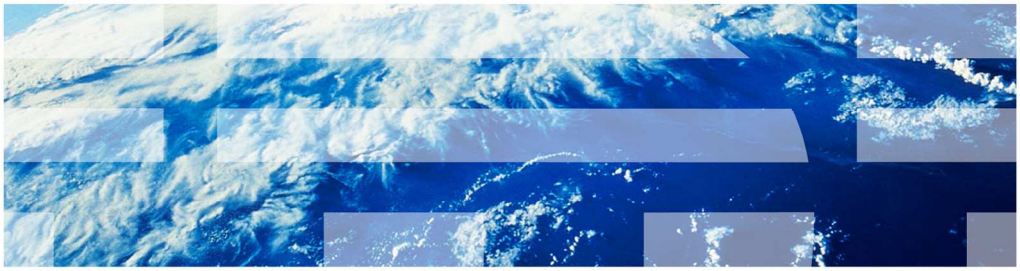
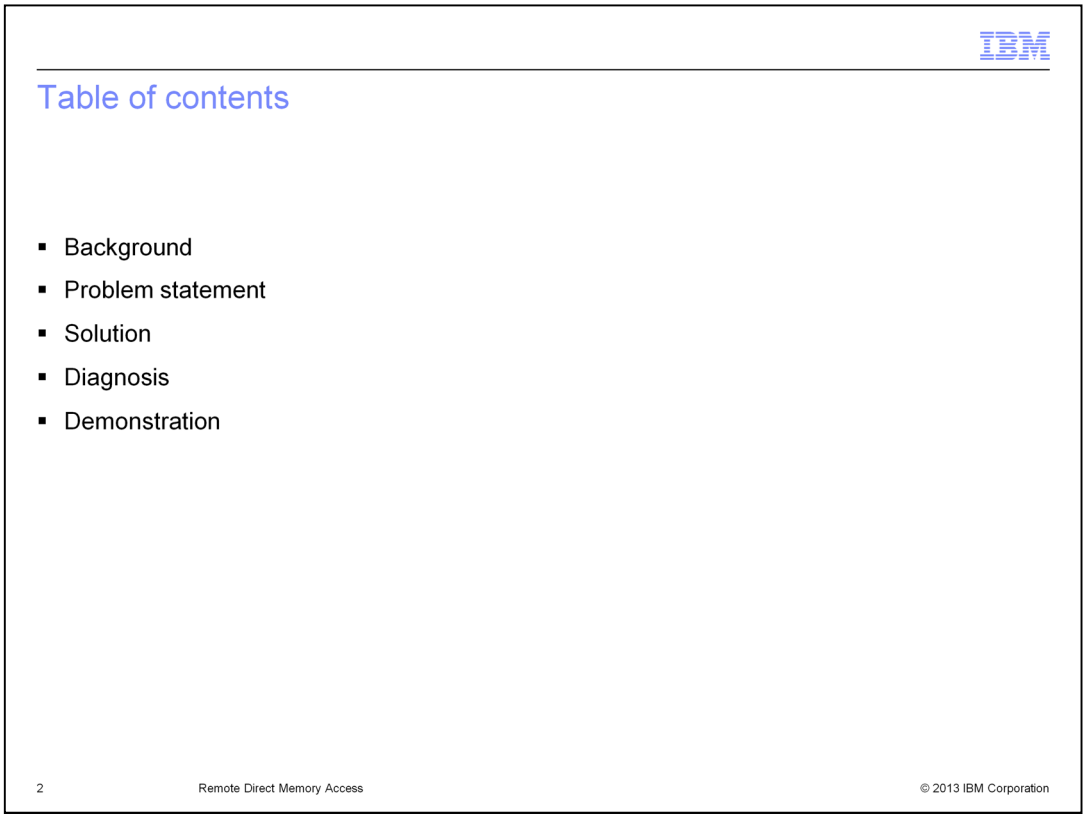
z/OS V2R1 Communications Server

Shared memory communications over RDMA



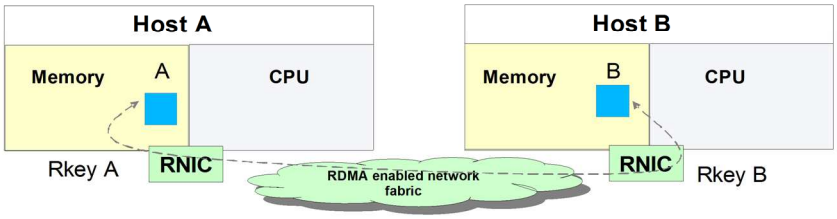
This presentation provides information about the Shared Memory Communications - Remote (SMC-R) function.



In this presentation, some background information is provided on the Configuration Assistant. Then the problem and the solution, addressed by the new function, are described in detail.

Background: Remote direct memory access

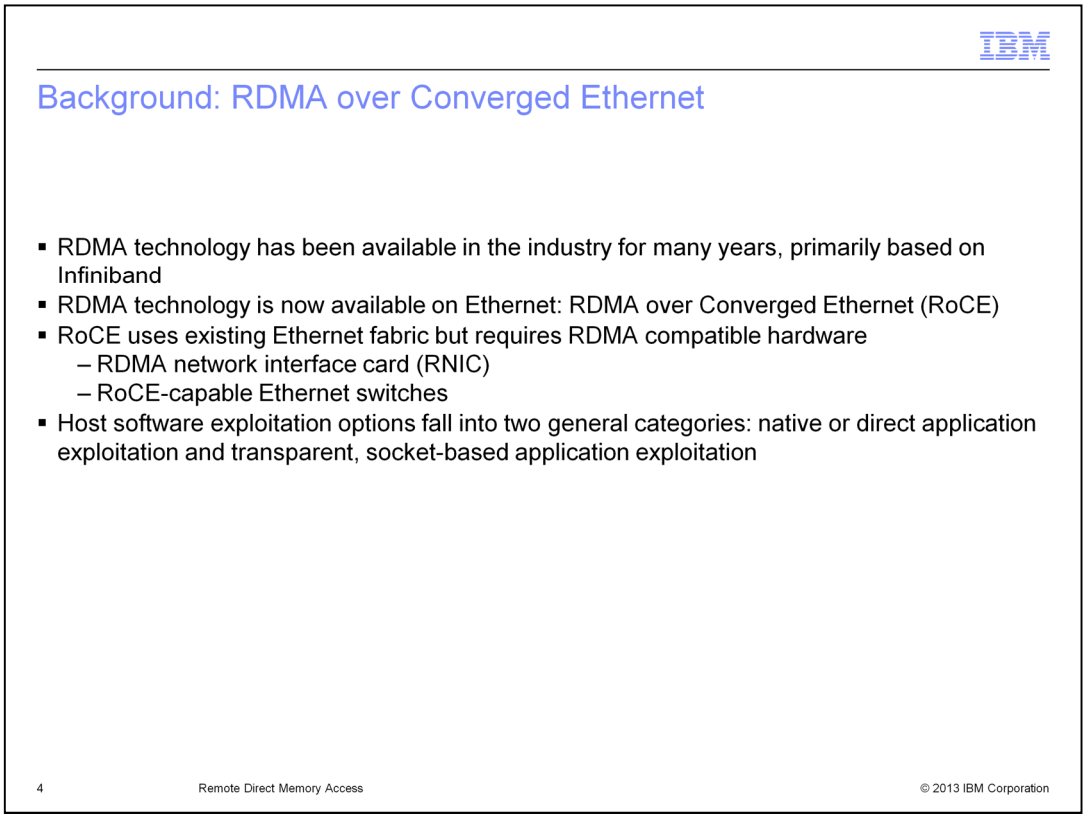
- Enables a host to read directly from or write directly to a remote peer's memory
  - Peer's processor or operating system not involved in transfer
  - Host registers specific memory for RDMA partner's use
  - Interrupts are still required for notification
- RDMA reduces networking stack overhead by using streamlined, low-level interfaces



3 Remote Direct Memory Access

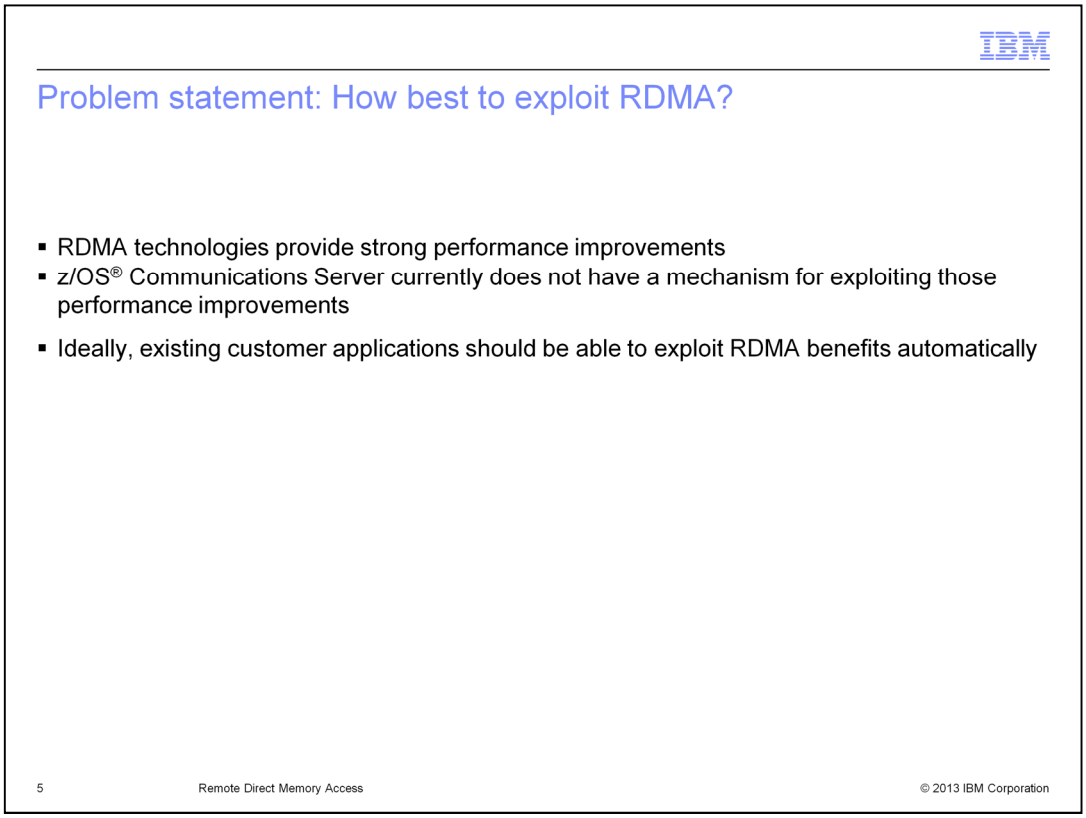
Remote Direct Memory Access, or RDMA, allows a host to access memory at a remote peer that is connected to the same RDMA-capable Ethernet fabric. RDMA protocols allow a host to read from or write into memory that the peer has allocated specifically for the use of this host.

Although interrupts are still required to alert the host that data has been received, significant performance improvements can be achieved using RDMA. Some performance gains are achieved because the processor and operating system at the receiving host does not participate in the RDMA transfer. Other gains are achieved because the TCP stacks can use simpler processing to send data using RDMA.



RDMA processing, based on Infiband technology, has been available for some time in the industry. Computing standards such as RDMA over Converged Ethernet (RoCE) have extended the RDMA model to Ethernet networks.

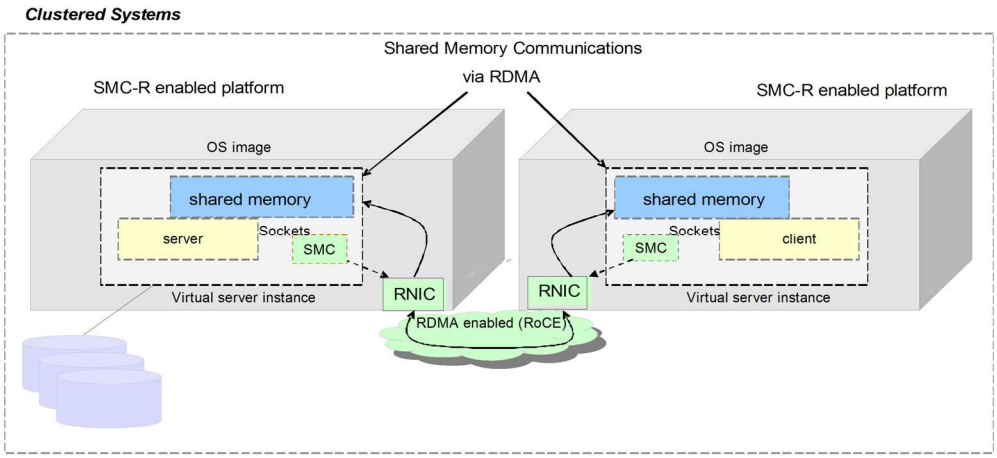
Specialized hardware is required to use the RoCE standards. Specifically, a new adapter known as an RDMA network interface card (RNIC) must be installed, and Ethernet switches capable of handling RoCE protocols must also be used. Applications can exploit RoCE processing directly, or the TCP/IP stack can shield the application from the RoCE protocols by exploiting the function at the socket layer.



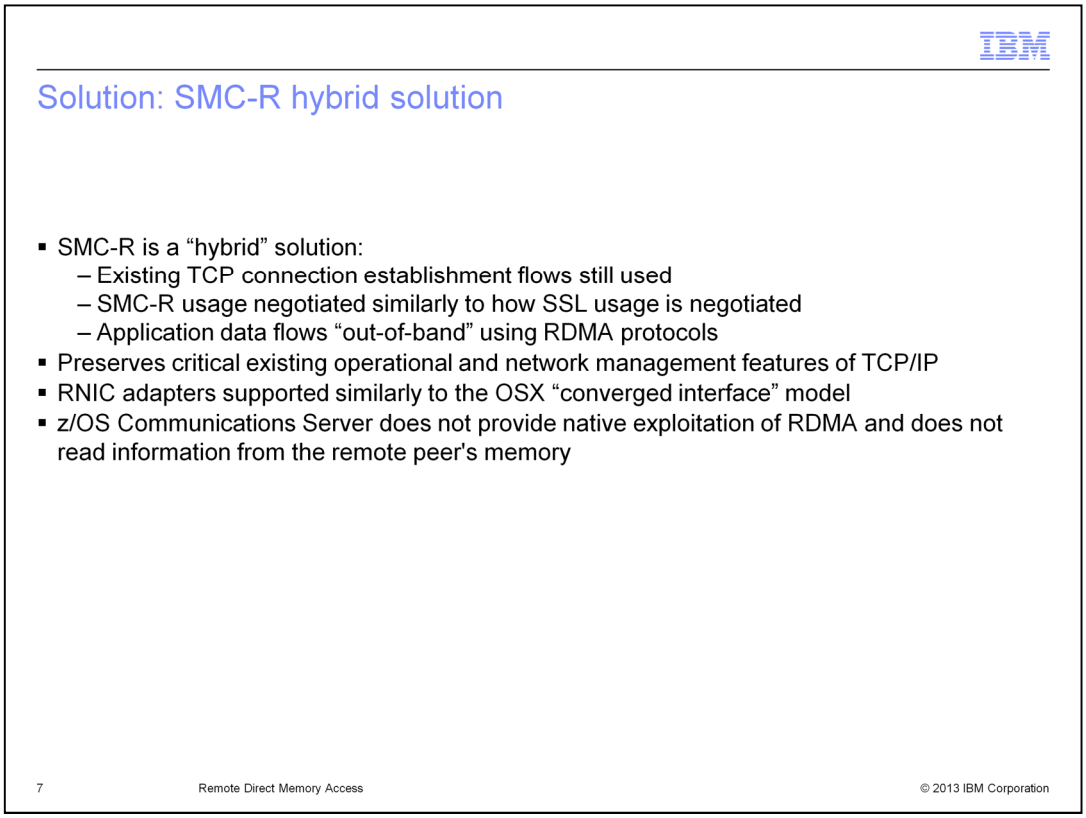
Before V2R1, z/OS Communications Server was unable to exploit RDMA because the necessary RoCE hardware was not available on the platform.

Solution: Shared memory communications over RDMA

- Shared Memory Communications over RDMA (SMC-R) defines a means to exploit RDMA technology for communications transparently to the applications



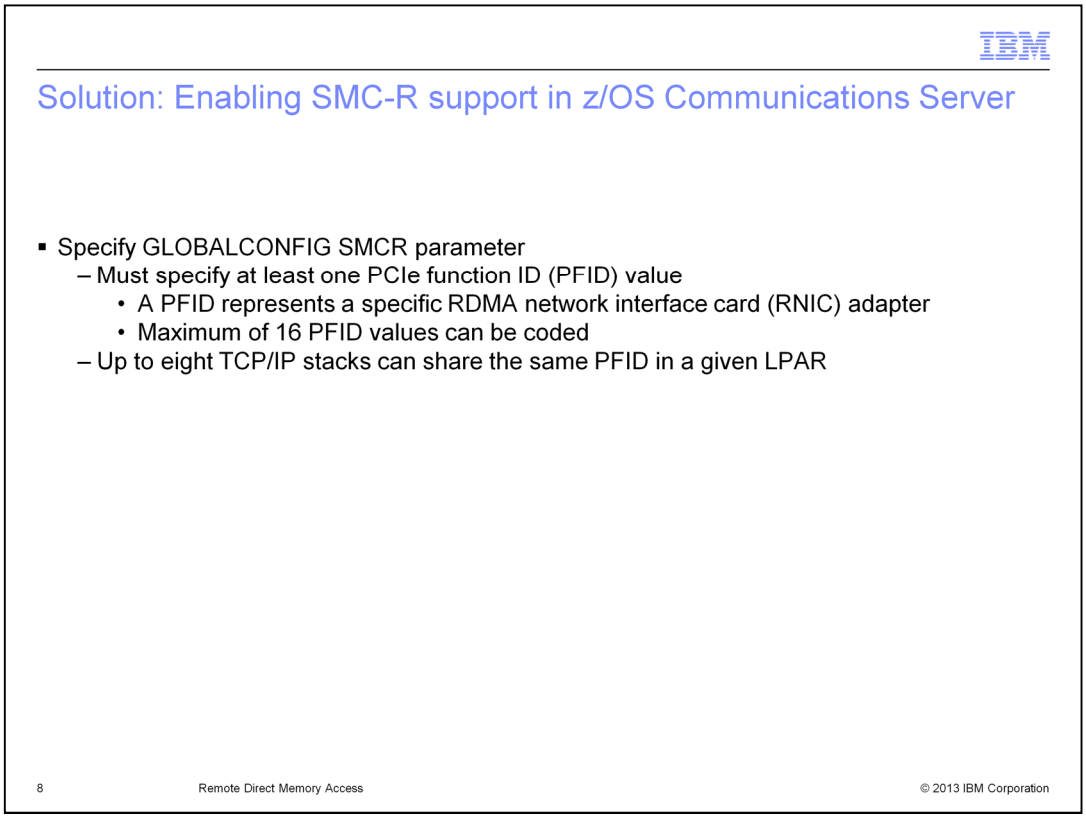
z/OS Communications Server will exploit RDMA technology using Shared Memory Communications over Remote Direct Memory Access (SMC-R) protocols. The SMC-R protocols provide a transparent socket-based exploitation model for RDMA, allowing existing applications to benefit without change.



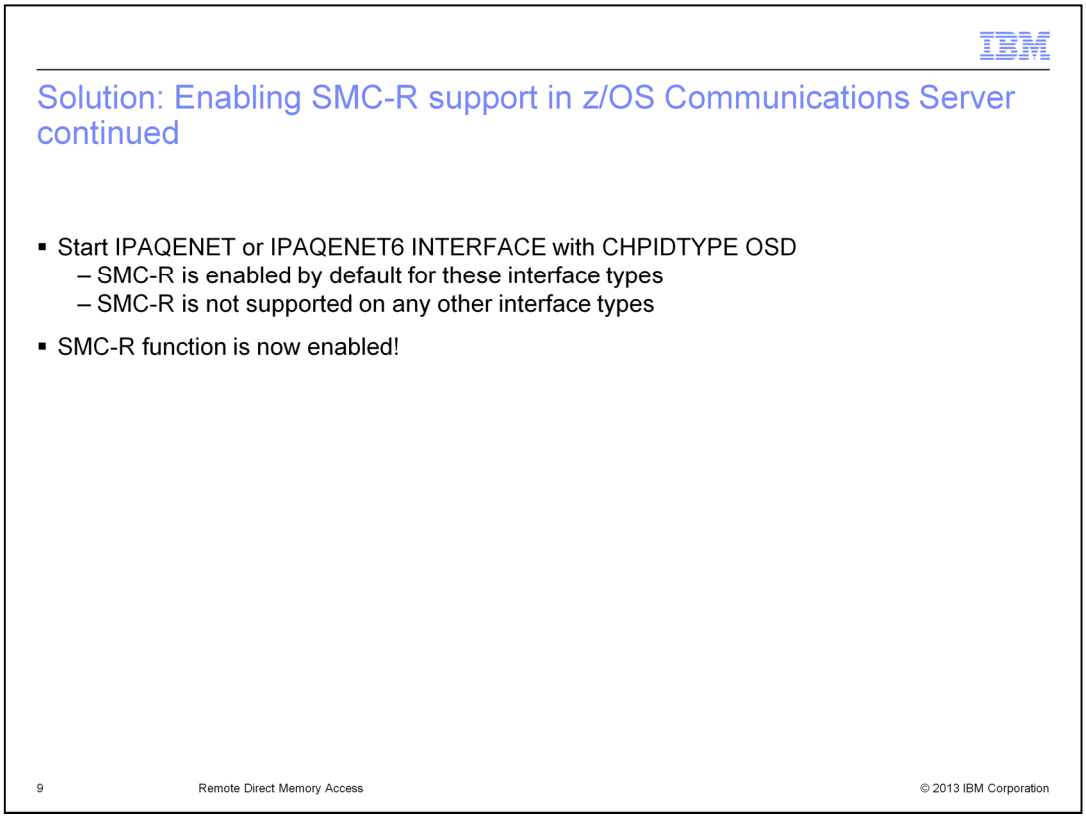
One of the primary goals for SMC-R processing is to minimize changes to existing TCP/IP stack operational and network management capabilities. SMC-R achieves this by using existing TCP connection establishment flows to determine SMC-R eligibility, in a manner similar to existing SSL negotiations. Once eligibility to use SMC-R is established, the application data is routed to the peer using RDMA protocols.

SMC-R also minimizes the impact to users by using the OSX “converged interface” model to activate the z/OS RNIC adapters. In this approach, you must only define the OSD interface, and z/OS Communications Server dynamically defines and activates the associated RNIC interfaces when SMC-R is enabled.

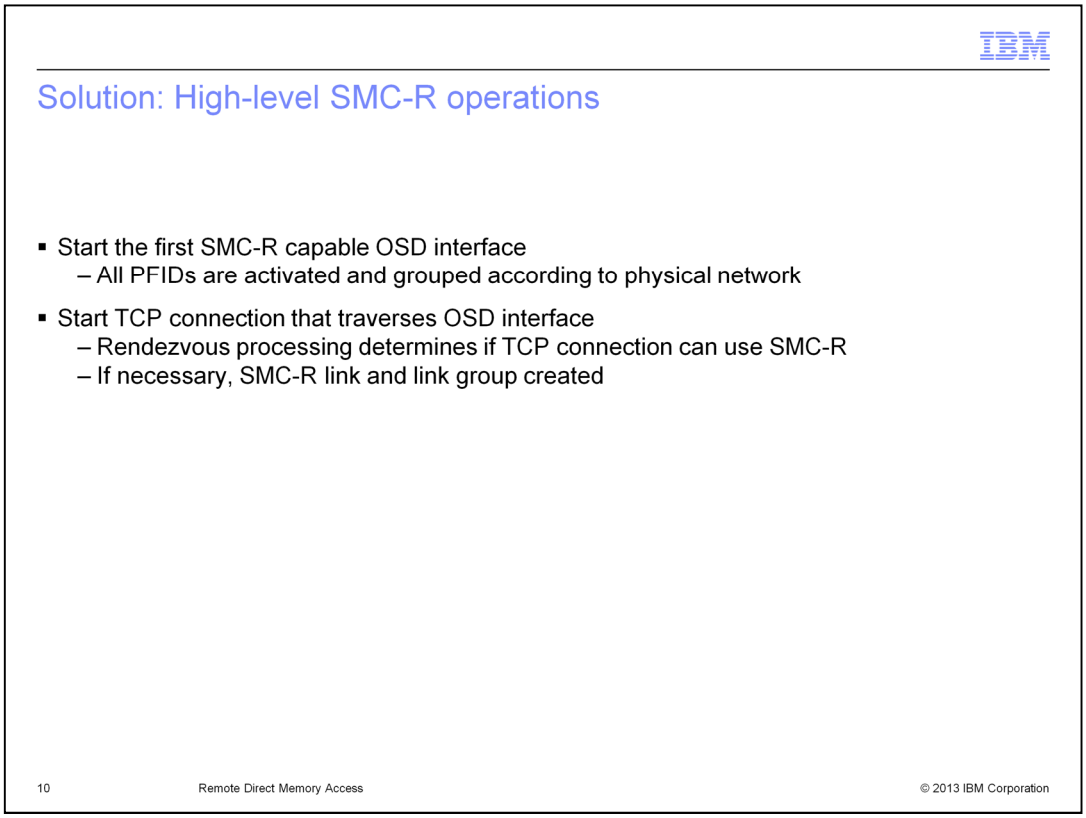
SMC-R is the only RDMA exploitation model provided by z/OS Communications Server. Applications cannot natively exploit RDMA in the z/OS Communications Server environment. In addition, the SMC-R protocols write data into the peer's memory, but data is never read from the peer's memory.



Configuring SMC-R within z/OS Communications Server is relatively simple. You must define, using the new GLOBALCONFIG SMCR parameter, the PCI Express (PCIe) function (PFID) values that represent the RNIC adapters available to this TCP/IP stack. The PFID values are defined for the system using HCD, and the numbers assigned there are specified on the GLOBALCONFIG SMCR parameter. A maximum of 16 PFID values can be coded for a given TCP/IP stack, and a maximum of eight stacks can share a single PFID within the LPAR. Additional information can be specified on the GLOBALCONFIG SMCR parameter, but only the PFID values are required for SMC-R to be operational.

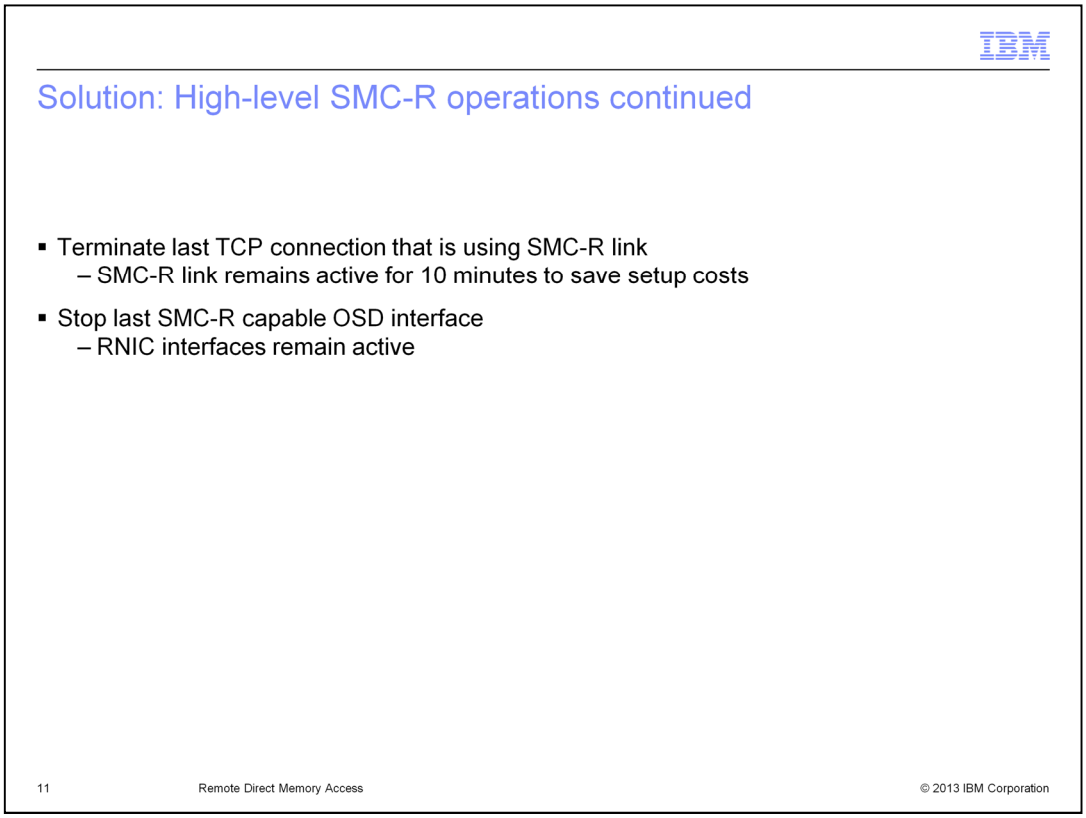


In addition to the GLOBALCONFIG SMCR statement, you can optionally define which IPAQENET and IPAQENET6 interfaces are eligible to use SMC-R. Only those interfaces with CHPIDTYPE OSD specified are able to use SMC-R, and they are eligible for SMC-R by default. When the first SMC-R capable OSD interface is started, z/OS Communications Server automatically starts the RNIC interfaces defined by the SMCR PFID values. Once the RNIC interfaces are active, the TCP/IP stack can use SMC-R.



SMC-R operations are initiated as part of processing involving SMC-R capable OSD interfaces, in a manner similar to the OSX “converged interface” model. All defined RNIC interfaces are started dynamically when the first SMC-R capable OSD interface is started. RNIC interfaces differ from the OSX model because z/OS Communications Server never automatically stops the RNIC interfaces, even when the last SMC-R capable OSD interface is stopped.

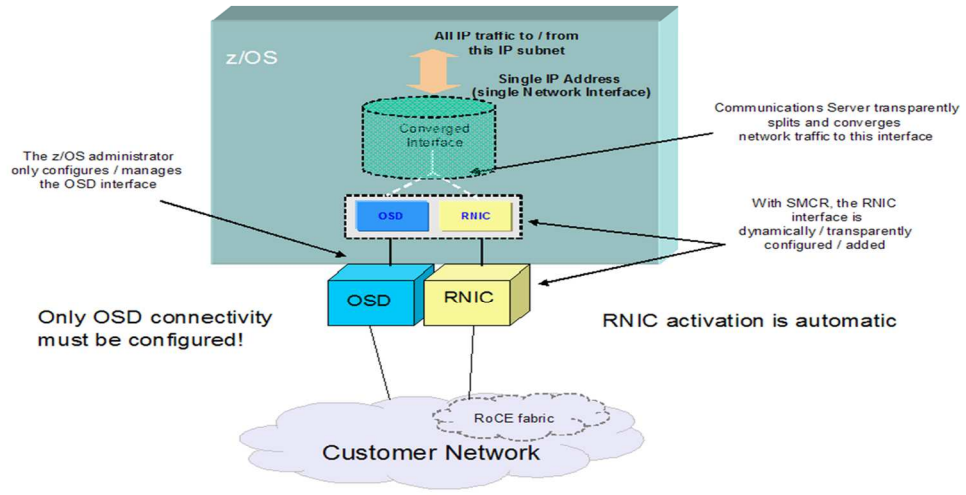
The decision to use or not use SMC-R for a given TCP connection is made during TCP connection establishment. The protocols for making that determination are called rendezvous processing, and they are discussed further in later slides. If this is the first TCP connection between this host and the peer node, additional SMC-R processing is performed to create the logical SMC-R connection between the two hosts. This connection, or SMC-R link, is also discussed in more detail in later slides.



Because of the overhead involved in starting SMC-R links, z/OS Communications Server does not terminate SMC-R links immediately when they are not being used by any TCP connections. The links remain active for approximately 10 minutes, in case new TCP connections between the two hosts are established.

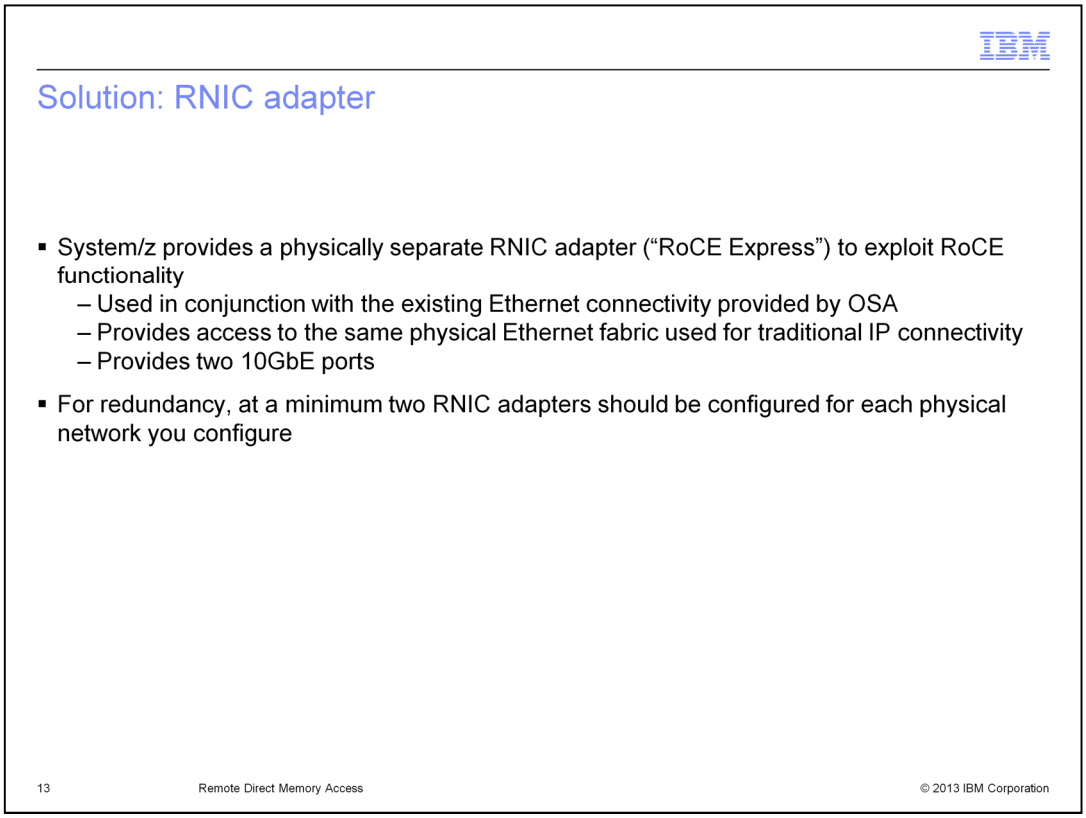
Solution: RNIC and OSD interaction

- RNIC activation is initiated as part of OSD interface activation
  - Assuming OSD defined using INTERFACE statement



This chart graphically shows the relationship of the RNIC and OSD interfaces. Similar to IQDX interfaces, the RNIC interface is not started directly, but rather is started dynamically by the TCP/IP stack when an SMC-R capable OSD interface is started. This means that you perform the majority of your operations using the OSD interface, while allowing z/OS Communications Server to manage the RNIC interfaces for you.

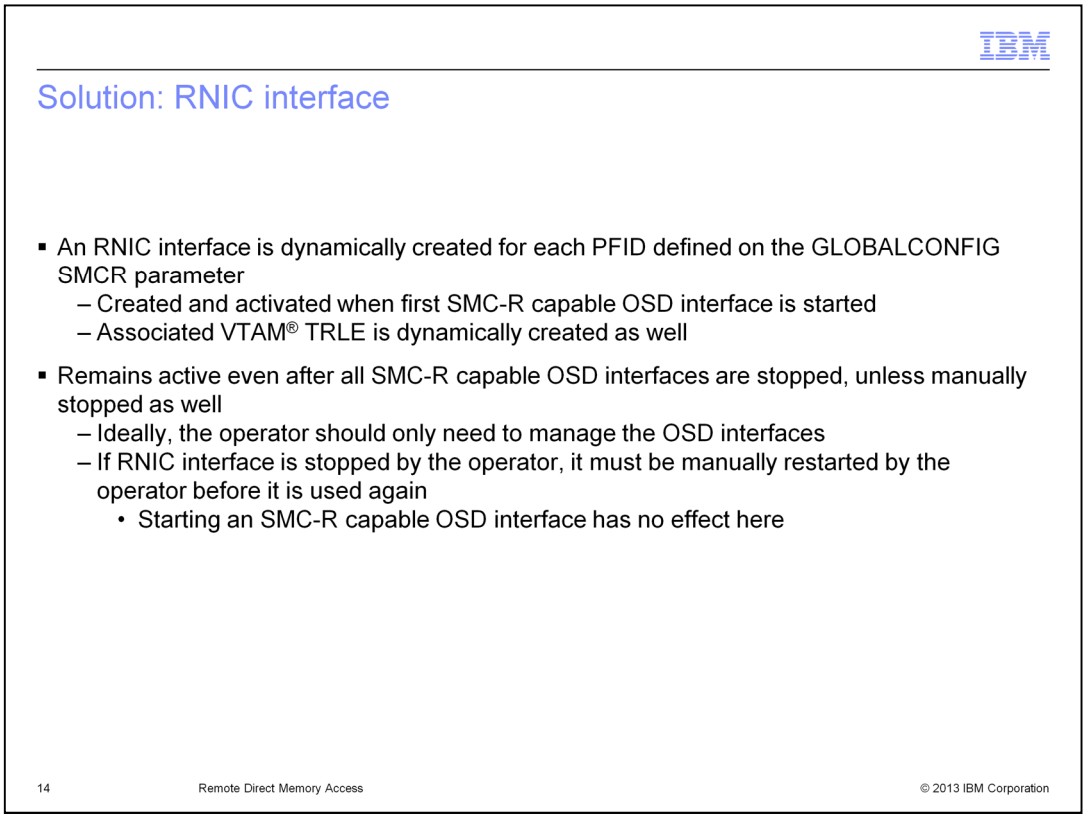
An OSD interface must be defined using an IPAQENET or IPAQENET6 INTERFACE statement to be eligible for SMC-R processing.



System/z provides a new RNIC adapter called “RoCE Express” for use with RoCE processing. The adapter is physically separate from the OSA adapters, but can access the same physical Ethernet fabrics as the OSA adapters. This means that the same Ethernet fabric can be used for both traditional IP connectivity and RDMA processing.

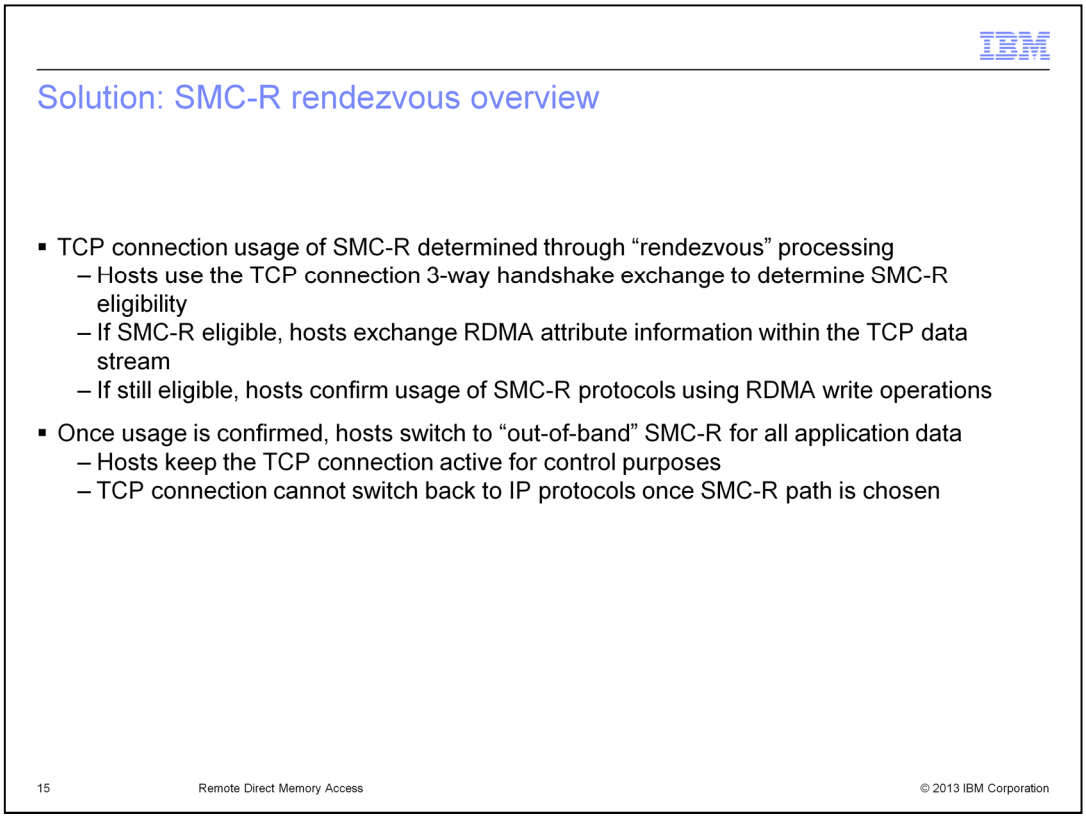
The System/z RNIC adapter supports two 10GbE ports, although z/OS Communications Server will only use one of the ports for SMC-R processing.

The topics of redundancy and physical networks are covered in depth later, but you should configure at least two RNIC adapters for each physical network that you plan to use. The use of the two RNIC adapters not only provides redundancy, but also allows z/OS Communications Server to load balance TCP connections across the RNIC adapters.



You use the GLOBALCONFIG SMCR parameter to define which RNIC interfaces a TCP/IP stack can use for SMC-R processing. z/OS Communications Server dynamically creates the RNIC interfaces and the VTAM TRLE that represents the RNIC adapter.

The intent of the “converged interface” model is that the operator does not have to manage the RNIC interfaces, bur rather just manages the OSD interfaces. RNIC interfaces, unlike IQDX interfaces, remain active even after the last associated OSD interface is stopped. If you want to stop the RNIC interface, the operator must manually stop them. The act of manually stopping the RNIC interface shifts the responsibility of managing the RNIC interface from z/OS Communications Server to the operator. That means the operator must also manually restart the RNIC interface if the RNIC is to be used again for SMC-R processing.



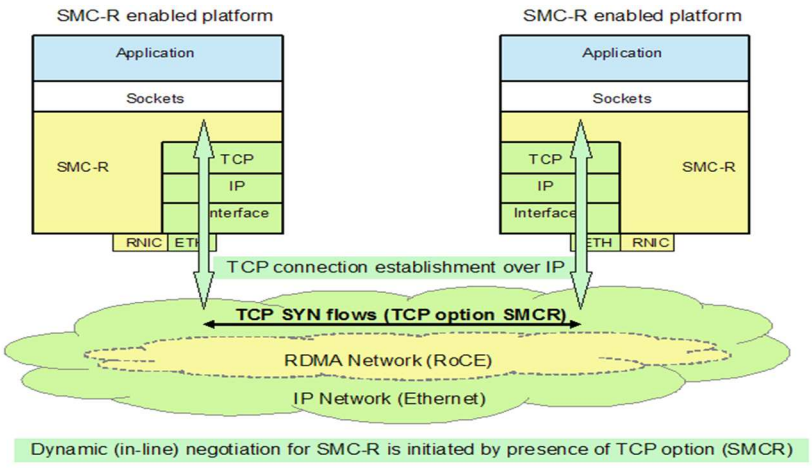
The protocol for determining whether a TCP connection can use SMC-R processing is called rendezvous processing. There are three stages to rendezvous processing. The first involves an exchange of a new SMCR option during the traditional 3-way handshake that establishes a TCP connection. If both client and server indicate that SMC-R is possible, then Connection Layer Control (CLC) messages are exchanged in-band over the TCP connection. These CLC messages exchange RDMA attribute information necessary to select, or create, the underlying logical connection, or SMC-R link, to be used by this TCP connection. Finally, Link Layer Control (LLC) messages verify that the SMC-R link is operational by performing RDMA write operations.

If rendezvous processing successfully selects or creates an SMC-R link to be used, subsequent application socket data is exchanged “out-of-band” using the SMC-R link. Once the choice is made to use SMC-R protocols, the TCP connection cannot revert to using traditional IP protocols, even if the SMC-R link or RNIC interface encounter errors subsequently. z/OS Communications Server prevents any application socket data from being exchanged until the choice of SMC-R or IP protocols has been made.

The TCP connection stays active for control flow and connection termination processing, but otherwise remains idle.

Solution: SMC-R rendezvous processing (1 of 2)

- SMC-R capability negotiated during TCP connection establishment
- SMC-R attributes exchanged within TCP connection data stream



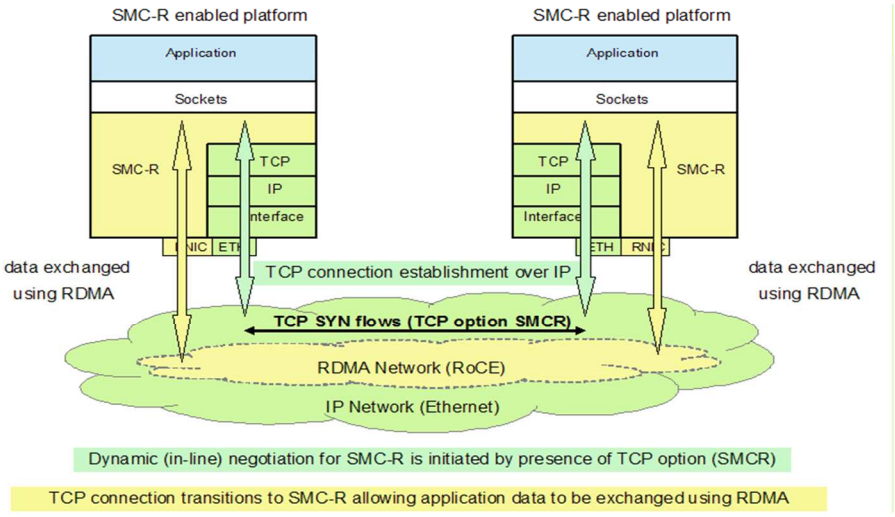
This diagram shows the exchange of the new TCP option SMCR during the 3-way handshake that establishes a TCP connection. Both the client and the server must include the new option in order for the TCP connection to be eligible for SMC-R processing.

Once that exchange is complete, new SMC-R CLC messages are exchanged over the TCP connection. The CLC messages exchange information such as global ID, virtual MAC, queue pair (QP) number, and the remote memory buffer (RMB) address that has been allocated for the peer to use. The QP and RMB concepts are discussed in more detail on later slides.

The TCP/IP stack runs a series of timers during rendezvous processing to ensure that the SMC-R negotiation completes in a timely fashion. If any step during the negotiation takes too long to complete, the attempt to use SMC-R for this TCP connection is abandoned and the connection uses traditional IP protocols.

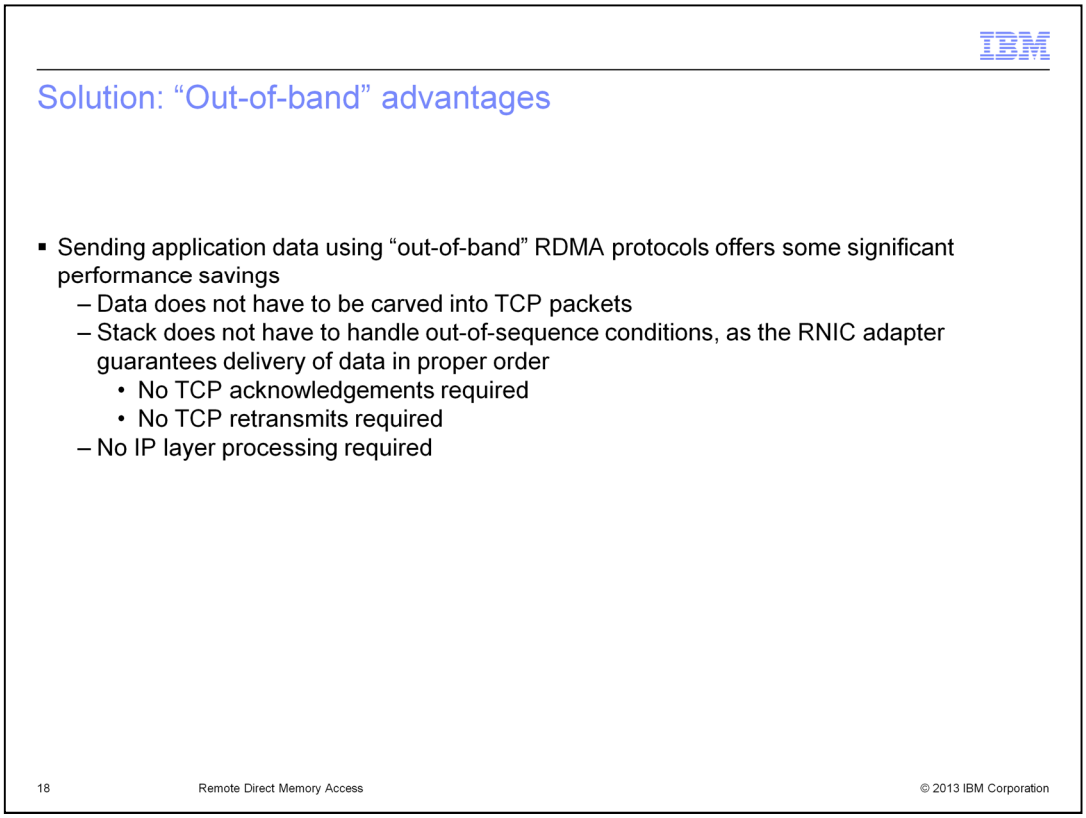
Solution: SMC-R rendezvous processing (2 of 2)

- Application data exchanged using RDMA protocols



Assuming that the CLC portion of rendezvous processing completes successfully, LLC messages are exchanged to verify that the RDMA fabric is operational. If these LLC messages are successful, then the application socket data flows “out-of-band” across the RDMA fabric. If the exchange is unsuccessful, or times out, the TCP connection reverts to using IP protocols.

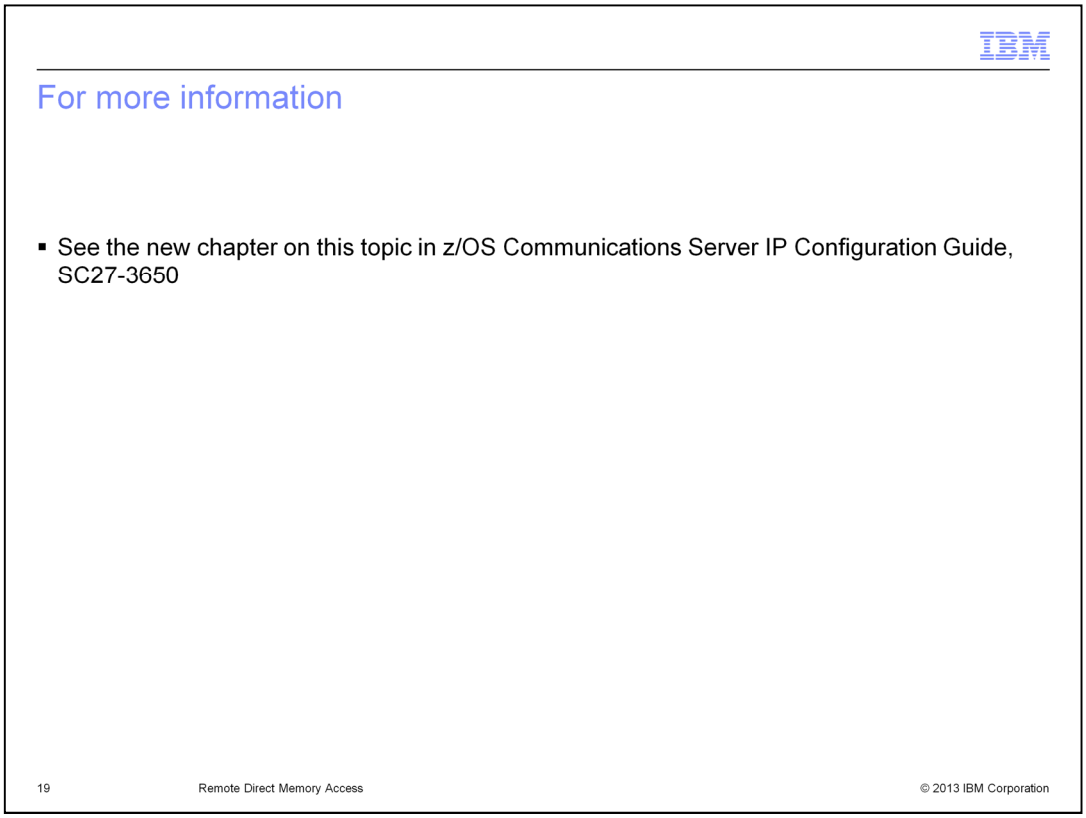
Additional LLC messages are exchanged between the peers in order to manage the SMC-R link. These messages are exchanged as part of recovery processing should one of the RNIC adapters fail, requiring the active TCP connections to move to the other RNIC adapter. This concept of failover processing is covered in later slides.



Significant performance savings are achieved by switching to “out-of-band” RDMA protocols. One advantage is that the TCP/IP stack does not have to break the application socket data into smaller packets to be transported across the IP fabric. Instead, the data is moved as larger chunks of data, up to the size of the remote memory buffer made available by the peer. z/OS Communications Server selects a buffer size for the peer based on the receive buffer size specified by the local application.

The RNIC adapter is designed to guarantee delivery of the RDMA data in order to the peer. This means that traditional TCP layer processing for retransmitting lost packets is not necessary with SMC-R. It also means that the TCP layer does not have to exchange acknowledgements to verify that data has been received properly. This greatly streamlines the TCP layer processing for SMC-R, providing additional performance gains.

Additional gains are achieved because the entire IP layer is bypassed in favor of a new, more streamlined SMC-R processing layer.



Shared memory communications is a major new function. The IP Configuration Guide goes into a lot of detail about the new concepts involved, along with configuration and implementation details.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

1. Did you find this module useful?
2. Did it help you solve a problem or answer a question?
3. Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_cs21smc.ppt

This module is also available in PDF format at: ../cs21smc.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information