How the NXP ASK Embedded Networking Router and Software Converts the LS1043A Into a Smart NIC or Router

Product Owner: Marlan Winter, NXP Semiconductors USA, Inc.

When NXP crafted the routing software stack called the applications solution kit (ASK), the requirements for this routing stack had a fairly short list of difficult primary concerns.

The list of high-level goals, often seemingly conflicting, are:

- To perform routing and switching at fixed function router latency and throughputs for any packet size when using an NXP system on chip (SoC) with packet and security acceleration
- To use a well-known router stack that the industry has already accepted
- To support well-known and documented methods of setting up Linux® networking
- To utilize NXP packet and security accelerators (PFE, DPAA 1 &2, SEC) to create a "near-single-chip"programmable smart router with nearly ALL of the general-purpose processor (GPP) cycles available to programmers for value-added applications—not just pushing packets
- To provide a rapidly reconfigurable routing system that fits well with potential SDN-ish deployments

Underlying these goals are some simple industry truths:

- Processing packets in software only is very slow and is sometimes actually called the "slow path." This is particularly acute in Linux.
- Processing packets in a neat and organized software hierarchy, which roughly corresponds to the OSI 7-layer model is VERY slow. This is particularly acute in Linux.
- You can process packets 100% in software, but if you can bypass the layered software stack, the packet throughput and latency dramatically improve. This is particularly acute in Linux.
- Bypassing the layered software stack dramatically reduces CPU utilization, even if there is 100% software-only processing. This is particularly acute in Linux.
- Hardware processing, instead of software processsing, is well known in the industry to yield improvements in latency, throughput, power consumption, and CPU loading.

The OpenWRT embedded router software stack

The basis of the ASK is a good industry-standard embedded router stack. While the choices, particularly in the open-source space, aren't terribly plentiful, there are some good ones.

NXP chose the OpenWRT software stack for its good traction within the industry and its alignment with how NXP creates and maintains software.

OpenWRT advantages customers inherit by using the ASK

- NXP provides all the source. Should they choose to do it, customers can cross-check and manage versions, updates, patches and more without NXP, but such a decision would be subject to making sure that the fast path system isn't compromised by changing something in the kernel.
- OpenWRT has GUI configuration interfaces and a command line interface API for controlling the embedded router. Customers can use this without any NXP intervention. ASK note: NXP does not create custom GUI or user interfaces for customers as we have found this is best left to customers or GUI developers. NXP stops at the platform level.
- Because of the API, particularly the command line API, customers can integrate their management and control software to our ASK. NXP has found that many of our performance-conscious embedded networking customers have a mature and well-developed management and control concept. Therefore, customers need only integrate their software rather than try to change to NXP.
- NXP build and deploy for our Linux-based software works with OpenWRT's build and deploy methodology which makes integrating into a software design flow fairly routine.
- OpenWRT has a great ecosystem and many user space apps are available. NXP's idea with the ASK is to help customers get to a high-performance platform and customers put their differentiation on the platform. (NOTE: NXP has done "just-add-branding-only" routers for customers too – but we don't widely advertise this.)
- One of the primary benefits of the Linux networking stack is that it has been optimized for features and code readability, but it has not been optimized for performance. Hence it supports all kinds of protocols but has bad throughput and latency.

Packet and security acceleration capabilities

NXP has a long history of creating high-performance semiconductors for moving packets.

One key innovation from NXP was the creation of small hardware peripherals connected to NXP CPUs, such as those based on Arm®v8 and Power Architecture® cores.

NXP created a peripheral rather than standalone packet management because we found that, for the bulk of the applications NXP wanted to help customers create, they needed a combination of a CPU and packet management. In other words, most of our customers wanted to add intelligence (e.g., a smart network interface card + smarts, a smart router, and a smart switch such as a layer-3 switch) to their networking enabled application. Not every customer wanted every smart feature, so creating a networking SoC from a general-purpose processor and networking or packet management hardware seemed prudent. Finally, when we coupled security with the system, we wanted the device to work in a pipeline so that the accelerators could work together with minimal general-purpose CPU intervention.

The main takeaway for this section is that the CPU and accelerators are meant to work together. The OpenWRT system NXP created does exactly this, even though Linux itself has a long heritage of software-only functions for processing packets.

Marrying the OpenWRT stack and the SoC containing packet and security acceleration

NXP’s task was two-fold: 1) keeping OpenWRT as true to its original form as possible (i.e., not making a bunch of exceptions to how it worked), and 2) uniting OpenWRT to the NXP packet engines so that customers received all the benefits of high throughput, low latency, low power, and routine, industry-standard configuration. The ConnTrack module is one of the key open-source modules that enables uniting OpenWRT and NXP packet engines.

There is one more small piece of background information needed before the theory of operation for the ASK can make sense. The NXP packet engines, (PFE, DPAA 1, and DPAA 2) are designed to directly accept an internet packet from a PHY, place it in memory, manage it without modifying DDR memory, analyze it, munge it if required, then terminate the packet, or forward it based on the packet’s rules to the correct output port/PHY, all without much CPU intervention, if any.

A more technical person would say that the ASK uses the standard Linux networking control plane and that this is an all-software and familiar methodology for system administrators and network operators. The NXP “magic” applies to the dataplane. Software for control plane and NXP hardware for dataplane along with the NXP ASK makes this possible.

Offloading the dataplane (called CPU offloading) is a suitable way to keep the comprehensive feature set of Linux and the OpenWRT distribution while delivering the higher performance of hardware packet management and freeing the CPUs for other tasks. Imagine paying for a 4-core Arm® SoC, but getting the equivalent capability of a 24-core system—with throughputs and capacity that a 24-core system could never achieve (this is for illustrative purposes - contact NXP for exact performance numbers).

This setup works PHY to PHY or ethernet interface to ethernet interface. This near automatic packet processing can be done on these interfaces because the system is designed around ethernet and the inbuilt idea of ethernet packets. Other interfaces, PCIe®, SPI, UART, etc., are interfaces which bring in all different types of data—much of which is not in a ethernet packet format. For these interfaces to take advantage of the NXP hardware fast path defined by the NXP packet engines, these non-ethernet interfaces must have a CPU-based conversion path to jump their data to the fast path or back off again.

Operation theory for the ASK OpenWRT stack with hardware fastpath

When the first packet from a new connection (related stream of packets) arrives at the input of the ethernet interface, this packet is received into the hardware and the hardware signals the software, just like every single ethernet does.

This first packet progresses through Linux, as implemented by the OpenWRT distribution, all the way up to the Linux userspace. When it hits userspace, the Linux is setup to run the ConnTrack module.

The Conntrack module then matches the packet to the already setup standard Linux tables and, via NXP kernel modifications, sets up the NXP packet engines so that all the remaining packets belonging to that connection are handled 100% by the NXP system. This assumes of course that the packet is not terminated but forwarded. This is a 100% reasonable assumption in a router, switch, Wi-Fi gateway, gateway or other pass-thru device.

The CPU is not involved with any subsequent packets in the stream once this route is setup in the hardware. In many cases, NXP has seen the CPU utilization go below 5% and down to 1% if the number of first packets is modest. When the connection is finished, a reverse operation is performed and the hardware path is thereby disconnected.

There are some packet streams which need to be processed in the CPU and that is okay. If you can push 75% of the packets to the autonomous system, you have effectively turned your 4-core (in the LS1043A example) Arm Cortex®-A53 SoC into the equivalent of a 24-core Cortex-A53 SoC (performance comparison are for illustrative examples only). You would likely need that 24-core SoC if you did not have the packet engine. NXP systems have low power consumption numbers because our systems don’t need all those cores for managing the networking software stack. This is a big deal - provable by searching Google for “networking CPU offload”.

What about Wi-Fi? Does it come on PCIe?

PCIe isn't ethernet; therefore, the PCIe-connected Wi-Fi needs to come into PCIe and the data needs to be registered as received by Linux. Once it is received, the Wi-Fi driver provided by the Wi-Fi card manufacturer (which in turn received it from the Wi-Fi SoC manufacturer) is then activated.

NXP has worked over many years with many Wi-Fi chipsets and cards due to our ASK. Because of this history of cooperation, we have our own Wi-Fi center of excellence and can do things with Wi-Fi drivers that most organizations would struggle to do. In other words, we can do something with Wi-Fi drivers in a month or two that might take another organization with less Wi-Fi driver experience around 4-6 months to accomplish.

NXP knows how to modify the Linux and/or proprietary drivers for many Wi-Fi chipset manufacturers to intercept the Wi-Fi packet, rapidly convert it to ethernet format, and jam it into the packet fast path. If the packet is travelling in the opposite direction, we grab the packet from the fast path and convert to the Wi-Fi format for the driver so the packet can go to the radio.

NXP has seen CPU cycles dedicated to processing Wi-Fi dramatically decrease, sometimes by as much as 50%, but we usually see a 25-30% reduction in CPU Wi-Fi cycles.

Throughput increases to a level where the CPU is not the system performance bottleneck anymore. A little-known fact is that modern Wi-Fi radios can run over stout multicore systems pretty easily. In other words, economically achieving your full radio throughput may be challenging, unless, perhaps, you use the NXP SoC described here. Throughput increase challenges will probably worsen as modern radios, like 5G radios, increase in their capability. Fortunately, NXP understands how to meet these challenges successfully.

Finally, the latency on the Wi-Fi decreases to near hardware speeds (the CPU has to do a little bit of intercept). If you're making a home gateway and you want your customers to be able to do real gaming or augmented reality and the gateway uses Wi-Fi, consider the NXP ASK.

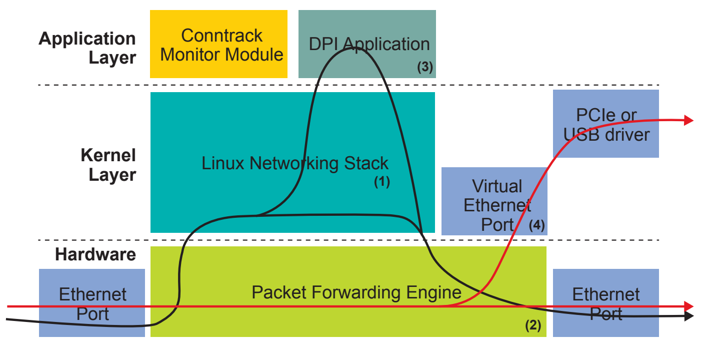
SDN API

We are seeing customers use our ASK and remotely configure their CPE over the internet or VLANs because there are ample APIs. In the ASK case, they aren't using an SDN API as strictly defined as OpenFlow, for example, but they are defining their network across 1000s of devices remotely via software.

Pictorial Representation of the ASK operation

Here is a pictoral description of the theory of operation:

1) Initial packets from a new flow pass through Linux® networking stack
2) Conntrack monitor module (CMM) programs the packet forwarding engine (PFE). All subsequent packets in that flow are automatically forwarded by the PFE
3) Optionally, packets are forwarded to a DPI application. Once identified, the flow can then be offloaded to PFE.
4) PFE also supports connections to Wi-Fi® or cellular modems via a Virtual ethernet port



How to Reach Us:

Home Page: **www.nxp.com**
Web Support: **www.nxp.com/support**

USA/Europe or Locations Not Listed:
NXP Semiconductors USA, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.nxp.com/support

Europe, Middle East, and Africa:
NXP Semiconductors Germany GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.nxp.com/support

Japan:
NXP Japan Ltd.
Yebisu Garden Place Tower 24F,
4-20-3, Ebisu, Shibuya-ku,
Tokyo 150-6024, Japan
+0120 950 032 (Domestic Toll Free)
https://www.nxp.jp/
https://www.nxp.com/support/support:SUPPORTHOME

Asia/Pacific:
NXP Semiconductors Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@nxp.com

www.nxp.com/ask

Document Number: ASKTHOFOPER&MWP REV 0