White Paper

CoSPs, MNOs
Telecommunications, Wireless Communications

Mavenir, Microsoft and Intel Team for Real-Time Layer 1 vRAN Control

Mavenir trials Microsoft's Janus Framework powered by Carrier Grade OS Azure Linux to provide customized deployment data and debugging for its Open RAN distributed units (DUs). System uses hardware acceleration built into 4th Gen Intel® Xeon® Scalable processors

MAVENIR

Microsoft

The open and virtualized architecture of Open RAN brings many innovations and cost advantages. Programmability of the network is one of the more important Open RAN advantages allowing for better network performance and support for advanced network services.

The recent popularity of RAN intelligent controller (RIC) has put a focus on the programmability advantages of Open RAN. Two types of RIC have been defined. The non-real time RIC (non-RT RIC) platform uses rApps to provide service management and orchestration capabilities. The near real time RIC (near-RT RIC) uses xApps to program network optimization actions that take place in near real time - between 10 milliseconds and one second in execution time.

This leaves a real-time (<10 ms) programmability and observability gap at layer 1 (PHY) resulting in no way to observe and act on data in a critical network layer that drives big impacts on network performance. Exposing layer 1 telemetry data makes the PHY more observable enabling troubleshooting, security, automation and network optimization.

Furthermore, layer 1 telemetry comprises detailed per-slot information and can consume significant bandwidth if transported elsewhere for processing.

Microsoft is addressing this challenge with its Janus Framework, a layer 1 real-time observability and control framework that includes RAN telemetry, analytics, and artificial intelligence (AI). Janus uses eBPF, a kernel programming technology, to extract layer 1 intelligence in real time.

With Janus, mobile network operators (MNOs) or their partners can load codelets inline in the RAN functions in a safe manner. This paper describes how Mavenir is trialing Janus to provide more visibility into its DU processes. The L1 component in the DU is provided by Intel® FlexRAN reference architecture running on Intel® architecture CPUs.

Table of Contents

Janus provides detailed RAN access through eBPF

To get real-time visibility Janus provides access to internal RAN structures in a similar way as eBPF provides access to internal kernel structures, allowing statically verified user programs to run in line with the RAN code. Janus leverages a user-mode eBPF implementation and extends it for the RAN use case, allowing third-party applications to collect and aggregate custom RAN data, and even implement real-time control policies. Janus adds some extra security features such as real-time preemptions. It also provides a standard way to export the data to the rest of the telemetry stack based on a framework shown in Figure 1.

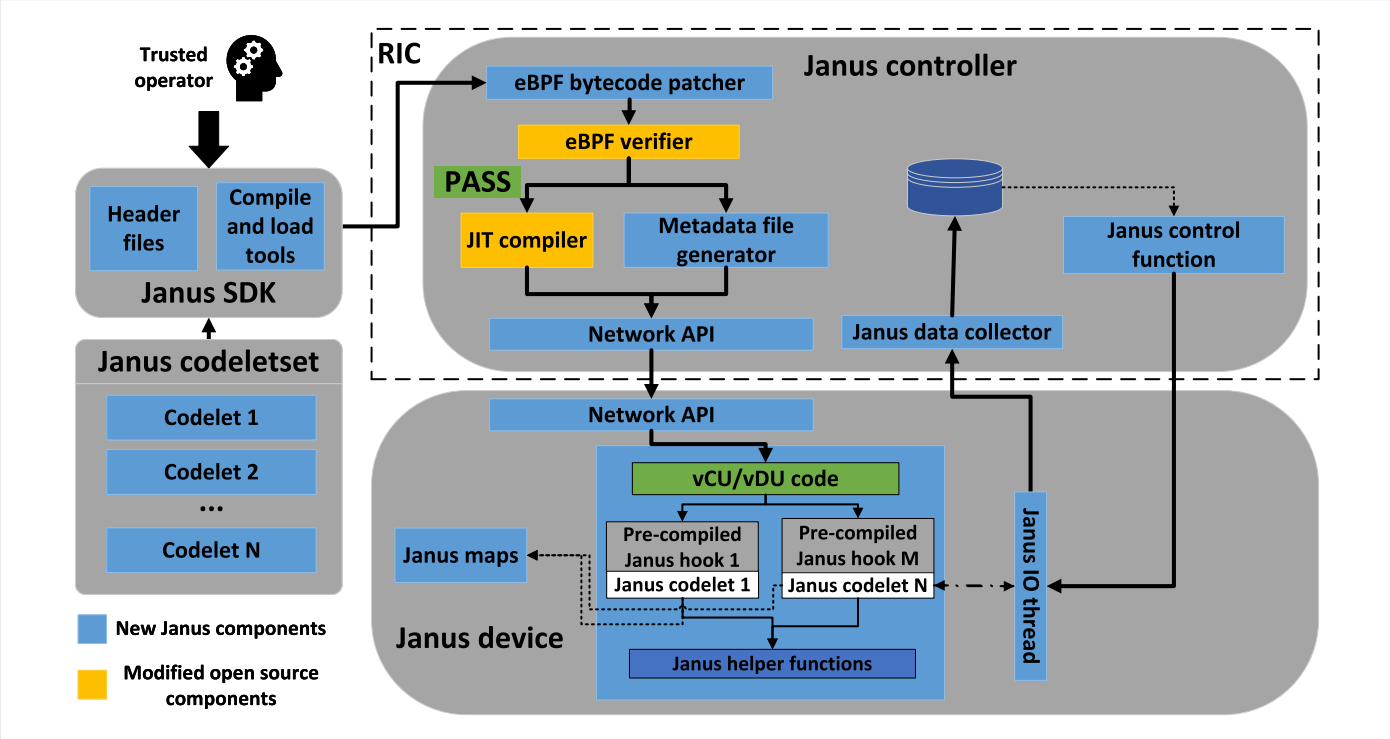

Figure 1. Janus Framework architecture showing codelets, devices and the controller.

The virtual environment is called a Janus device and can be a virtualized distributed unit (DU), virtualized central unit (CU) or any vRAN software function that allows execution of custom code. A Janus controller is also part of the solution and is responsible for controlling the Janus devices and codelet sets. Developers upload their codelet sets to the controller, with load/unload instructions for one or more Janus devices.

These codelets are designed to invoke Janus call points, or hooks, at selected places in vRAN software functions. When invoked, custom eBPF code gets read-only access to selected internal vRAN context, including various 3GPP-defined data structures and events. Codelets are written in C and compiled into eBPF bytecode.

Lastly, Janus adds a µs-level control over the execution latency to the codelets to adapt them to vRAN layer 1 requirements by preempting the software from exceeding a defined runtime threshold.

The static verification is extended to cover flexible output data structures along with an extension of the static verification to provide several optimizations to ensure a non-preemptible design in the fast path to minimize the impact of the codelets on vRAN performance.[1]

Mavenir Evaluates Janus for Production RAN Software Workload

Mavenir is a leading Open RAN software and technology provider and saw that it could use Janus with its DU to provide observability needed for customized reporting and debugging of initial deployment. The company began a months-long evaluation to work with the technology and incorporate it into its DU to provide observability required for customized reporting and automating the initial DU deployment.

Mavenir's DU is a containerized, Open RAN Alliance-compliant network function that is part of the Mavenir 5G NR Open vRAN solution. The DU interacts with the radio unit (RU) providing processing for lower-level networking up through the Packet Data Convergence Protocol (PDCP) layer of the protocol stack before sending digitalized radio signals into the network.

The Mavenir DU utilizes FlexRAN to process layer 1 data flows. Intel® FlexRAN™ Reference Architecture for Wireless Access is a baseband PHY reference design for 4G and 5G networks. It facilitates the implementation of cloud-based RAN functions, including baseband processing and radio resource management. FlexRAN is designed to provide a framework for building scalable, high-performance RAN solutions on Intel® Xeon® processors. For added performance, the software leverages the Intel® Advanced Vector Extensions for vRAN (Intel® AVX) instruction set.

In multiple Open RAN deployments with major operators, Mavenir sought a solution that provided flexible and on-site customizable real-time observability to debug the PHY and improve the DU's layer one instrumentation. Also, with various configurations of different operators' RAN networks, there has been an increasing need to support customizations in L1 processing in schedulers and massive MIMO channel estimation algorithms, etc. The Janus framework's ability to insert codelets for customization looks appealing in DU software development and operational support for on-site advanced debug capability.

When introducing a new framework into the production workload, one important consideration is the performance impact, especially if it is a high capacity, low latency RAN DU workload. Mavenir evaluated both L1 and L2 workloads integrated with Janus, and the performance impact for both was well within the acceptable range.

There are three components integrated into the workload: The Janus framework, the hooks, and the codelets. A hook is typically 1-2 lines of code embedded into the workload source code. An empty hook, i.e., a hook that does not have an associated codelet, introduces less than 1ns[2] execution time per hook; codelets with no-memory copy added the execution time of the codelets in the calling thread context; Janus framework runs a couple of low-priority threads consuming less than 0.01 CPU cores. The overhead and latency are within a typical L1 PHY processing latency requirement of 125 microseconds.

Security is another critical evaluation item. In addition to the secured codelet management and encrypted codelet load/unload mechanism, a guardrail preventing a codelet from consuming longer CPU cycles than expected is a must. Mavenir verified the Janus codelet verifier function, which enforces code preemption when the codelet runs exceeding the pre-configured CPU quota per codelet. The codelet verifier also checks each codelet for image integrity before loading. There are built-in security access features in the codelet verifier that can be configured to limit the codelet's access to DU software in read-only mode. All the above security measures protect the DU software from being compromised by codelet malfunction.

Expanded DU Debugging

With high capacity, low latency DU software, it's challenging to output checkpoint data as needed given the performance impact, e.g. a 20ms PCAP dump at the front haul interface may push the DU server to the CPU overload threshold. The standard software practice such as log level management, predefined probes with runtime CLI debug parameters can output KPIs to help debugging. However, each has its own challenges to provide flexible runtime operational data for DU debugging.

The current FlexRAN methodology is that when the Info Trace tool is triggered, the PHY and MAC buffers are streamed out to the network controllers where they are captured in a PCAP file and then to a packet analyzer such as Wireshark to stream the messages to the network controller. Figure 2 shows how this is done:

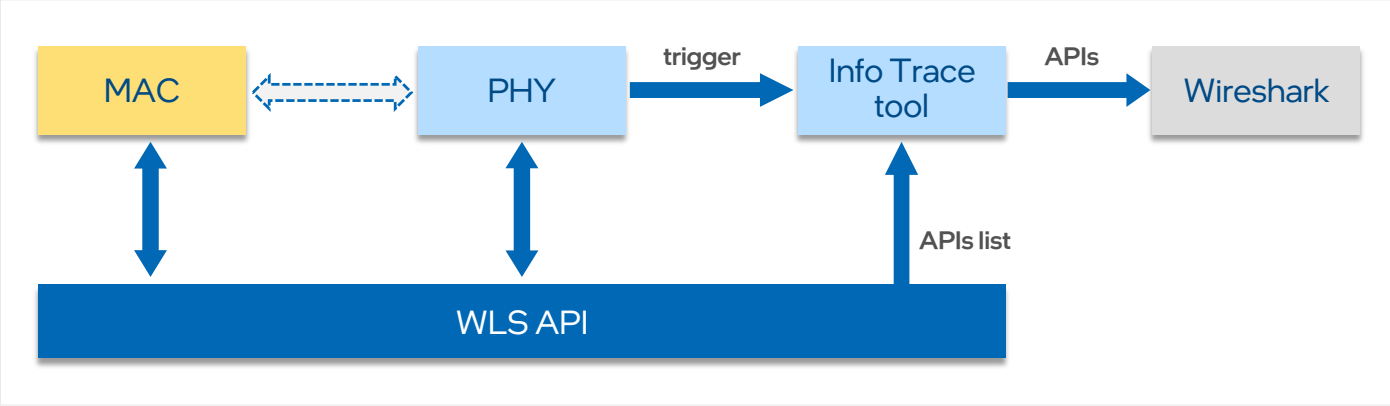

Figure 2. DU API debugging process.

The data is limited to SRS channel estimation and down link beam weights. But there is a lot more information that could help with debugging if it was observable.

The 3GPP Open RAN standard has built-in performance metrics that can be used for DU layer 1 instrumentation. However, these data are aggregated over an entire cell site and over a long interval. This broad data set makes it hard to pinpoint problems. More detailed data can be collected but this feature takes CPU cycles and can impact RAN performance. As a rule, this advanced data collection mode is used only when troubleshooting.

Mavenir created an advanced data extraction tool for RAN deployments that uses Janus to customize the individual datasets that the MNO wants to extract at runtime and to configure what time the data should be extracted. This customizable data extraction feature helps MNO DevOps teams improve their ability to control the network.

With Janus, Mavenir saw a way to easily add a new advanced debugging tool for on-site customizable DU instrumentation. It would allow advanced features such as threshold-triggered data extraction, configurable output data size, and correlated L1 and L2 data extraction, all customizable at operation run time without needing a workload rebuild or restarting the running system.

This advanced debugging tool can help address the need for an on-site IQ Sample extraction: In field DU deployment, it's challenging to debug the high UL BLER issue without the IQ Sample dump from L1. However, the intermittent nature of the high UL BLER makes it challenging to capture the IQ samples at the right time. A blanket IQ sample dump puts the DU in an overload condition. Depending on the targeted data, an L1 patch build with additional code is required. The site operation engineers have to restart the L1 to apply the new patch.

The Janus integrated debugging tool allows operation engineers to load selected codelets with site configurations (In this case, the configuration is 20,000 packets). The loaded codelet gets periodically called by the embedded hooks to check the SNR threshold. It will inform the real-time data collector thread to output the IQ samples when SNR crosses the threshold. Should a different KPI be desired as the threshold for the data extraction trigger, which usually needs to be decided on-site, the operation engineers can simply load the code with other parameters or, in some cases, a different codelet. There is no need to build the patch; neither is there a need to restart the L1.

In this use case, the hook is one line of code embedded in the L1 at development time; the codelet is a simple C-code-based software program with less than 100 lines of code running logic operation; the real-time data collector automatically outputs the extracted data to an external data analytics tool (Wireshark plug-ins are available) for root cause analysis.

By loading different codelets, Mavenir can expand to support other use cases such as interference detection, energy saving, L2 TTI stretch, etc. Janus empowers Mavenir's RAN development and deployment with unprecedented flexibility for improved time to market. The data extracted through the Janus framework can be transferred via the E2 interface for further AI/ML-powered RAN Intelligent Controller (RIC) processing.

With this instrumentation, MNOs can steer the RAN performance with insights and control, thus opening many opportunities for network optimization for better TCO.

Below is the block diagram of the tool mentioned above:

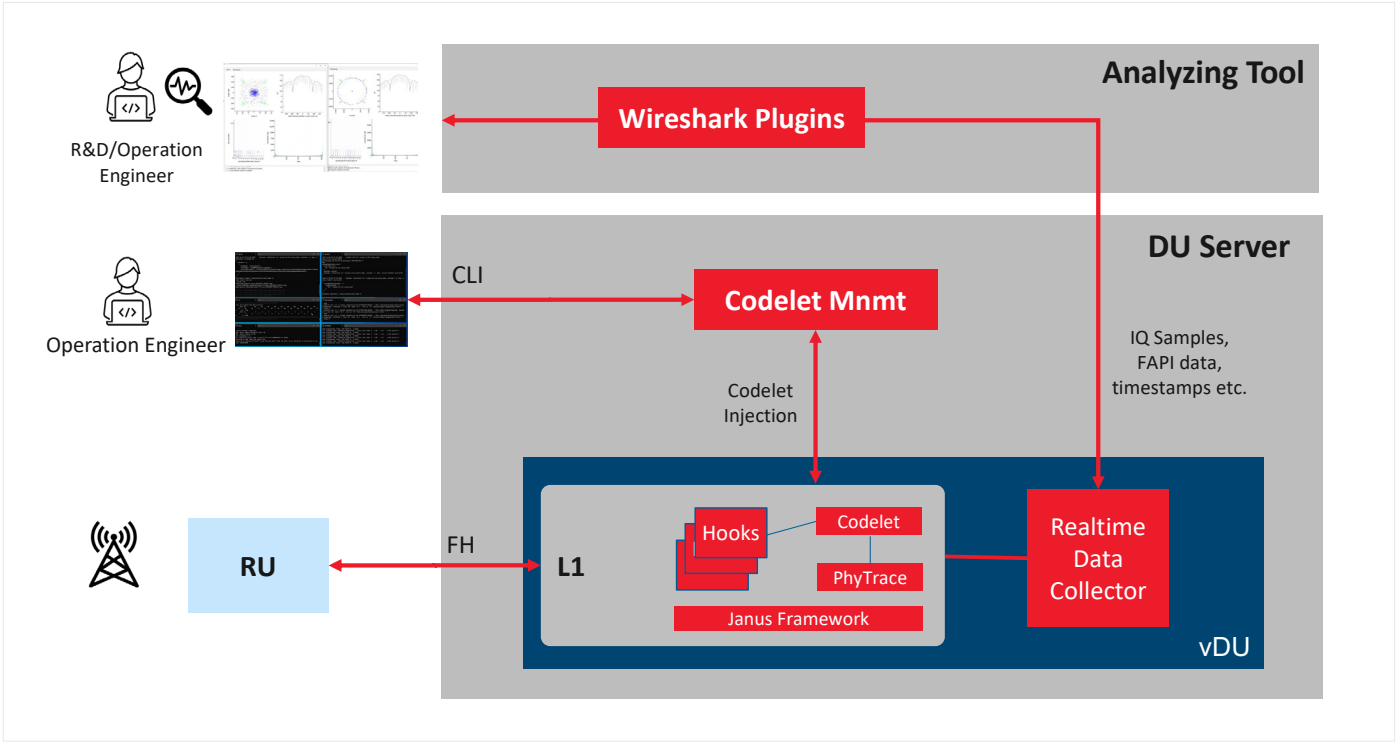

Figure 3. Janus-enabled on-site advanced debugging tool.

4th Gen Intel® Xeon® Scalable Processor with Built-in Accelerators

The Janus Framework software for Mavenir DUs was designed to use hardware accelerators for vRAN performance by offloading computationally heavy layer 1 tasks such as LDPC decoding or forward error coding (FEC). 4th Gen Intel® Xeon® Scalable processors have built-in accelerators to improve performance in AI, analytics, networking, storage, and HPC.

In Mavenir's Janus product development, the company used servers powered by 4th Gen Intel® Xeon® Scalable processors with Intel® vRAN Boost, which offloads FEC data flows to an on-chip accelerator for processing. The product family supports up to 52 high-performance cores and six on-chip accelerators.

By fully integrating acceleration directly into the CPU, Intel has eliminated the need for an external acceleration card. This unique design innovation—Intel® vRAN Boost—not only reduces system complexity, but also provides substantial power savings.

In fact, 4th Gen Intel® Xeon® Scalable processors with Intel® vRAN Boost deliver up to twice the capacity and an additional ~20% compute power savings versus their previous generation processor[3]. The processors also have a range of features for managing power to further optimize performance per watt.

FlexRAN Software Architecture - eBBU Pool Framework and Janus Codelets

The FlexRAN Reference Architecture includes a baseband unit (BBU) pooling framework that allows the application to distribute the DU tasks between multiple cores from one to N cores. The framework uses a distributed architecture where each core in the pool can produce tasks to, and consume tasks from, a shared set of priority queues. No task is pinned to run on a dedicated core. In Figure 4, the dark blue ovals represent tasks that get executed on different cores, each task execution time varies depending on the task type.

Due to the flexibility of FlexRAN's software implementation of RAN, and Janus' efficiency and the speed of execution, we are able to execute Janus codelets within various FlexRAN tasks, inlined. This allows real time dynamic access to L1 telemetry on a per tasks granularity.

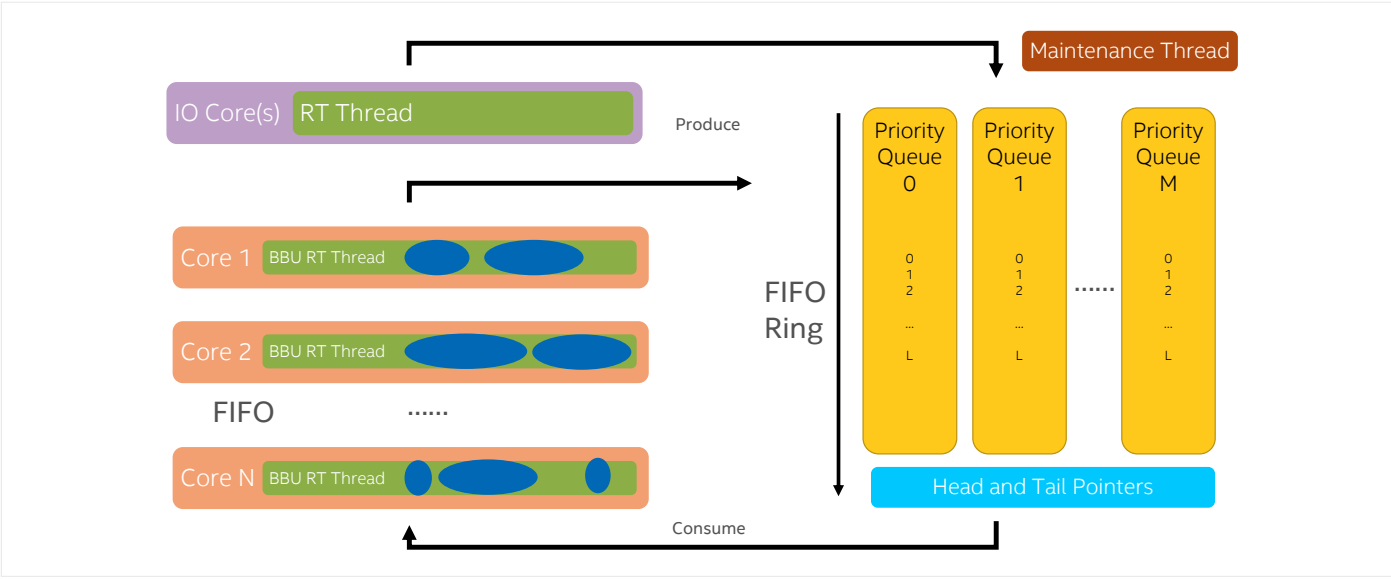

Figure 4. Example FlexRAN eBBU architecture.

Conclusion

Wireless service programmability is an important reason to deploy Open RAN networks. Already non-RT RICs and near-RT RICs can use applications to expand control of the network or to add functionality. But these two controllers can't operate in the time window of layer 1 which means an important part of the network is without observability or innovation.

One of the reasons why Microsoft has developed the Janus Framework is to meet the need for additional layer 1 programmability. Janus specifies eBPF codelets that can have real-time access to data from any part of the RAN stack. Mavenir has successfully customized Janus to improve DU debugging and instrumentation.

The company is in the process of productizing its research so that it can be made available to customers. Intel technology is a key part of Mavenir's work as its debugging capability is designed to work with Intel® FlexRAN to expand the standard debugging data available from the platform. Also, Janus is designed to utilize look aside accelerators for performance and the 4th Gen Xeon® Scalable processor has built in vRAN Boost that accelerates FEC processing.

While the collaboration between the three companies has led to some breakthroughs in extracting data for layer 1 observability, the companies are working on ways to use the Janus to exert control back on the layer 1 processing for even better network control and automation.

Learn More

Mavenir Open vRAN

Microsoft programmable RAN platform with dynamic service models

4th Gen Intel® Xeon® Scalable processor with vRAN Boost

Intel and Rakuten Power a Revolution in How Mobile Networks are Built

FlexRAN™ Reference Architecture

Intel® Network Builders

[1] Foukas, X. et al, "Taking 5G RAN Analytics and Control to a New Level", ACM Mobicom 2023.
[2] Depending on the speed of the CPU.
[3] For workloads and configurations visit www.Intel.com/PerformanceIndex. Results may vary.