May 2022

Compliant Coherent Module Functional Management

Anonymous

Compliant Coherent Module Functional Management

ABSTRACT

The present disclosure relates to compliant coherent module functional management. As network operators disaggregate the purchasing of network elements and co-resident modular coherent technology, it is now important to control the behavior of the pluggable coherent module to match the known-good configuration that has been validated and approved for deployment. Network operators wish to have a single product usable for deployments generally across any/all network equipment. However, different network platforms place different and unique requirements on the pluggable modem. Such unique requirements must be verified and validated at the system level, and once verified, network operators would want to avoid or minimize making any changes which might invalidate such verification, causing additional testing, delays and deployment risk/cost.

DETAILED DESCRIPTION

Coherent modem modules built with identical hardware may be manufactured with different firmware. The combination of hardware and firmware define the total product, and system verification would be considered a function of both hardware and firmware.

Pluggable coherent modems (a class of coherent modem modules that are pluggable by users in the field) are designed to operate in accordance with published standards and MSAs, and as such are manageable by the host platform (e.g. a router network element). The third-party host platform (i.e. host platform vendor different from the pluggable coherent modem vendor) could manipulate the firmware of the module to suit the local environment, and if the module was moved from one platform to another, each platform would carry the obligation to program the plug to suit. This places a large obligation on the third-party platform vendors to implement firmware management, which would be costly and potentially problematic to verify as they do not develop and own the firmware of the pluggable coherent modems.

Standard electrical interfaces on coherent pluggable modules do not anticipate any management connection other than a local low-speed MDIO or I2C (bit banger) standards compliant interface; only the host platform can directly communicate with the pluggable module. While this might be acceptable to widly used “gray” client plugs, it provides challenges to smart coherent colored plugs. A centrally deployed controller could, if provided with a direct connection to the pluggable module, directly manage the firmware operating on any plug in any third-party host platform. However, the lack of such connectivity prevents this today.

The use of either in-band our out-of-band management channels are well known and an industry standard way of configuring network elements, and/or downloading software/firmware to a network element or host platform. Configuration of pluggable coherent modems is today considered the role of the hosting platform through a CMIS

type of standard interface. A network controller wishing to configure the pluggable coherent modem would have to communicate with the third-party host platform to effect any change on the pluggable coherent modem. As such, this approach is not known to be available. Platform independent configuration of a pluggable modem, via the use of an in-band comms channel included in a client traffic payload is also not novel. The new problem is, how to enable the sale of a pluggable modem, based on common hardware but with time-varying firmware, into a customer that needs bespoke configurations depending on the use case. Some host platforms may automatically update plug firmware to suit themselves. That is a workable solution however the plug is modified by this action and made unusable in other platforms. Hence the need for dynamic configuration management at the plug level.

It is proposed that significant value can be generated for network operators by combining the following elements into a solution for centralized management of the firmware running on industry compliant coherent modules:

1. Pluggable coherent modems with integrated management connectivity delivered to the module embedded in the Client ethernet payload traffic.
2. A Centralized software controller which is able to logically (e.g. through IP connectivity) communicate with the coherent pluggable module.
3. The ability by the centralized controller to detect the configuration of the coherent module (firmware, hardware, provisioning information, default behaviors etc.).
4. The ability of the centralized controller to distribute to the coherent module the validated configuration (via a target firmware load or database) that is required for any specific module based on at least, but not limited to, the third-party host platform.
5. The ability of the centralized controller to automate such compliant module functional management independently of the third-party host platform.

The present invention provides the use of the in-band (logical channel) client side management channel with the notion of a centralized software controller to solve the emerging commercial problem of how to deploy a common module (built with optimized firmware) into a network operator who wants to use the same module in any system. To accomplish this, the firmware must be pushed to the module independently from the host platforms and managed to match the unique requirements.

The coherent pluggable modem, enabled by 7nm silicon technology, is the first low power pluggable technology capable of having a full high performance processor complex embedded in the module. Such processors are easily capable of IP based communication; however they are impaired by the lack of ethernet I/O capability in the socket. It is not industry practice to include an Ethernet port on a pluggable module. Ethernet traffic is however present as client traffic to the module. Modern technology includes an ability to extract and inject specifically addressed (MAC, VLAN or IP) packets from the client traffic flow, enabling direct communication with the DSP. In effect, module management traffic can ride along with the client payload, bypassing the I/O constraint. The communication can be secured using known techniques such as SFTP, SSH, and other techniques of the like.

Using this mechanism, the pluggable modem can communicate with a centrally deployed network controller. This communication path now enables the remote controller to effectively manage all aspects of the plug such as,

- Firmware
- Configuration
- Client Provisioning
- Multiplexing provisioning
- Optical provisioning
- All forms of monitoring and logs
- Alarm monitoring
- Inventory

Specific to this application, the configuration and firmware management enables the deployment of a common hardware pluggable module into any use case. Generally, configuration, provisioning, and monitoring are valuable in the overall solution.

The following Figures are presented to depict examples of deployment. Figure 1 is an example of Integrated Deployment, Figure 2 is an example of issues with Disaggregated/Multi-Vendor Deployment, and Figure 3 is an example of the present invention Compliant Coherent Module Functional Management in a Disaggregated / Multi-Vendor Deployment Environment, wherein the labels are as follows,

1. Firmware Upgrade
2. Management Data
3. Management Interface
4. Data Interface
5. DSP to extract and inject specifically addressed packets from the client traffic flow to DSP for control and management.

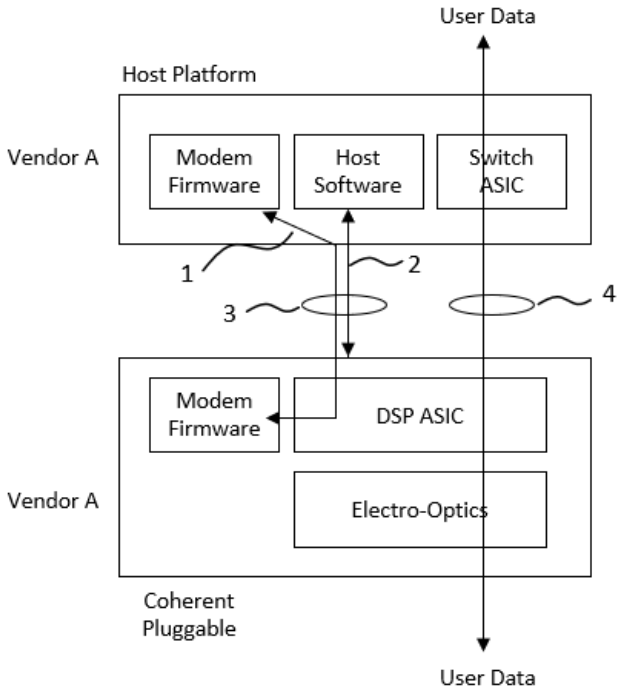

Figure 1

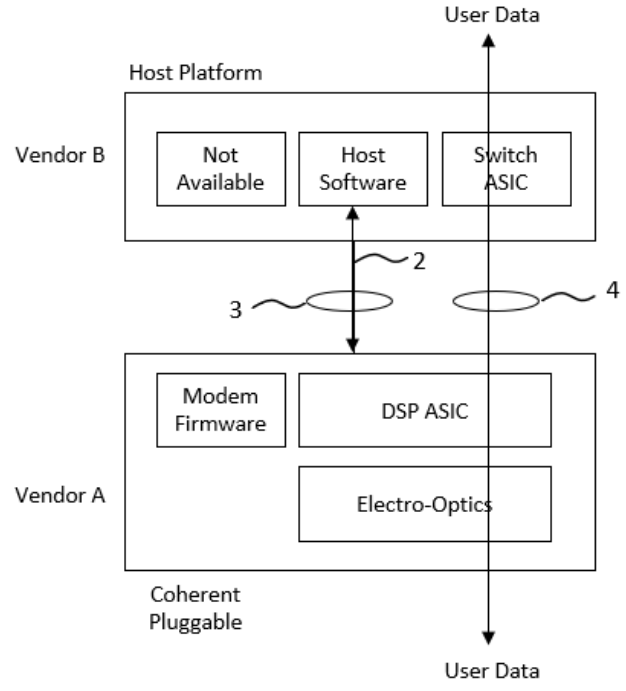

Figure 2

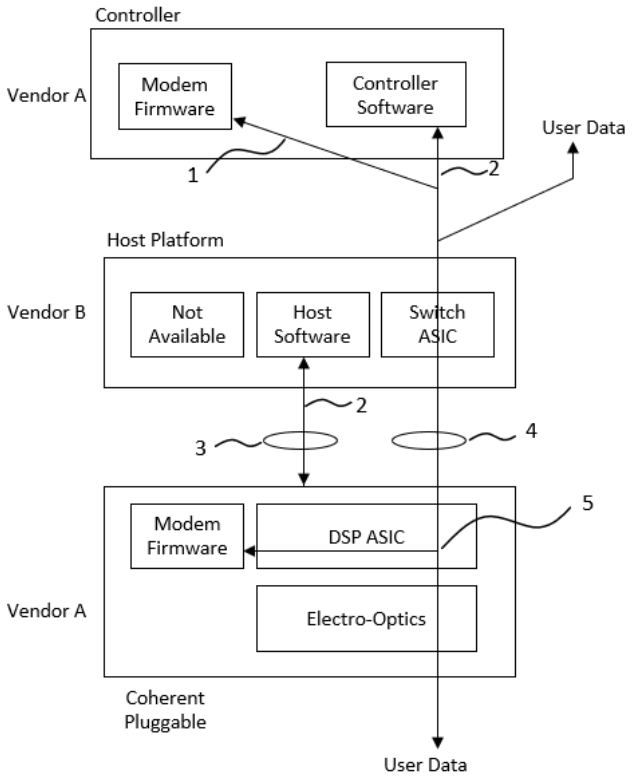

Figure 3

It will be appreciated that some embodiments described herein may include one or more generic or specialized processors ("one or more processors") such as microprocessors, digital signal processors, customized processors, and Field-Programmable Gate Arrays (FPGAs) and unique stored program instructions (including both software and firmware) that control the one or more processors to implement, in conjunction with certain non-processor circuits, some, most, or all of the functions of the methods and/or systems described herein. Alternatively, some or all functions may be implemented by a state machine that has no stored program instructions, or in one or more Application-Specific Integrated Circuits (ASICs), in which each function or some combinations of certain of the functions are implemented as custom logic. Of course, a combination of the aforementioned approaches may be used. Moreover, some embodiments may be implemented as a non-transitory computer-readable storage medium having computer-readable code stored thereon for programming a computer, server, appliance, device, etc. each of which may include a processor to perform methods as described and claimed herein. Examples of such computer-readable storage mediums include, but are not limited to, a hard disk, an optical storage device, a magnetic storage device, a ROM (Read Only Memory), a PROM (Programmable Read-Only Memory), an EPROM (Erasable Programmable Read-Only Memory), an EEPROM (Electrically Erasable Programmable Read-Only Memory), Flash memory, and the like. When stored in the non-transitory computer-readable medium, the software can include instructions executable by a processor that, in response to such execution, cause a processor or any other circuitry to perform a set of operations, steps, methods, processes, algorithms, etc.

Although the present disclosure has been illustrated and described herein with reference to preferred embodiments and specific examples thereof, it will be readily apparent to those of ordinary skill in the art that other embodiments and examples may perform similar functions and/or achieve like results. All such equivalent embodiments and examples are within the spirit and scope of the present disclosure.