        - On CPE enable PTMoADSL for both lines:
            1. xdslctl0 configure --TpsTc 0xF
            2. xdslctl1 configure --TpsTc 0xF

13 Performance

All the performance metrics – excel sheets are kept in,
\\brcm-irv\dfs\projects\comengine\sw-docs-depot\Bonding\636x Bonding\Performance
folder (Internal,not accessible outside BRCM network).
They are kept with Release, DateStamp & each mode of bonding supported, so it should be easy to pick the one with the latest date for the current performance numbers.
All the numbers are taken with SmartApps application using smartbits traffic generator.
Currently all the numbers are taken in bridged mode. Routed mode numbers will be updated soon. Since flowcache is used, the routed/bridged numbers for bonding are expected to be close & similar.

In general,
For ADSL-ATM/PTM Bonding

1. we achieve wirespeed of the aggregate of the both the line rates for all packet sizes.
2. Connected rates are 2.5 Mbps US (aggr)/54.5 Mbps DS (aggr)

For VDSL- PTM Bonding

1. Connected rates are 40 Mbps US (aggr)/99 Mbps DS (aggr) – (20 US/50 DS each line)
2. Unidirectional traffic numbers are given in the excel sheet. US direction – CO is the bottleneck. DS direction – CPE is the bottleneck. Downstream direction, we achieve pretty close to the line aggregate rate(s).
3. Bidirectional traffic - We achieve 20-30/80 for small packet sizes of 64-127 bytes/pkt and 40/80 for all the rest of the packet sizes from 128-1514 bytes/pkt. (concurrent & manually run, verified, as smart apps does not allow asymmetrical bi-directional throughput testing).

14 Enhancements

None.