Introduction

The FTS® for Bluetooth® protocol analyzer Virtual sniffing function simplifies Bluetooth development and is easy to use. It also can be used to add value to other Bluetooth development tools such as Bluetooth stack SDKs and Bluetooth chip development kits.

This white paper discusses:

- Why HCI sniffing and Virtual sniffing are useful.
- Bluetooth sniffing history.
- What Virtual sniffing is.
- Why Virtual sniffing is convenient and reliable.
- How Virtual sniffing works.
- Virtual sniffing and Bluetooth stack vendors.
- Case studies: Virtual sniffing and Bluetooth mobile phone makers.
- Virtual sniffing and you.
- Where to go for more information.

Why HCI Sniffing and Virtual Sniffing Are Useful

Because the Bluetooth protocol stack is very complex, a Bluetooth protocol analyzer is an important part of all Bluetooth development environments. The typical Bluetooth protocol analyzer "taps" a Bluetooth link by "sniffing air". For many Bluetooth developers sniffing the link between a Bluetooth Host CPU and a Bluetooth Host Controller—also known as HCI sniffing—is much more useful than air sniffing.

HCI sniffing direct provides visibility into the commands being sent to a Bluetooth chip and the responses to those commands. With air sniffing a software engineer working on the host side of a Bluetooth chip has to infer and often guess at what their software is doing. With HCI sniffing, the software engineer can see exactly what is going on. HCI sniffing often results in faster and easier debugging than air sniffing.

FTS for Bluetooth's Virtual sniffing feature is a simple and easy way to perform HCI sniffing. Virtual sniffing isn't limited to just HCI sniffing, but it is the most common use and this white paper will focus on the HCI sniffing application of Virtual sniffing.

It is also important to understand that FTS for Bluetooth is a multi-mode product. FTS for Bluetooth does support traditional air sniffing. It also supports serial HCI sniffing (for the H4 (HCI UART), H5 (3-wire UART) , and BCSP (BlueCore Serial Protocol) protocols), USB HCI (H2) sniffing, and Virtual sniffing. So with FTS for Bluetooth nothing is sacrificed—the product is simply more functional that other Bluetooth protocol analyzers.

Bluetooth Sniffing History

Frontline has a strong appreciation for the importance of HCI sniffing because of the way we got involved with Bluetooth. Also, because of our company history, we are uniquely qualified to offer a multi-mode analyzer that provides so many ways to sniff and supports such a wide variety of protocols. This brief Bluetooth sniffing history should help you understand our approach to Bluetooth protocol analysis.

In the early days of Bluetooth, there where no commercially available Bluetooth protocol analyzers, so developers built their own debug tools and/or used protocol analyzers that weren't built for Bluetooth. Many developers built homegrown HCI analyzers—basically hex dumps and crude traces—because they recognized the need for visibility into the HCI interface and because it was too difficult to build air sniffers. Several companies developed air sniffers because they saw a market need and because they realized that they could charge a high price (US$25,000 and higher).

Two Bluetooth chip companies, Silicon Wave and Broadcom were using Frontline's Serialtest® serial analyzer to capture serial HCI traffic and then they would manually decode the HCI byte stream. This manual decoding was far too much work and so, independently, Silicon Wave and Broadcom each requested that Frontline produce a serial HCI Bluetooth analyzer that would have all the features of Serialtest. In response to these requests Frontline developed SerialBlue®—the world's first commercially available serial HCI analyzer.

The response to SerialBlue was very positive. When we asked our Bluetooth customers what they wanted next we quickly learned that there was a need for an affordable air sniffer that provided the same quality as SerialBlue. We also learned that the ultimate Bluetooth analyzer would be one that sniff air and sniff HCI simultaneously.

As work was progressing on our combination air sniffer and HCI sniffer the functional requirements for Bluetooth analyzers were changing. It was no longer good enough just to decode the core Bluetooth protocols (LMP, HCI, L2CAP, RFCOMM, and OBEX). Applications were beginning to be built on top of Bluetooth and therefore application level protocol decoding was becoming a requirement. For example, people were starting to browse the Internet using Bluetooth-enabled phones and PDAs therefore a good Bluetooth analyzer would need to support TCP/IP, HTTP, etc.

For Frontline to support for these higher levels protocols was no problem since they were already in use in other Frontline analyzer products. People have been using Frontline Serialtest serial analyzers and Ethertest Ethernet analyzer to troubleshoot TCP/IP and Internet problems for many years.

As we continued to work closely with the Bluetooth community we also came across one other requirement: sniffing itself had to be made easier. We took a two-pronged approach to this problem. We simplified air sniffing (and we continue to work on simplifying the process of air sniffing) and we invented Virtual sniffing.

Virtual Sniffing—What is It?

Historically, protocol analyzers have physically tapped the circuit being sniffed. For example, an Ethernet circuit is tapped by plugging into the network. A serial connection is sniffed by passively bridging the serial link. A Bluetooth air sniffer taps the piconet by synchronizing its clock of the piconet Master.

Not only is there a physical tap in traditional sniffing, but the sniffer must have some knowledge of the physical characteristics of the link being sniffed. For example, a Bluetooth air sniffer must know the BD_ADDR of at least one piconet member to allow it perform clock synchronization. A serial sniffer must know the bit rate of the tapped circuit or be physically connected to the clock line of the circuit.

With Virtual sniffing the protocol analyzer itself does not actually tap the link and the protocol analyzer does not require any knowledge of the physical characteristics of the link.

In computer jargon, “virtual” means “not real”. Virtual memory is memory that doesn’t actually exist. Virtual reality is something that looks and feels real, but isn’t real. So we use the term Virtual sniffing, because there is sniffing taking place, but not in the traditional physical sense.

The Convenience and Reliability Of Virtual Sniffing

Virtual sniffing is the most convenient and reliable form of sniffing and should be used in preference to all other forms of sniffing whenever practical.

Virtual sniffing is convenient because it requires no setup to use except for a very small amount of software engineering (typically between one and four hours) that is done once and then never again. Once support for Virtual sniffing has been built into application or into a development environment none of the traditional sniffing setup work need be done.

This means:

- NO piconet synchronization.
- NO serial connection to tap.
- NO USB connection to tap.

Virtual sniffing is reliable because there is nothing that can fail. With Virtual sniffing all data is always captured.

How Virtual Sniffing Works

FTS for Bluetooth Virtual sniffing works using a feature called Live Import. Any application can feed data into FTS for Bluetooth using Live Import. A simple API provides four basic functions and a few other more advanced functions. The four basic Live Import functions are:

1) Open a connection to FTS for Bluetooth.
2) Close a connection to FTS for Bluetooth.
3) Send an entire packet to FTS for Bluetooth.
4) Send a single byte to FTS for Bluetooth.

All applications that send data to FTS for Bluetooth via Live Import use the first two functions. Usually only one of the two Send functions are used by a particular application.

When FTS for Bluetooth receives data via Live Import the data is treated just as if FTS for Bluetooth had captured the data on its. The entire protocol stack is fully decoded.

With Virtual sniffing the data can literally be coming from anywhere. FTS for Bluetooth doesn't care if the data being analyzed is being captured on the machine where FTS is running or if the data is being captured remotely and passed into FTS over an Internet connection.

Virtual Sniffing and Bluetooth Stack Vendors

As the complexity of the Bluetooth protocol stack increases Bluetooth stack vendors are realizing that their customers require the use of a powerful Bluetooth protocol analyzer. Even if the stack vendors stack is bug free, there are interoperability issues that must be dealt with.

The homegrown hex dumps and trace tools from the early days of Bluetooth just aren't good enough anymore. And building a good protocol analyzer is not easy. So stack vendors are partnering with Frontline. This permits the stack vendors to concentrate of improving their stack.

The typical Bluetooth stack vendor provides a Windows-based SDK. The stack vendor interfaces their SDK to FTS for Bluetooth by adding a very small amount of code to the SDK, somewhere in the transport area, right about in the same place that HCI data is sent to the Host Controller.

If FTS for Bluetooth is installed on the PC and the Virtual sniffer is running then the data will be captured and decoded by FTS for Bluetooth, in real-time. If FTS for Bluetooth is not installed or the Virtual sniffer isn't running then no harm is done. Virtual sniffing is totally passive and has no impact on the behavior of the SDK.

One Frontline stack vendor partner likes virtual sniffing so well that they asked us to create a new version of FTS for Bluetooth that only does Virtual sniffing. They now ship a Virtual sniffer with every SDK they sell. In addition to providing the Virtual sniffer, if their customer wants to take advantage of other FTS for Bluetooth features, such as sniffing, they will sell the customer a full version of the FTS for Bluetooth.

Another Frontline stack vendor partner feels so strongly about FTS for Bluetooth that not only have they built Virtual sniffing support in their SDK, but they have made FTS for Bluetooth an integral part of their product offering. They are actively encouraging all customers on a worldwide basis to adopt FTS for Bluetooth as their protocol analysis solution.

Case studies: Virtual Sniffing and Bluetooth Mobile Phone Makers

Case Study # 1

A Bluetooth mobile phone maker had been using a homemade HCI trace tool to debug the link between the Host CPU in the phone the Bluetooth chip. They also were using an air sniffer. They replaced their entire sniffing setup by moving to FTS for Bluetooth.

In the original test setup the Host CPU in the phone would send debug messages and HCI data over a serial link. A program running on a PC logged the output from the Host CPU.

To implement the new system using Virtual sniffing, a small change was made to the PC logging program and it now sends the data to FTS for Bluetooth using the Live Import API. The HCI traffic is fully decoded and the debug messages are decoded as well.

The decoder for the debug messages was written using FTS for Bluetooth's FrameDecoder feature. FrameDecoder allows FTS for Bluetooth user to write custom decodes and to modify decodes supplied with FTS for Bluetooth. FrameDecoder is supplied as a standard part of FTS for Bluetooth. In this case there are also HCI Vendor Extensions that the customer created a custom decoder for.

The air sniffer that was formerly used has been replaced by the standard FTS for Bluetooth air sniffer.

Case Study # 2

A second Bluetooth mobile phone maker plans to use Virtual sniffing in conjunction with a Linux-based custom test platform they have developed. Currently they capture serial HCI traffic on their Linux system and use a set of homegrown utilities to decode the captured data.

They plan to send the captured serial HCI traffic out of the Linux system using TCP/IP over Ethernet. Over on the PC running FTS for Bluetooth they will use a simple TCP/IP listening program to bring the data into the PC and this program will hand the data off to FTS for Bluetooth using the Live Import API.

Virtual Sniffing and You

If you are a Bluetooth stack vendor, Bluetooth chip maker, or a maker of any other products where integrating your product with FTS for Bluetooth's Virtual sniffing is of interest please contact Frontline to discuss your requirements. There are numerous approaches that we can use to structure a partnership program with you. We believe that a partnership with Frontline is an easy and cost-effective way for you to add value to your product offering.

If you are end customer and you want to take advantage of Virtual sniffing, all you need to do is buy FTS for Bluetooth. Virtually sniffing comes standard with product.

Where To Go For More Information

- Contact Eric Kaplan, Bluetooth Product Manager.
  ekaplan@fte.com
  +1 (434) 984-4500

- If you have a copy of FTS for Bluetooth installed then in the FTS for Bluetooth folder double-click on Virtual Sniffer (Data Source) and press the Show Live Import Information button.

- The Live Import Developer Kit contains detailed technical information, code samples, etc. A shortcut to the Live Import Developer Kit installer is available by going into the Optional Components folder of your FTS for Bluetooth installation.