IBM CORPORATION

Moderator: Susan Greenlee
March 24, 2004
1:30 p.m. CT

Operator: Good day, everyone. This is (Brett Bouger), and I'd like to welcome you to today's IBM teleconference entitled, "z/VM Version 4 Release 4 Networking Choices: Moving from Guest LAN to Virtual Switch."

Before we begin, let me quickly cover a few administrative items. First, if you have not received a copy of today's presentation of materials, you can have a copy of the PDF sent to you via e-mail. If you would like to receive a copy immediately, please press star zero at this time and an Operator will get your contact information. Also, today's program is being recorded.

And now, I would like to introduce your host for today's program, Susan Greenlee. Susan, please go ahead.

Susan Greenlee: Hello and welcome on behalf of IBM's Systems Group. Thank you for choosing to spend some of your valuable time with us today. This is the second in a series of quarterly Linux on zSeries technical educational teleconference prepared for zSeries customers. My name is Susan Greenlee. I'm an IBM Linux Technical Support Marketing Specialist for zSeries, and it is my pleasure to serve as your moderator for today's session.

Today, we will provide you with an overview of the networking choices available to you with z/VM Version 4.4 as they pertain to Linux on zSeries. We'll define networking components and when to use them like Guest LAN, V-Switch, V-LAN and we'll cover the important considerations you should understand when migrating from Guest LAN to V-Switch.

So, we (enter) session today we are pleased to have Richard Lewis, a member of the Washington System Center Advanced Technical Support Organization for 14 years. As a member of Advanced Technical Support, Richard works with technical sales specialist and IBM customers to assist with implementation, problem resolution and education.

Richard was a VM system programmer for six years before he joined the Washington System Center and for the past four years, Richard has worked with Linux on zSeries. He's a frequent presenter at (Share) and was part of the team that implemented the Linux Community Development System. That's an IBM mainframe on the Internet supporting independent open source developers of Linux software. Richard, thank you for joining us on this forecast.

Richard Lewis: Thank you very much, Susan. I'm very happy to be here.

Susan Greenlee: We'll get to Richard's remarks in just a moment, but first let me cover the flow of the call today. Rich will present for about 50 minutes then we will conduct a brief electronic survey and this will followed with an opportunity to ask questions about today's material. Finally, I will close with directions on how to obtain more information about Linux and a replay of this teleconference. So, with that, let us get started. Rich, can you help our audience understand what networking options are available today with z/VM and when should they be used?

Richard Lewis: I'd be happy to. Well, we're going to begin this presentation by looking at the agenda page. The purpose of this presentation is to assist customers who have perhaps already installed z/VM Version 4 Release 3 and perhaps have Linux Virtual Machines running on that system.

Most likely customers in that situation would have implemented connectivity by using the Guest LAN facility that is provided with z/VM. Now that z/VM Version 4 Release 4 is available, there are additional networking choices available to customers implementing Linux Virtual Machines and really any other Guest operating system as well.

So, what we'd like to do in this presentation is just take a brief moment to review what was available with Version 4 Release 3, look at what's new with Version 4 Release 4, (level-set) everybody in terms of the purpose and definition of the various system objects and components that make up that support and then consider why you might choose a virtual switch, for example, over a Guest LAN – what situations would apply and when actually using a Guest LAN might make more sense and be a better choice.

Since we're considering the movement of customers from Guest LAN to this new virtual switch function we'll also cover some considerations that you need to be aware of when you're doing that migration and then lastly we'll go through a few charts that show some performance information comparing Linux Virtual Machines running on a Guest LAN and Linux Virtual Machines using a virtual switch.

So, with that, please turn to the next chart which is entitled "z/VM 4.3 Guest LAN Support" and here is just a brief summary of what was available with z/VM 4.3 as it relates to Guest LANs. The Guest LAN function in z/VM was initially made available in Version 4 Release 2. And at that time, it only supported the HyperSockets architecture. With z/VM Version 4 Release 3, the HyperSockets support was enhanced to add multi-task support and then QDIO Guest LAN support was introduced.

A QDIO Guest LAN is meant to simulate an OSA-Express Gigabyte Ethernet card running in QDIO mode so with this new function there was a new type of virtual network interface card that would simulate that type of an interface and there was a new type of Guest LAN that could be defined that would provide the media for those virtual network interface cards. The support that was included with this matched what was available on a real OSA-Express card and that is it would allow operation in Unicast, Multicast or Broadcast mode but was limited to IP Version 4.

Now, also, in z/VM 4.3, there was a facility added to allow CP accounting records to be produced that would provide statistics concerning packets, bytes that were transmitted or received over Guest LAN interfaces and also would allow the distinction between an end-point generating traffic and a router Virtual Machine generating traffic as well. There was service that was produced during this time-frame and it's listed at the bottom of this chart.

The first PDF that is listed there was on the GA RSUs so if anyone saw z/VM 4.3 that piece of service was included if you install the RSU. The remaining three PDFs that are listed are on RSUs that have been produced since the general availability of z/VM 4.3 and those PDFs are certainly recommended. They are all on the most current RSU that's available for z/VM 4.3.

Now if you'd turn to the next chart – z/VM 4.4 Guest LAN support – let's look and see what has changed with the latest release of VM. We still have a Guest LAN support supporting both the HyperSockets architecture, as well as the OSA-Express QDIO.

Enhancements to the HyperSockets architecture include IP Version 4 broad-cast support so processes that are running within, for example, a Linux Virtual Machine that depend on being able to generate or receive broadcast packets are now supported on both a HyperSockets Guest

LAN, as well as a QDIO Guest LAN and then an important enhancement – the support for 802.1Q V-LAN tagging has been added both to the HyperSockets Guest LAN support as well as QDIO Guest LAN support. In addition, QDIO Guest LAN now supports IP Version 6 addressing so that is one enhancement to that part of the Guest LAN function.

Included in z/VM 4.4 is a new special purpose type of Guest LAN that has been named a virtual switch and in just a moment we'll define more fully what a virtual switch is. Its purpose is, again, to connect Virtual Machines to each other and specifically an intention there would be to connect Linux Virtual Machines to one another but more importantly it also connects those Virtual Machines to an external network in a transparent manner.

In addition to that, the virtual switch is a V-LAN aware object so it provides V-LAN support even for guest operating systems that are not V-LAN capable themselves. This provides tremendous capability if an organization has implemented the 802.1Q V-LANs throughout using an OSA-Express device, the V-LAN can be extended through the virtual switch to the Guest operating systems running on VM.

Enhancements were also made to the virtual network interface cards so they can be created and that is the ability to specify a locally administrated media access control address or MAC address. Now once again, there is service that has been produced that directly affects this function and this service has been produced since the general availability of z/VM Version 4 Release 4. The first PDF mentioned on the chart – (UM30893) – is on the GA RSU tape. All of the remaining PDFs that are listed here are only available as corrective service at this time. They will, no doubt, be included on another RSU that is produced for z/VM 4.4.

One of (the) PDFs that has been listed there – (UM31036) – applies just to customers who are running z/VM on a (z990). This is a PDF that enhances performance for real devices such as a real IQD or HyperSockets channel or a real OSA-Express card but since the virtual switch function uses real OSA-Express cards, the performance enhancement applies to that object as well and that's why I've listed it here.

The remaining PDFs that are listed deal with restart when there is a failure on a link, problems with the query LAN command, problems when a virtual network interface cards is detached when a stream of packets are still be sent and problems dealing with the broadcast support, so, again, that service that you would no doubt want to make sure that you install before you use a virtual switch or Guest LANs in a production environment.

Moving to the next chart, let's turn our attention now that we've briefly surveyed the support that's there to better defining some of the terms I've just used. First off, I'd like to turn your attention to the chart titled, "What is a z/VM Guest LAN?" I've mentioned this term but I haven't actually defined what it means. A Guest LAN is a system object that's created by z/VM and it represents a LAN segment. In other words, this is the media that would be used by packets that are traveling between hosts. I mentioned that it is a system object and it does not actually occupy or use any real hardware. It is a pure simulation that is provided by the control program portion of z/VM.

Now, since it represents the media that would interconnect Virtual Machines or host on a network, you can think of it as functioning very similar to a non-intelligent network hub. As a LAN segment, it supports what would generally be expected to be supported on a physical LAN segment and that is in terms of QDIO both IP Version 4 and Version 6 addressing and it supports packets that are generated as Unicast, Multicast or Broadcast. In addition, there can be a LAN segment that is defined as a HyperSockets type and that, again, supports IP Version 4 packets in all the three modes – Unicast, Multicast or Broadcast.

Now I want to emphasize the fact that no real hardware is involved here. This, in fact, is a system object and not a device of any type so there are no underlying hardware requirements to be able to define and use a z/VM Guest LAN. Now, since we're talking about hosts that are connected to a LAN segment, and this LAN segment resides simply on the z/VM system itself, there needs to be some way to interconnect that LAN segment with a physical LAN segment that would be a part of the network infrastructure of your business.

And that function is accomplished by having a router Virtual Machine. This would be either the z/VM, TCP/IP stacked or a Guest operating system such as Linux functioning as a router and it's purpose is to then join the LAN segment that is created on the VM system with the physical LAN segment. The router then would have a network interface on both the Guest LAN, as well as the physical LAN that it is attaching to. Another characteristic of Guest LAN since there is no real hardware involved there isn't a limit on the number of Guest LANs that you can actually create. The limitation would basically simply arise based on resources available to the z/VM system.

Now to put this in perhaps a little better perspective, if you would turn to the next chart, "Guest LAN Configuration Example" – here is a picture of what I've just been describing. You can see in this example that there are two Guest LANs defined and the z/VM TCP/IP stack is serving as the router. It is interconnecting the Guest LANs to the external LAN segment that is part of the 9.130.31 network and you'll notice also that each of the Guest LAN segments are in their own sub-net. Those Guest LAN segments are also in a different network from the physical network and this is what's required when you're using Guest LANs.

The Guest LAN segments, themselves, need to each be a sub-net and the Guest LANs typically need to be in a different network from the physical network they're attached to. Notice, also, that each Guest LAN supports only one architecture so the Guest LAN on the right where you see the Linux Guest with an interface type of ETH0 – that would be a QDIO Guest LAN. The Guest

LAN on the left is a HyperSockets Guest LAN and you'll notice that the Linux Virtual Machine's there have a network interface name of HSI0 reflecting, again, the different architecture's of the LANs that they're attached to.

If you would turn over to the next chart – "Defining a z/VM Guest LAN." These are simply some examples of how a Guest LAN can be defined since it's a system object. It can be defined either with commands that are dynamically executed or it can be defined by placing a definition statement in the system configuration file. Guest LANs can be created as either being unrestricted or restricted. An unrestricted Guest LAN allows any Virtual Machine that defines a virtual network interface card of the right type to connect to that Guest LAN.

In other words, it would be analogous to walking up to a hub and plugging in the RJ45 cable that comes out of your work station network card that's available to anybody. A restricted Guest LAN limits that connectivity to simply Virtual Machines that have been pre-authorized and it requires that either the owner of the Guest LAN authorize that connectivity or a privileged user on the z/VM system do that authorization as well. The CP privilege class that allows for the definition of Guest LANs that are owned by the system is Class-B and that would be the privilege Class required to authorize Virtual Machines to connect to a system owned Guest LAN.

A Guest LAN that's dynamically created by a general user is considered a transient Guest LAN in that it will not survive if the owner of the Guest LAN happens to log-off and all who are connected to that Guest LAN log off. A Guest LAN that is defined as owned by the system is considered a permanent Guest LAN in that it will survive even if all Virtual Machines connected to it log-off and break their connection. In both cases, the LAN itself is a system object.

The persistence or transient nature is simply a reflection of who created it and what user is associated as owner with that Guest LAN. Now you can see that in these examples – the first example I'm defining a LAN named (TEST-L) and in this case the owner ID is the system and it is a type QDIO Guest LAN so this would require a Class-B privilege to define. It also would require that Virtual Machines connecting to the Guest LAN have defined their virtual network interface card to be of the same type and that is QDIO.

The next example shows a general user defining a LAN. It's named (TEST-L) also but you see the owner ID is (TEST-VM). Again, this a type QDIO LAN but it is defined as restricted so before anyone can connect to that Guest LAN the owner or a privileged user would have to issue a set LAN CP command to grant Virtual Machines the authorization to connect to it and lastly there is a definition of a HyperSockets Guest LAN. This time it is also owned by the system but notice in this case there is no type specification and that's because the type of hypers is the default and that's implied if you don't make a type specification. In addition, a HyperSockets Guest LAN has the capability to define a maximum (frame) size and that option is not available for QDIO based Guest LANs.

So, we've talked about Guest LANs and I've mentioned that you need a virtual network interface card to attach to it. Let's turn to the next chart and define what a z/VM virtual network interface card actually is.

Again, it's a simulated device so there's no real hardware involved at all and it is a representation of either an IQD channel interface which would apply to HyperSockets or an OSA-Express QDIO channel interface. If you're familiar with those real devices, you realize that to make use of a port on those devices you need three sub-channels – one for control commands, another for read and another for write so always – a consecutive grouping of three devices are associated with any use of one of those physical devices. The same thing is true when a virtual network interface card is defined.

When you define a card, you specify the base, address of that sequence of three addresses. The implication is that three devices will be defined. It is certainly possible to define more than three devices and that would allow you to simulate the situation where particular Guest operating system is sharing the port on a real device. Many times this is done by zOS Guest that will have multiple TCP/IP stacks listening to the same port or could have multiple link in device definitions associated with a single port. By specifying more than three devices on the virtual network interface definition, it's possible to have the guest operating system believe it is in that same environment.

Now the Guest LAN I mentioned earlier is a system-wide object. The virtual network interface card is a device that is owned and localized to the Virtual Machine in which it's defined. The function of the virtual network interface card is the same as a real network interface card would be and it is to allow the attachment of that host to a LAN media and it allows them that host to process IP datagrams. A Guest virtual network interface card that's defined for a Guest can either be automatically attached to a target Guest LAN when the user logs on or it can be dynamically attached after the user logs on.

In the first case, the network interface card would need to be defined in the CP directory entry for the Virtual Machine so that that definition is in place when the Virtual Machine logs on. Part of that specification is to describe what system object be it a Guest LAN or a virtual switch the network interface card should be attached to at log on. If a device is defined after log-on then a CP command and the name of that command is (couple) needs to be used to associate the virtual network interface card with a LAN object.

There are some examples on the next chart where that chart titled "Defining a z/VM Virtual Network Interface Card" and the first set of examples are showing what you would do in a z/VM directory entry to define a virtual network interface card. The first example uses the special directory statement and the syntax of that definition is to specify the base address – in the case, "Able Dog Zero." The type of network interface card that's being specified – in this case, "QDIO." The number of sub-channel addresses to be associated with it and then this case three and then the owner and name of the system object it's being connect to.

In this case it is a LAN or a virtual switch. There's no distinction here that is owned by the system and named (TEST-L). The second statement is basically the same as the first but uses the new (definitioned) key word that is part of z/VM 4.4. The (Nick DEF) statement is the preferred method for defining a virtual network interface card (in) z/VM 4.4. It offers all of the same capability that existed on the special statement but extends the options beyond what you can do with a special statement. You can use the (Nick DEF) statement to actually define (immedia) access control ID, a locally administrated MAC ID to associate with this virtual network interface card and you can define the (chipped) number that should be associated with this network interface – something that is typically important in a zOS environment.

Again, the second example defines a QDIO network interface card (at) address, "(Able, Dog, Zero)" that consists of three sub-channels and will automatically be connected to the object owned by system named (TEST-L). The two examples below that show the use of a special statement and a (Nick DEF) statement to define a virtual network interface card that is of type HyperSockets or hypers. And again, you can see that that is the default. If I don't specify type QDIO what I end up with is a virtual network interface card that is appropriate for use on a HyperSocket Guest LAN. We'll see in just a moment that the virtual switch support does not support HyperSockets so if you're going to attach to a virtual switch you need to always define your (Nick) device as a QDIO type device.

The second set of examples shows the CP commands to dynamically define a virtual network interface card and attach that card to a system owned object such as a Guest LAN or a virtual switch. You can see that the definition is very similar to what is required in the directory. The difference here is it's a two-step process. If you define the device – the device is, in fact, defined but it's not usable. It's the same as having a workstation with a network card in it but no RJ45 cable plugged into it.

That cable isn't virtually plugged in until you issue the CP (couple) command and whether or not that command succeeds and whether or not the statements in the directory succeed depend on whether or not the target object is restricted or not. If it's unrestricted then it will always succeed. If it's a restricted object, it will only succeed if the Virtual Machine has been authorized to attach to that object.

Now we've covered Guest LAN and virtual network interface cards. Let's look at the last component that is associated with z/VM 4.4 and that is a virtual switch. If you would move to the next chart we'll define what a virtual switch actually is. A virtual switch is, again, a system owned object and it's a device that's going to connect Virtual Machines to each other and to a physical media. In the case of the Guest LAN that simply provided connectivity between Virtual Machines. There was no connection between that LAN segment and a physical media unless a router Virtual Machine with multiple interfaces was created and attached to the Guest LAN.

A virtual switch on the other hand can supply that connectivity between Virtual Machines but it can also supply connectivity to a physical LAN without the introduction of a router. This is the most powerful attribute or characteristic of a virtual switch and something that makes it very popular and very useful in a Linux Guest environment. You can think of a virtual switch as an object that functions similar to a layer 2 switch, and this would be a physical layer 2 switch. The major difference here between a physical switch and a virtual switch is the packets are being distributed or moved between hosts based on IP address at the IP layer as opposed to MAC address at the media layer but conceptually, it's doing the same thing.

It is simply moving packets from the physical network and delivering them to the virtual network interfaces of Guests that are attached to it. Now the benefit of this is you eliminate the need to actually have an IP routing function in between the Guests and the physical network and you then eliminate the network latency that might be associated with that. The virtual switch is the connectivity to the physical media and in this case it would be an OSA-Express card operating in QDIO mode.

The data transfer is handled by the CP portion of z/VM but the control function of the card is handled by a z/VM TCP/IP stack. That stack isn't involved in the data transfer as it would be if it were functioning as a router. It is involved in the card setup and card maintenance so it's an integral part of the virtual switch environment. It's just a different use of the z/VM TCP/IP stack.

Now another benefit of a virtual switch is that it can provide port based V-LAN tagging and filtering. Now I said earlier that in z/VM 4.4 802.1Q – V-LAN support was added to guest LANs as well as to virtual switches. The major difference here is that the virtual switch will provide port based V-LAN tagging where a Guest LAN will simply allow V-LAN tagged frames to flow on the media. The virtual switch unlike the Guest LAN can only be defined by a privileged user – a Class-B user – and it can not be owned by a general user (if it's) always owned by the system.

In addition, it is always a restricted object so the only way that a Guest Virtual Machine can attach to it is by previously being granted the authority to do so. Similar to a Guest LAN a virtual switch can be dynamically created by CP commands or it can be created when the system is IPLed by having the appropriate statements in the system configuration file. If you move over to the next chart – "Defining a z/VM virtual switch" – here I've listed some examples that show the various ways in which a virtual switch can be defined.

It's similar to a Guest LAN in that you use either a CP command or a command in the system configuration file and the construction of that command is very similar indeed to a Guest LAN definition. You define a V-Switch by giving it a name. Unlike a Guest LAN, you define a real device that the V-Switch will connect to. You also define a user ID that will function as the controller and this needs to be a user ID that's running the z/VM TCP/IP stack and then lastly, you might need to define a port name if there is a requirement for the real device that the V-Switch is going to be using for a port name to specified.

When you are creating either the command in the system configuration file or using the CP command, the port name portion of the command should be at the very end. Now it's possible to define a V-Switch without specifying a real device address. In this case, the V-Switch would simply allow for communication between Virtual Machines that are attached to it. It would not allow for the flow of packets from the real network to Virtual Machines until a real device is defined.

The second example there shows the CP command and it is basically identical to the command you would put in the system configuration file. Now, since I said it's a restricted object, you need to have authorizations granted and typically you would want to do that at IPL time so that Virtual Machines that are going to be auto-logged on the system and expect to be able to connect to the virtual switch can do so. So, it's possible to put in the system configuration file modify V-Switch commands that allow the authorization to connect to individual Virtual Machines.

The last example shows that you can dynamically authorize a Virtual Machine to connect but it is done with the set V-Switch command instead of the modify V-Switch command so notice the difference in syntax if you are putting the authorization in the system configuration file, you want to use modify. If you are dynamically executing the command, you want to use set V-Switch.

Now, a Virtual Machine that is not authorized to connect to a virtual switch is not going to be capable of clearing the virtual switch either so if you have Virtual Machines that want to monitor the status of a virtual switch and they're not Class-B Virtual Machines, then you would need to authorize those as well to be able to issue the CP commands that enable you to query the virtual switch.

In this presentation, I'm not going to dwell on the query commands. I've shown the define commands for both the Guest LAN, the virtual network interface card and the virtual switch. While there are definition commands, there are also query commands that can be executed to see the status of a virtual network interface card, a virtual switch or a Guest LAN.

Now when a virtual switch is defined – since it is a more intelligent system object in the Guest LAN there are various states associated with it and the next chart shows some of the more significant states. There's quite a list of states associated with a virtual switch. A virtual switch when it's queried and shows a status of defined is a virtual switch that's ready to use, except there are no real devices associated with it, so all traffic would flow between Guests. It would not flow outside the VM system or into it.

Similarly, the status of disconnected is displayed when real hardware is defined but is not operational. In other words, something is physically wrong with the port. The status that you want to see is ready and that indicates that there is a controller that has been associated with the virtual switch and is functioning. There are real devices that are associated with the virtual switch and they are functioning and the virtual switch is ready to transfer packets from the physical media to the Virtual Machines connected to it.

Status of controller not available would be the case where a virtual switch is defined and real addresses are identified but a TCP/IP Virtual Machine is not logged on that is capable of serving as a virtual switch controller. In order to be capable of doing that, the TCP/IP Virtual

Machine needs authorization to connect to the V-Switch system service – that's an IUCV authorization and it also needs to have a V-Switch controller statement included in the profile TCP/IP.

Now before we go on and look at a V-Switch example, let's define one more term that was used earlier and that is V-LAN. If you turn to the next chart, there is a very brief description of what a V-LAN is and this is a description of the 802.1Q V-LAN definition. It's basically the capability to take one physical LAN segment that has hosts attached to it and instead of having that LAN segment operate as a signal broadcast domain to logically divide it up into multiple broadcast domains where hosts on that single LAN segment communicate with a adjacent hosts that they need to communicate with but not with every host that is attached to that physical LAN segment.

The logical definition of the LAN segment is accomplished by associating a V-LAN ID – simply a number with either a port that a particular host is attached to or the network interface that that host has on the physical LAN segment and then only allowing packets to be delivered between hosts that have the same V-LAN ID. The V-LAN ID is carried in the IP packet and IP packets carrying V-LAN IDs are termed "tagged" or V-LAN tagged packets and the purpose of this is to optimize or reduce network traffic, and also to optimize the connectivity where hosts that need to communicate with one another can do so but do not need to be physically located adjacent or on even the same LAN segment as each other.

Now as you see at the bottom with z/VM 4.4, the TCP/IP stack for VM supports both HyperSockets and QDIO V-LAN and the Linux 2.4.17 kernel at that level has support for a V-LAN tagging. In other words, Linux at that level is V-LAN aware. There's also a URL if you want to learn more about a V-LAN.

Now one question that might come to mind is what would be the difference between Guest

LAN supporting V-LAN and a V-Switch supporting V-LAN then the primary difference is that a V-Switch can actually do V-LAN tagging which means that an operating system can participate in a V-LAN using a V-Switch even if that operating system is not V-LAN aware. In other words, it doesn't understand how to add V-LAN tags to datagrams and it doesn't understand what to do with them when a tagged datagram is received.

A Guest LAN simply allows V-LAN tagged packets to flow over the media. It does not perform any tagging operation itself or any explicit filtering. It's then up to the operating systems that are attached to the Guest LAN to be V-LAN aware in order for a V-LAN to be implemented on a Guest LAN.

Now if you'll switch to the next page, there's an example of configuring a V-Switch and here you can see that a V-Switch has been defined in the center of the diagram named (Test-SW) and there are a number of Linux Virtual Machines that are attached to that V-Switch. One important attribute to notice is that the IP addresses of the Linux Virtual Machines are in the same sub-net as the external LAN segment so unlike a Guest LAN, Virtual Machines that are attached to a V-Switch have IP addresses that are in the same network as the physical attachment for the VM system.

Now, also, in this case you can see that V-LAN IDs have been associated with the various ports that Guests are connected to. There are two V-LANs defined – the Virtual Machines (LYN001) and (LYN002) participate in V-LAN ID three and the other two participate in V-LAN five and so packets coming into (LYN001) and (2) will only be delivered if they have a V-LAN ID of (three). Packets coming into the system with a V-LAN ID of five will only be delivered to (LYN003) and (LYN004).

Now notice I also have two TCP/IP Virtual Machines defined. In this case, though, the

TCP/IP Virtual Machines are not serving as routers. They are simply serving as controllers to control the real OSA-Express devices that are there. So, that's why the line is off to the side and joined with the line that represents the physical OSA-Express device. Now I've shown two real devices associated with the V-Switch and two TCP/IP Virtual Machines and that's because there is (fail-over-support) in the V-Switch.

In other words, if one of the physical devices fails, for example, device 2104 – if that device goes in-operational the V-Switch will immediately try to use device (1Baker00). Similarly, if the TCP/IP Virtual Machine experiences an (ab-band) or something like that and no longer is functioning as a controller, the V-Switch will immediately employ (TCP/IP-2) as its V-Switch controller to continue and maintain operations.

Susan Greenlee: Rich, I now understand what a Guest LAN is and what a virtual switch is and what a virtual LAN is but can you tell me when would I use Guest LAN or virtual switch? When would I use one over the other?

Richard Lewis: Well, that's a very good question, Susan. If you turn to the next chart I have some guidelines as to when you might consider a Guest LAN as opposed to a virtual switch. A Guest LAN is necessary if you need the functions of a router. For example, to do packet filtering, firewall types of functioning or a network address translation.

In other words, if you want the Virtual Machines on the VM system to be a hidden network and you would like them to simply use one IP address that would be associated with a gateway Virtual Machine and be masqueraded out onto that network you would need to use a Guest LAN to do that since you would need the functions of a router. Also if want to use the HyperSockets architecture, you, again, need a Guest LAN or real HyperSockets device because a virtual switch only supports the OSA-Express QDIO architecture.

And then lastly if you're doing lots of testing where you'd need the capability for a general VM user to be able to quickly define a LAN object and allow other Virtual Machines to attach to it so that you can test the flow of packets between two Virtual Machines or some other scenario like that, then a Guest LAN is the appropriate device to use because it allows general users to define it.

On the other hand, a virtual switch is better suited for environments where performance is critical since there is no intervening router, network latency is reduced in addition the CPU requirements to actually move data into and out of the system are lower as well with a virtual switch. And also many times the difficulty of actually carving out a new sub-net for the IP addresses of Linux Virtual Machines – that represents a real impediment in many organizations to implementing Linux Guests on VM.

With a virtual switch, you simply need to identify IP addresses in your physical network that can be associated with the Linux host running on VM and they don't need to be in a separate sub-net, since, again, there is no gateway router. Lastly, a V-Switch is required if you want to implement V-LAN support particularly for operating systems that are not V-LAN aware since the virtual switch can perform the V-LAN tagging function and perform the V-LAN filtering function it is the appropriate device to use in those situations.

Now we've discussed a quite a bit about each of these objects. Let's turn our attention to migrating from a Guest LAN environment to a virtual switch and so if you'll turn to the next chart we'll begin that discussion. Since a virtual switch is a restricted system object, if you've been using unrestricted Guest LANs you'll need to come up with some scheme to be able to automatically authorize Virtual Machines to use it. Now since an authorization only lasts for the current IPL, there needs to be some way to perform that authorization over and over again.

It certainly wouldn't be difficult to IPL your system and enter a set V-Switch command for one Virtual Machine but if you're talking about 20 or 100 Virtual Machines each time you IPL the VM system you don't want operator intervention to perform those authorizations before you can actually bring those Virtual Machines up. So, you need a scheme to be able to do that and we'll look at a couple of options that you can consider for doing that.

Second, if you were using a Guest LAN, you probably had your Linux Virtual Machines in a different sub-net. When you move to a virtual switch you'll want to assign IP addresses for the Linux Virtual Machines that are part of the physical network so this might involve a change to outboard routers to no longer route a sub-net through a gateway that's defined on the VM system and it would involve changing the Linux Virtual Machines themselves to give them new IP addresses that match the physical network.

Now if the Guest LAN that you're coming from is a HyperSockets Guest LAN then that's going to involve even more changes because you'll need to change the network interface card that is defined for each of the Virtual Machines to be a type QDIO card and you'll also need to make changes within the Linux Virtual Machine to change the network name from HSI to ETH. This would be done in the modules.com file where you would associate an alias of ETH0, for example, with the QETH device driver instead of an alias of HSI0 and then you would also have to change the (IF) config files that bring up that network interface to have a name of ETH0 or ETH1 as opposed to HSI.

Now also there are changes that would be required for the VM TCP/IP stack and we'll get to those in just a moment. If you'll turn to the next chart there are some information about one scheme that you might employ to do the authorization for the virtual switch when you IPL and that is to include those statements – the authorization statements in the system config file.

Now one nice thing about the system config file is you have the capability to imbed the content of another file that resides on the same (CT Parm) disk as the system config file and this allows you then to define the V-Switch in the system config file and point to another file that actually contains the modified V-Switch commands. That way, when you create another Linux Virtual Machine rather than having to modify the system config file itself, you simply add a new modify statement to the imbedded file.

Now this way you can keep a large number of authorizations on the (Parm) disk without cluttering up the system configuration file. An alternative to having those authorizations in the system config file is to use the profile exec of a Virtual Machine such as (auto log-1).

If you turn to the next chart, you'll see there a sample (REX-exec) that could be called by the profile exec from (auto log-1) and this (exec) simply has a variable named "off user IDs" that consist of (blank delimited words) and those words are the user IDs of the Virtual Machines you're going to authorize and then you'll notice there is a (do-loop) there that executes the set V-Switch command for each of those user IDs that are listed.

So, here, again, to automate the authorization, you would need to maintain this (VS-Off exec) on (auto log-1's) – 191 mini disk adding new Virtual Machines to the off user IDs variable as you create them. Notice again also the difference in command when the authorization was in the system config file we used a modified command. When it's in the (auto log-1) profile or external to the config file it's a set V-Switch command. Now lastly, let's consider the migration from the standpoint of the TCP/IP Virtual Machine.

If you'll move to the next chart you can see that the role of the TCP/IP Virtual Machine changes when you go from a Guest LAN to a V-Switch. Many times the TCP/IP Virtual Machine and a Guest LAN environment is a router as well as the vehicle by which you log on to VM user IDs. When you move to a V-Switch environment the VM TCP/IP stack is still required to support log-ons to Virtual Machines but it no longer serves a routing function.

Instead, it will serve a controller function so you need to decide whether you want to have your VM TCP/IP stack directly attached to a physical device so that it can provide support for log-ons or you could elect to take the TCP/IP Virtual Machine and simply give it a virtual network interface card and have it attached to the same V-Switch that all of the Linux Guests attach to becoming just another host IP address on that V-Switch and supporting then the log-on capability to your VM system.

Now if you also could elect to have – and it would be a good idea to do so to have more than TCP/IP Virtual Machine available and those TCP/IP Virtual Machines would then be set up and enabled to function as V-Switch controllers. The definitions required for a V-Switch controller are very simple. (There's) simply one command that goes into the profile TCP/IP and that is a V-Switch controller statement. The sample profile TCP/IP distributed with VM has a commented out statement that you would simply need to uncomment. There's also the requirement that the directory entry for TCP/IP have the IUCV star V-Switch directive and that is also already included in the TCP/IP definition that comes with z/VM 4.4.

So, if you turn to the next chart you can see pictorial representation of how you might utilize TCP/IP in the two different ways. The diagram on the left shows the VM TCP/IP stack having its own OSA on (one-Baker-00) – it's own IP address and it is then the vehicle by which you log on to the VM system. The dotted line shows that that same Virtual Machine is also providing V-Switch controller functions to OSA 2104 and then the Linux Virtual Machines are attached to the external network using the V-Switch itself not going through VM TCP/IP.

The example on the right shows the scenario where you might choose to not have a second real device. In this case, OSA 2104 is being used by the V-Switch and that is the connection to the physical network. The VM TCP/IP stack simply has a virtual network interface device that plugs into that V-Switch and it's own IP address and, again, provides log-on support. The dotted line shows that this TCP/IP stack is, again, functioning as the controller for that virtual switch as well.

So, to summarize where we are so far in terms of creating and migrating to a V-Switch, the steps that you'd need to consider would be to uncomment that V-Switch controller statement in your profile TCP/IP so that you prepare a TCP/IP Virtual Machine to be a controller. Then you need to define the virtual switch that you're going to use. You need to associate an OSA-Express real device address triplet with that V-Switch. You have a controller to include in that definition, as well.

You, then, need to potentially change the virtual network interface definitions for the Linux Virtual Machines if you're coming from a HyperSocket. If you're not coming from a HyperSocket Guest LAN then there is no change required there. You might simply need to change the IP address of the Linux Virtual Machines that will connect to the V-Switch. In prior to connecting those Linux Virtual Machines, you'll need to authorize them to connect to the V-Switch.

Well, lastly let's take just a few minutes to consider the performance of a V-Switch as opposed to a Guest LAN. The remaining charts that you have here before the summary show the performance runs that were done and documented in the z/VM performance report that is available on the Web. These were comparisons between the Linux Virtual Machines communicating with one another on separate physical systems and the comparison was to look at a Linux Virtual Machine going through a Linux router and then over to another real system or a Linux Virtual Machine going through a VM TCP/IP stack as a router to another physical system and then, lastly, a Linux Virtual Machine going through a V-Switch to the other physical system and compare both the throughput and resource utilization required in a number of different scenarios.

Now those scenarios involved request response streaming and connect request response. The streaming workload involved a client sending 20 bytes to a server and the server responding with 20 megabytes and that workload was run for an elapsed time of roughly 400 seconds and then several trials of that workload were conducted as well. The request response scenario involves a client sending 200 bytes to a server. The server responding with 1000 bytes and, again, that scenario going on for at least 200 seconds.

The connect request response scenario is to measure the TCP/IP three-way hand shake where a client connects to the server – send 64 bytes – receives an 8K response from the server and then disconnects from the server then reconnecting again so those are the scenarios that were measured and if you look at the charts that are shown, you'll see that in the case of the Linux to V-Switch, it typically has lower CPU utilization so milliseconds per megabyte is lower than either the Linux to Linux router or Linux to VM router case and the throughput is typically either the same or higher.

There were a couple of cases where the throughput of the V-Switch was fairly close to the throughput of the Linux to VM router but in most cases the throughput even is higher than both of the routed cases so the charts streaming – there is two charts for two MTUs – Maximum Transmission Unit sizes. There are also charts that compare the connect request response and the request response cases and you can see that in each of those cases the CPU utilization is lower with a V-Switch and the throughput either in megabytes per second or transactions per second is higher with the V-Switch.

So, again, that is a reaffirmation of the statement made earlier that use of a V-Switch is appropriate where performance is a critical factor. So, at this point, let's summarize. Hopefully, you have a better idea of what a z/VM switch actually is and when it might be appropriate to use. Hopefully, you can see that z/VM has extended the networking choices available to you by providing virtual switch by adding broadcast support to HyperSockets, Guest LAN, by supplying the capability to extend 802.1Q V-LANs the VM environment and also you have seen that the V-Switch itself allows you to connect Linux Virtual Machines to a physical network in a much more simplified manner as a Guest LAN did. There's no need for virtual sub-nets and it is a very high performance object.

At this point, then this concludes the presentation and I'll return to Susan.

Susan Greenlee: Thank you, Rich. It should be clear to our audience that with the capabilities of z/VM Version 4.4 there are a number of options available to set up a virtual network that can be used for Linux Guest deployment. That concludes our formal presentation for today. Before we get to the Q&A, I'd like to tell you how to obtain more information about today's topic. The last page of the presentation material is the information (forces) sheet.

You may want your peers to get this update so a Real Audio, downloadable replay of this teleconference with the charts will be available within the next 24 hours until June 30, 2004 and the link is provided here. An audio tape can also be ordered. You can look for information on our next session by subscribing to the "Linux Line Newsletter." In just a moment we'll begin our Q&A. First, though, we'd like to ask you a few questions and I'll turn it back over to Premier so we can tell the audience how to give us feedback about today's session.

Operator: Thank you. As mentioned, we would now like to conduct a brief electronic survey. Your feedback is very important and we appreciate the time you take to answer these questions. Having said that, let me tell you how you can give your feedback. Please listen to all options presented and then make your selection by pressing the star key, followed by the appropriate number on your telephone keypad.

For our first question, please tell us how you would rate the overall value of today's session. If you found this session to be highly valuable, please press star one. If you found this session to valuable press star two or if you found this session to be of little value press star three.

If you're interested in learning about how Linux on zSeries can help simplify the development of new applications, lower your total cost of ownership and enable e-business in today's – on demand world, register for the March 30th teleconference, customer experiences with Linux and zSeries. The featured speakers include two key IBM clients: Harry Roberts, CIO of Boscov's Department Stores; and (Jarod Shockley), Assistant Director of Technical Services, Boston University. Please call Premier Conferencing at 800-289-0583 and ask to register for session number 435430.

And moving on to our next question, would you like to be contacted by IBM for answers to additional questions you have on today's topic. If you would like to be contacted, please press star one or if you would not like to be contacted press star two.

Looking for open solutions for a demanding environment? Join IBM at the second annual 2004 Linux on Wall Street Show and Conference April 28th, 2004, at the Roosevelt Hotel in New York City. IBM has a comprehensive set of capabilities featuring Linux and (grid) computing to create cost effective and better utilize systems to help make your operating environment an important business differentiator.

Join us after the conference for a cocktail reception at IBM's Linux Center of Competence at 590 Madison Avenue and discover how IBM and selected partner products can help you realize the full price and performance benefits of moving to a Linux and (grid) infrastructure. For more information, visit www.linuxonwallstreet.com.

And for our final question today, which one of the following technical topics would you like to see presented as a one hour teleconference next quarter? If you would like to hear about a technical topic on z/VM and Linux on zSeries installation, please press star one. If you would like to hear about a technical topic on the best security practices for z/VM and Linux on zSeries press star two. If you would like to hear about a technical topic on Oracle for Linux on zSeries press star three. If you would like to hear about a technical topic on Lotus Domino v6.5 for Linux on zSeries press star, four. Or if you would like to hear about a technical topic (ensemble), press star, five.

Register online to access the report on a total cost of ownership study on Linux server deployments performed by (Robert Frances) Group. Go to www.ibm.com/software and search for total cost of ownership study.

And that concludes our survey. Thank you all for your input.

Susan Greenlee: (With) your feedback, we really appreciate it. And now let's get to our Q&A panel session.

(Brett), can you please explain to the audience how they can submit their questions to Richard?

Operator: Thank you, Susan. At this time, we would like to move on to taking your questions. Our question and answer session will also be conducted electronically today. If you would like to ask a question, we'll need you to queue up to do that and to do that simply press the star key followed by the digit one on your telephone at any time. We'll take as many questions as time permits and take them in the order that you signal us. If you're using a speakerphone, please make sure your mute function is turned off to allow your signal to reach our equipment. We'd also like to remind you that this conference is being recorded for purposes of redistribution via audio tapes and Internet.

The question-and-answer portion of the call will be included in this recording. Therefore, any questions you ask will be included in the recording. To ensure your privacy, we not be announcing your company name when your line is open for your questions. If you have questions you would like to ask but do not want your question recorded, please submit your questions to Richard Lewis at rflewis@us.ibm.com or to Susan Greenlee at sgreenle – that's s-g-r-e-e-n-l-e – @us.ibm.com.

Once again to pose your question, press star one. And while we wait for the ((inaudible)), question, Susan, I'll turn things back to you.

Susan Greenlee: Thanks, (Brett). While we're waiting for any questions from the audience, Richard, maybe you can answer one of the questions that I had. Do I need to use z/VM TCP/IP as a gateway for Linux Virtual Machines with a virtual switch?

Richard Lewis: Well, that's a very good question, Susan, and one that many times is asked. The answer to that question is no. The Virtual Machines that are connected to the V-Switch have a direct presence on the LAN segment that the physical OSA associated with the V-Switch is connected to so the Linux Guest would specify an external router on that network segment as their next (hop) router. There wouldn't be a router on the VM system itself. In this case, the TCP/IP Virtual Machine is working as a controller managing the real device and not functioning as a router that gateway function is outboard from the VM system.

Susan Greenlee: Thanks, Rich, and now I'll turn it back to you, (Brett) to see if there are any questions from the audience.

Operator: And Susan, at this point, no one has signaled. Just another reminder to our phone audience; press star one if you'd like to ask a question or make a comment today.

Susan Greenlee: Richard, I'll just ask you one other question – kind of a follow-up – so what is the customization that I would need to do for the TCP/IP stack to use the V-Switch?

Richard Lewis: OK. Again, a good question, Susan. What you'd need to do is to make sure that the directory definition of the TCP/IP Virtual Machine has the statement IUCV (Star) V-Switch specified. That allows the TCP/IP Virtual Machine to connect to the V-Switch system service and then you also need to make sure that in the profile TCP/IP the V-Switch controller statement is uncommented.

That statement optionally allows you to specify a range of virtual device addresses that'll be used by CP when attaching the control devices. Specifying that range, though, isn't required. What has to be there is the V-Switch controller statement.

Susan Greenlee: Well, thank you, Richard. (Brett), do we have any questions at this time?

Operator: Not at this point, but once again, as a final reminder press star one to ask a question or make a comment today. And we'll take a question from Gary Ernst. Mr. Ernst, please go ahead. Your line's open.

Gary Ernst: Thank you. I understood you to say that in – when you defined a V-Switch and you connect the Guests to the V-Switch that (there) – they have a presence on the physical media the OSA is attached to but the implication is you're not going to route to perhaps other Guest LANs behind it and I'm looking at the appendix here and ((inaudible)) V-Switch they have perimeter here non-router which apparently is the default and – but you can also specify (pri-router).

Richard Lewis: That's a very good question, Gary. What the key words you're referring to on the defined V-Switch statement specify is whether or not when the V-Switch is defined and is preparing to use the real OSA port whether the V-Switch itself should specify the primary router attribute or not. You would want your V-Switch to specify primary router if one of the Virtual Machines connected to the V-Switch is, in fact, acting as a router or other Virtual Machines behind it that would most likely be attached to a Guest LAN.

In that case, the Virtual Machine that's acting as a router needs to see all the packets that are coming in not just the packets destined for it's own IP address and therefore the V-Switch itself must declare that it is a primary router on the real OSA device.

Gary Ernst: OK. Thank you.

Richard Lewis: You're welcome.

Operator: And at this point we have no further questions.

Susan Greenlee: Well, then, I say we should probably draw a session to a close. If you think of any questions later, please give Richard – send him an e-mail. It's rflewis@us.ibm.com. His e-mail is also on the front of the presentation, or you can send an e-mail to myself – Susan Greenlee – it's sgreenle@us.ibm.com.

And on behalf of Richard and myself, thank you for attending our session, and I look forward to joining you again next quarter.

END