Shared Memory Communications for Linux on IBM Z

Jing Zhang
KVM on IBM Z Development

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*	DB2*	HiperSockets*	MQSeries*	PowerHA*	RMF	System z*	zEnterprise*	z/VM*
BladeCenter*	DFSMS	HyperSwap	NetView*	PR/SM	Smarter Planet*	System z10*	z10	z/VSE*
CICS*	EASY Tier	IMS	OMEGAMON*	PureSystems	Storwize*	Tivoli*	z10 EC	
Cognos*	FICON*	InfiniBand*	Parallel Sysplex*	Rational*	System Storage*	WebSphere*	z/OS*	
DataPower*	GDPS*	Lotus*	POWER7*	RACF*	System x*	XIV*		

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.

TEALEAF is a registered trademark of Tealeaf, an IBM Company.

Windows Server and the Windows logo are trademarks of the Microsoft group of countries.

Worklight is a trademark or registered trademark of Worklight, an IBM Company.

UNIX is a registered trademark of The Open Group in the United States and other countries.

* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

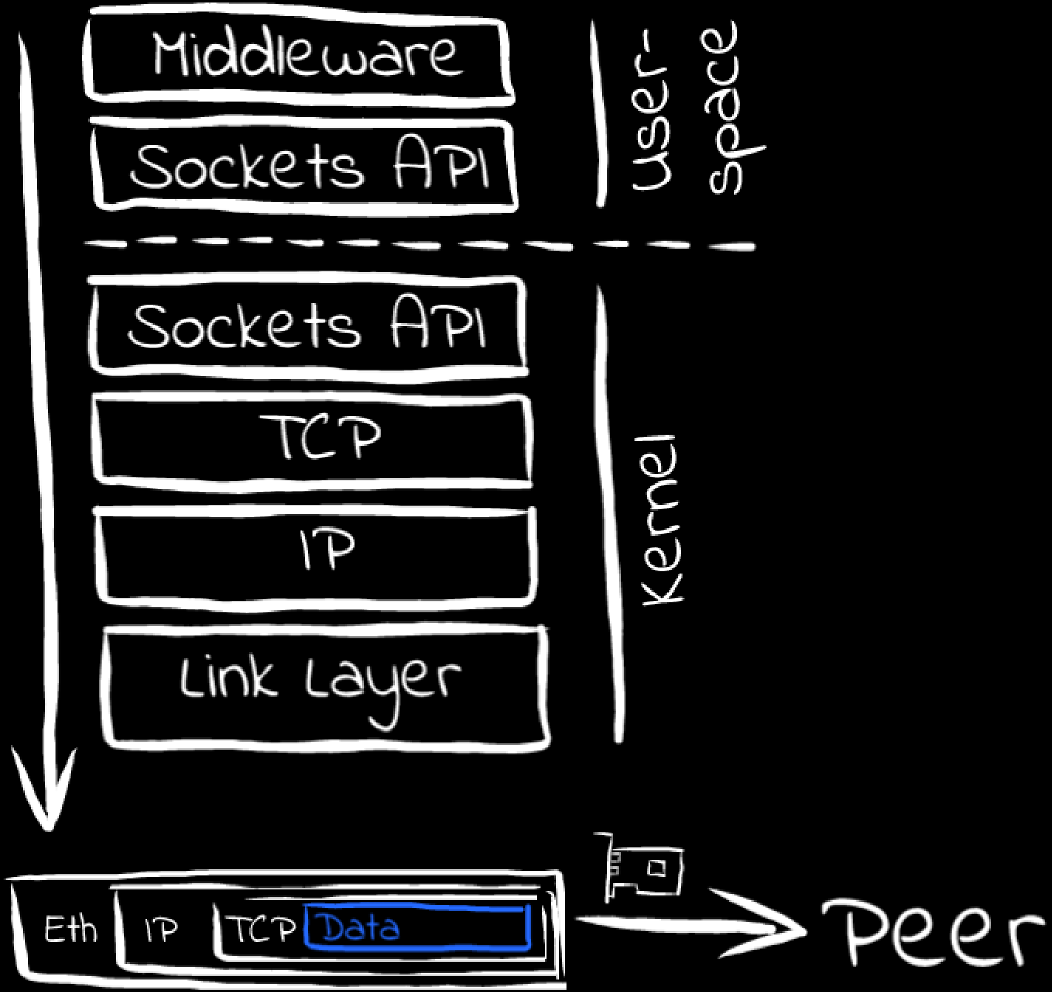
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

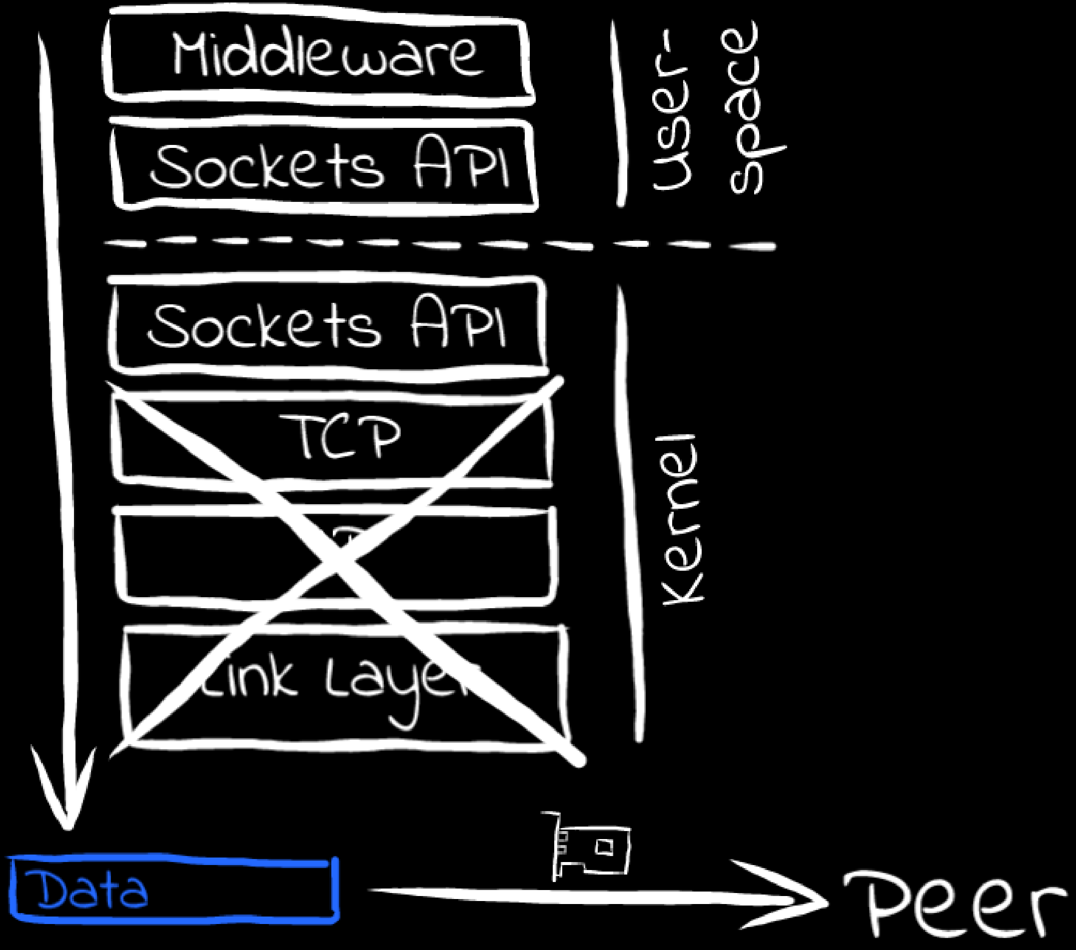
Agenda

- SMC Basics
  - Motivation
  - The SMC Protocol
  - Benefits
- SMC for Linux on Z
  - SMC-D and SMC-R
  - smc-tools
- SMC in Action
  - Usage Examples
  - Deploying SMC
  - Tips & Tricks
- Miscellaneous

What sending data through BSD sockets looks like

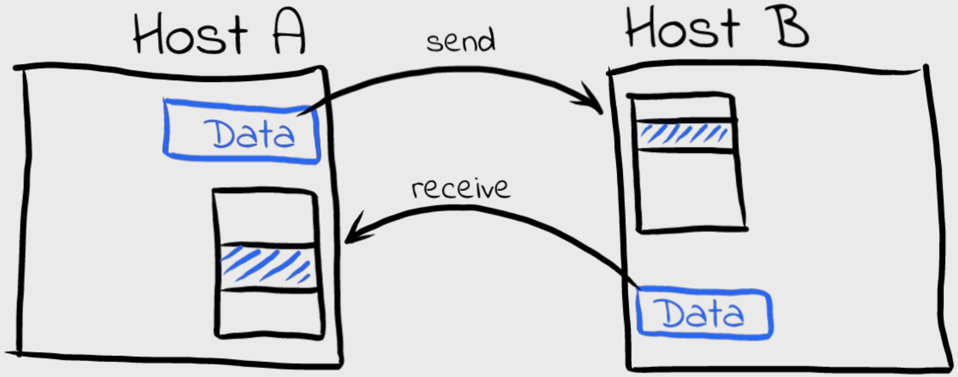


What if we had a simple buffer to write data to and let hardware do the rest...?



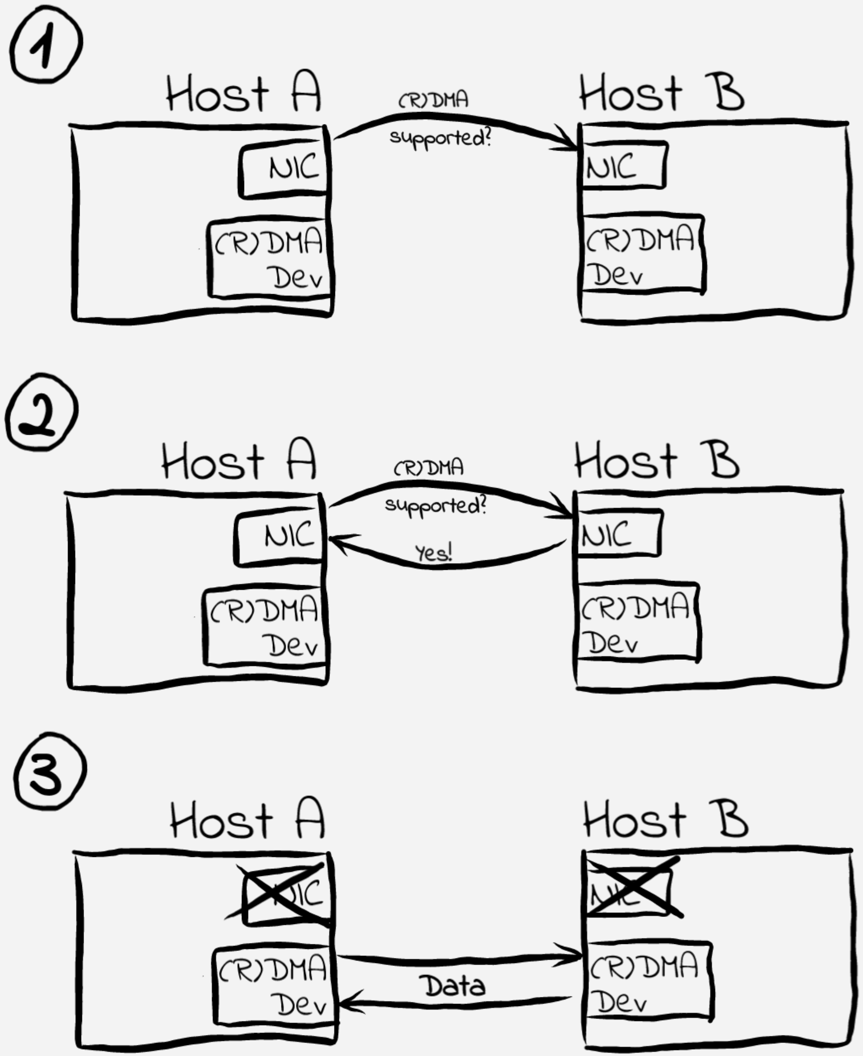
The RDMA Approach

- RDMA (**R**emote **D**irect **M**emory **A**ccess) based technology originating from Infiniband (IB)
- Enables a host to read or write directly from/to a remote host's memory with drastically reduced use of remote host's CPU (interrupts required for notification only)
- Native / direct application exploitation requires rewrite of network-related program logic, deep level of expertise in RDMA and a new programming model
- Therefore, provide a transparent approach:
  - **SMC-R**: Use RDMA over Converged Ethernet (RoCE) technology
    - Unlike IB, RoCE does not require unique network components (host adapters, switches, security controls, etc.)
    - Utilize existing Ethernet fabric with RDMA capable NICs and switches
  - **SMC-D**: Use DMA when both hosts are within a Z system via virtual PCI device



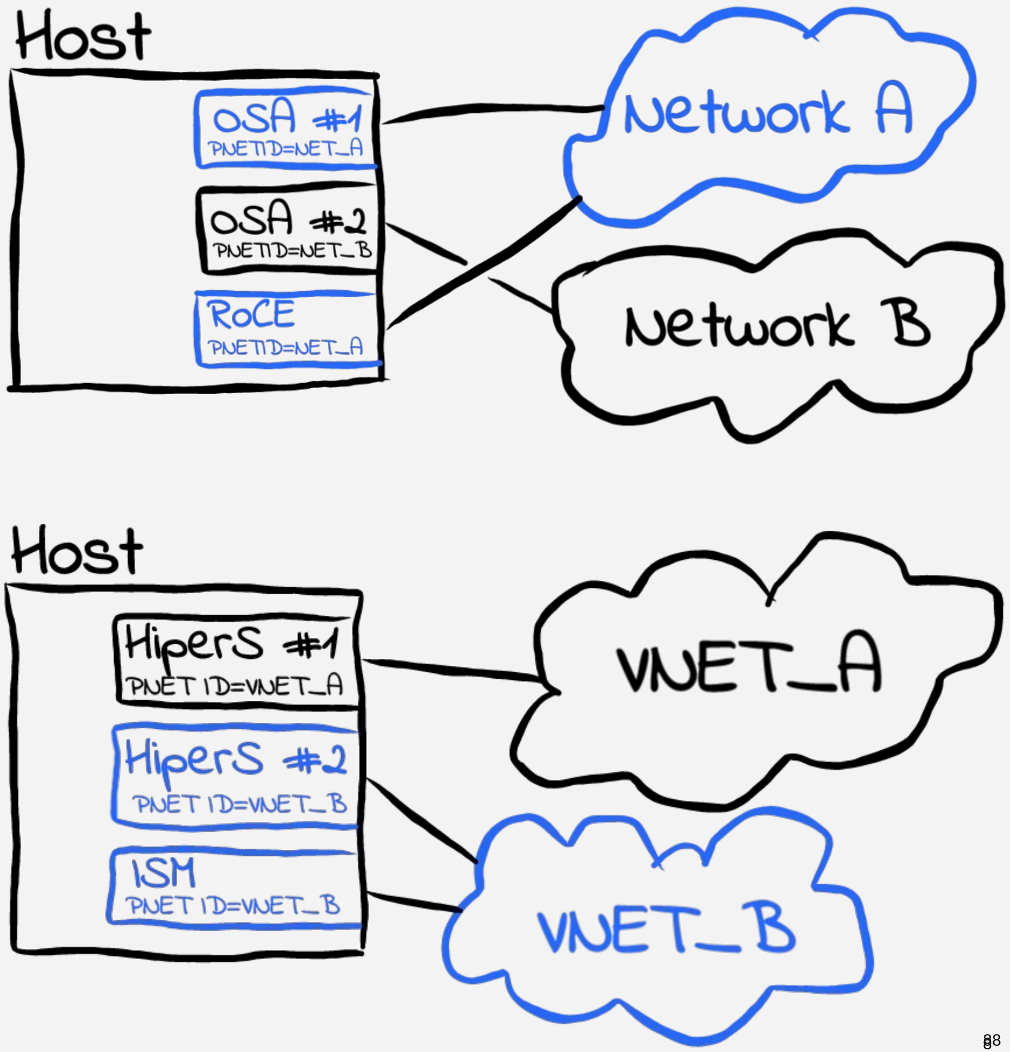
Overview

- For each new TCP connection:
  - Start out with a regular TCP/IP connection
  - Advertise and negotiate details about the peers’ (R)DMA capabilities
  - Switch over to an (R)DMA device for actual traffic depending on the peers’ capabilities
  - Original connection through NICs remains active but idle



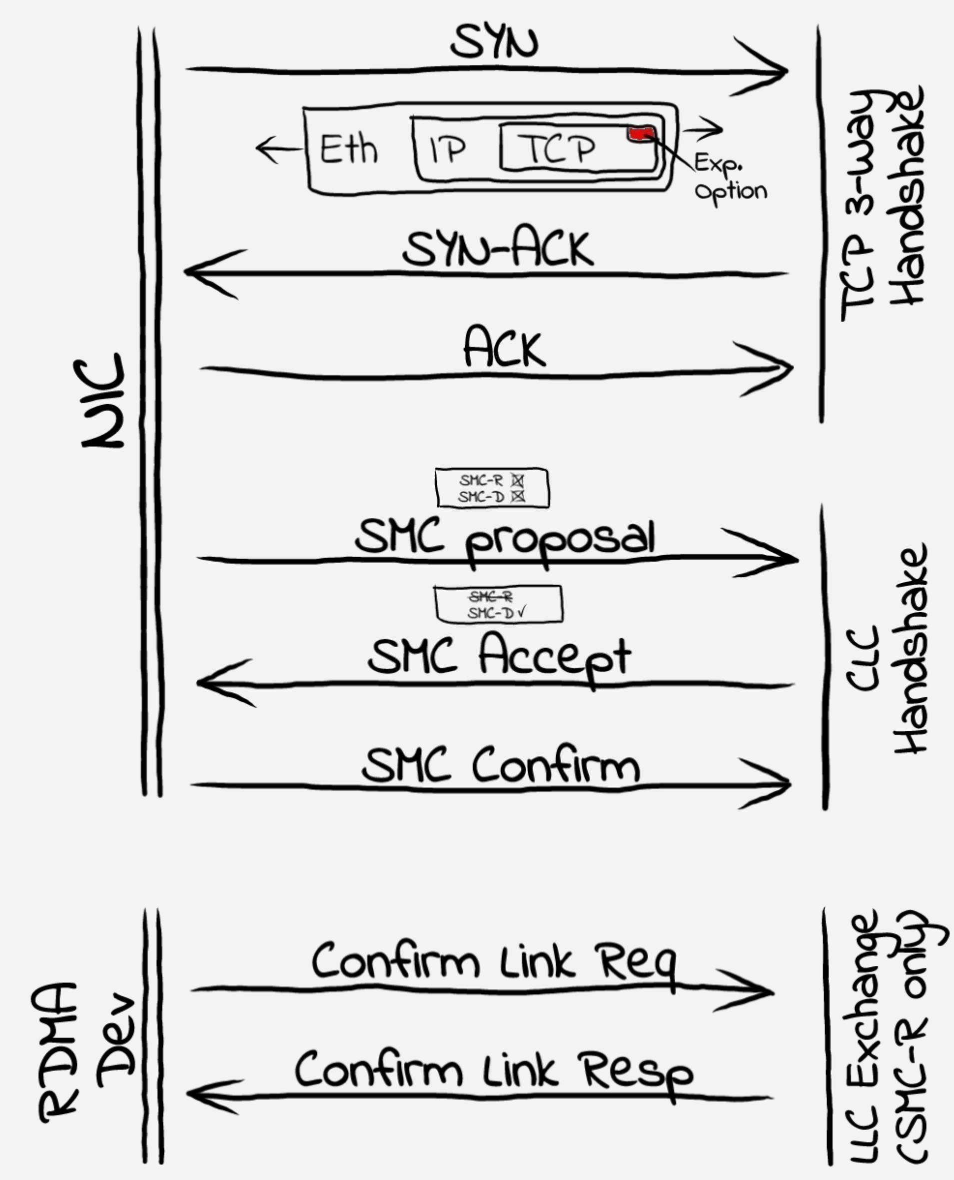
PNET IDs

- PNET ID: Physical network identifier
- Customer-defined value to logically group NICs and RDMA adapters connected to the same physical network
- Define in
  - IOCDS, or
  - using `smc_pnet` tool (SMC-R only)
- Typically associate
  - OSA and RoCE cards, or
  - HiperSockets and ISM devices



Protocol Details

- Start out with regular transport via e.g. OSA or HiperSockets
- SMC capability indicated by TCP experimental option during TCP 3-way handshake
- CLC handshake used to exchange info for SMC transport → adds additional round-trips
- Fallback to regular TCP/IP in case of failure at any point during setup
- In case of matching capabilities: Switch to (R)DMA device
- No fallback to or usage of regular TCP/IP connection beyond this point!
- See RFC 7609 “IBM's Shared Memory Communications over RDMA (SMC-R) Protocol” (https://tools.ietf.org/html/rfc7609) for a detailed description

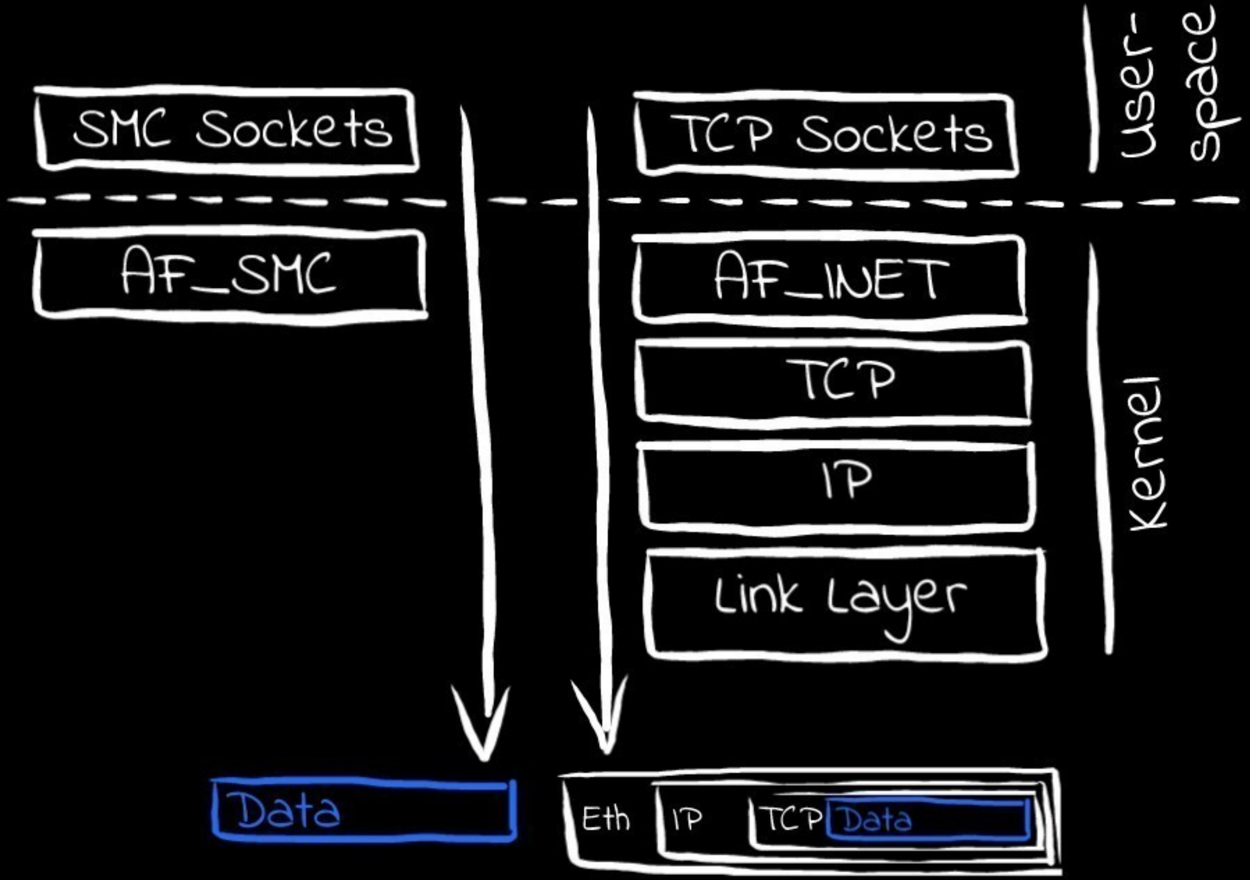


Why a Hybrid Protocol?

Leverages key existing attributes:

- Follows standard TCP/IP connection setup
- Dynamically switches to (R)DMA (SMC)
- Preserves critical operational and network management TCP/IP features such as:
  - Minimal (or zero) IP topology changes
  - Transparent to channel bonding, load balancers and VLANs
  - Preserves existing IP security model (e.g. IP filters, policy, VLANs, SSL etc.)
  - Minimal network admin / management changes
  - Built-in failover capabilities for RDMA devices

Less latency
Lower CPU usage



Full BSD sockets API compatibility

1) Install kernel with SMC-D/R support
2) Install smc-tools (shipped with distro, or see https://ibm.biz/BdiZ5m)
3) In IOCDS:
   - Define (R)DMA devices
   - Assign PNET IDs to networking devices
   - Alternative: Use `smc_pnet` in Linux (SMC-R only)
4) In applications' `socket()` calls, replace `AF_INET` with `AF_SMC`, i.e.:

```
int s, ipv6 = 0;
s = socket(AF_SMC, SOCK_STREAM, ipv6);
```

Run your applications unmodified

- SMC is transparent to existing applications – no changes required
- Use `smc_run`, also provided by smc-tools:

```
smc_run <my_application>
```

- Or use preload library directly, provided by smc-tools, to enable existing applications:

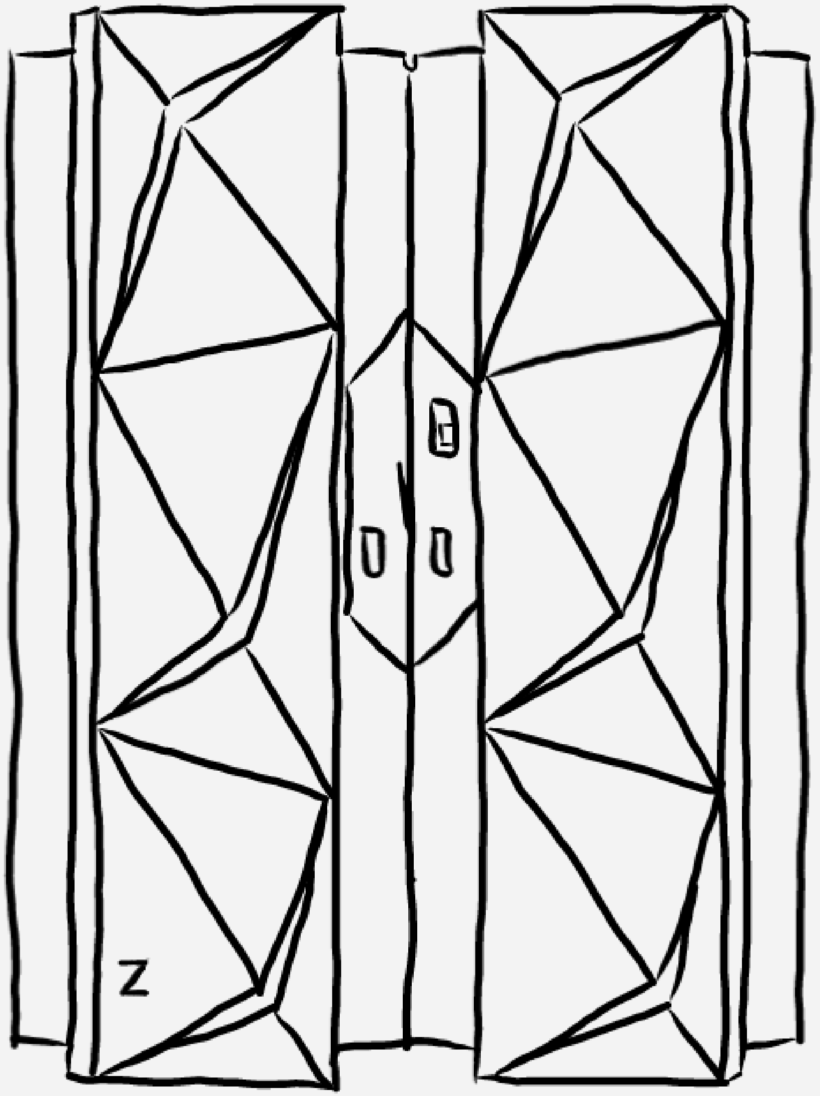
```
export LD_PRELOAD=libsmc_preload.so
```

Preserve Existing Security Model

- The hybrid nature of SMC (beginning with regular TCP/IP, then switching to SMC) allows existing IP and TCP layer (i.e. IP and port-based) security features to automatically apply to SMC connections.
- This includes:
  - SSL/TLS
  - IP Filters, Traffic regulation, Intrusion detection systems
  - Auditing based on IP addresses and ports
- Not supported:
  - IPSec tunnels
  - Deep Packet Inspection
- No changes from a user perspective required

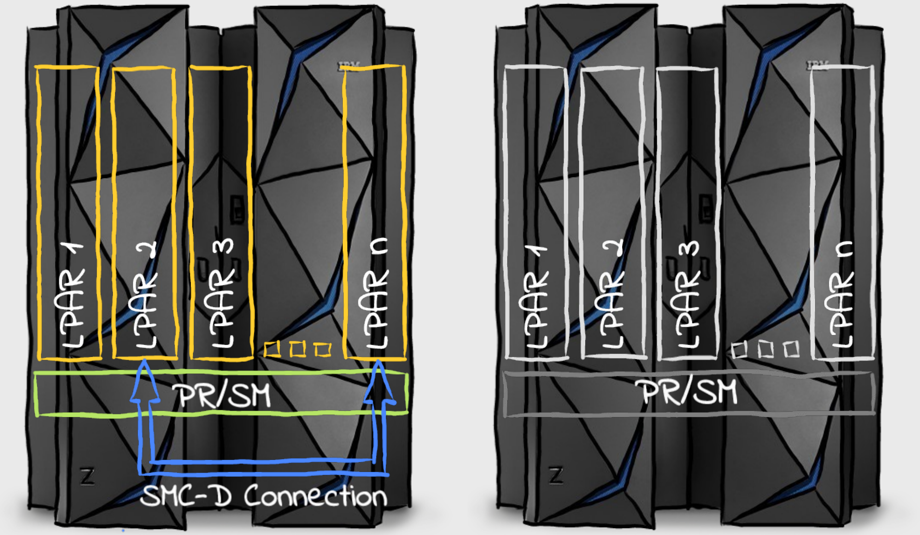
Agenda

- SMC Basics
  - Motivation
  - The SMC Protocol
  - Benefits
- SMC for Linux on Z
  - SMC-D and SMC-R
  - smc-tools
- SMC in Action
  - Usage Examples
  - Deploying SMC
  - Tips & Tricks
- Miscellaneous



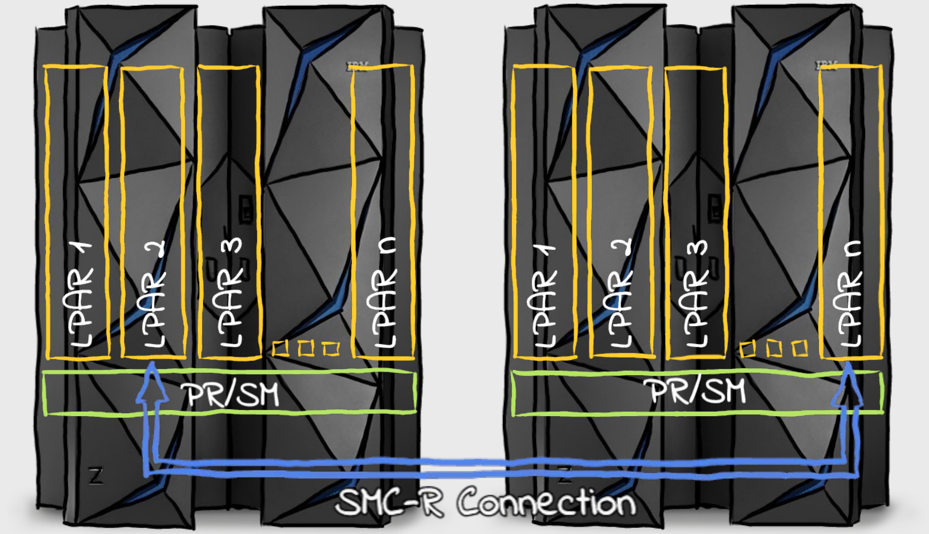
SMC-D Overview

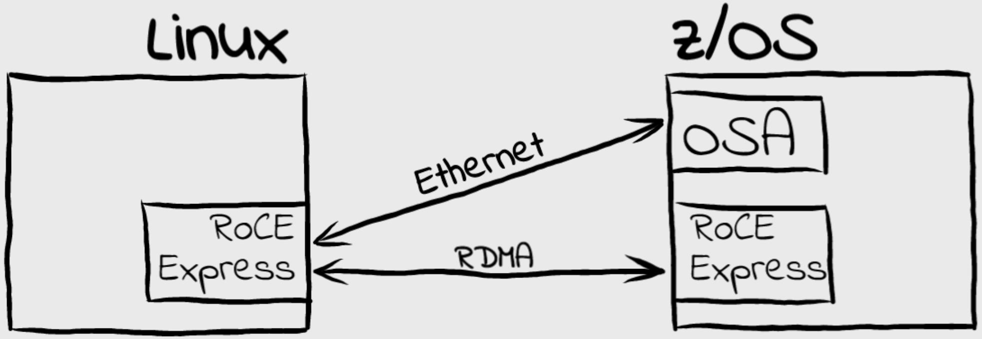
- Intra-CEC connectivity using **Internal Shared Memory** (ISM) devices
- IBM Z hardware requirements
  - IBM z13 (requires driver level 27 (GA2)) and z13s, or later
  - LinuxONE Emperor and LinuxONE Rockhopper, or later
  - Classic mode only (i.e. DPM not supported)
- ISM devices
  - Virtual PCI network adapter of new VCHID type ISM
    - No PCI bus usage
    - No extra hardware required
  - Provides access to memory shared between LPARs
  - 32 ISM VCHIDs per CPC, 255 VFs per VCHID (8K VFs per CPC total) I.e. the maximum no. of virtual servers that can communicate over the same ISM VCHID is 255
  - Each ISM VCHID represents a unique (isolated) internal network, each having a unique Physical Network ID
- PNET ID configuration
  - IOCDS only
  - Use HiperSockets, OSA or RoCE cards for regular connectivity



SMC-R Overview

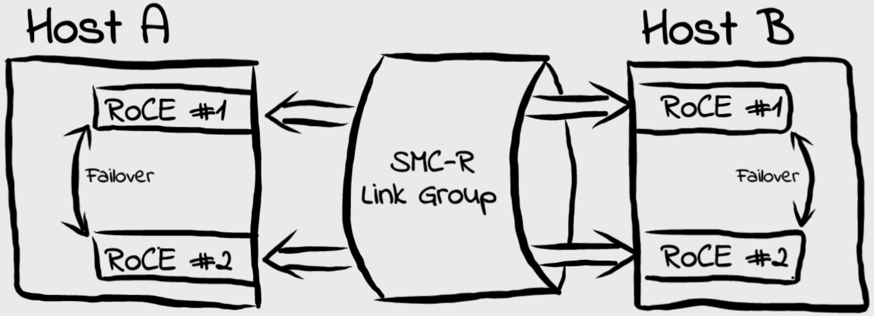
- Cross-CEC connectivity using **RoCE Express** cards
- IBM Z hardware requirements
  - IBM z12EC and z12BC or later
  - LinuxONE Emperor and Rockhopper or later
  - Classic and DPM mode supported
- RoCE Express cards
  - RoCE Express & RoCE Express2 cards supported
  - Switches need to support and enable Global Pause (standard Ethernet switch flow control feature as described in IEEE 802.3x)
- **Note**:
  Linux on Z can use a single RoCE card for regular and RDMA traffic!
- PNET ID configuration
  - IOCDS or `smc_pnet` (→see smc-tools package)
  - Use OSA or RoCE card for regular connectivity

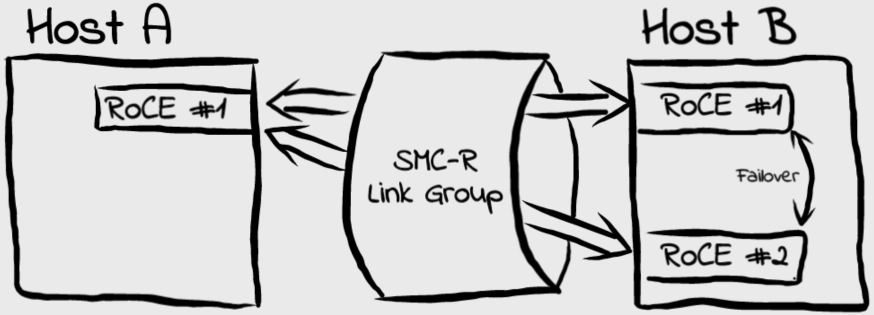


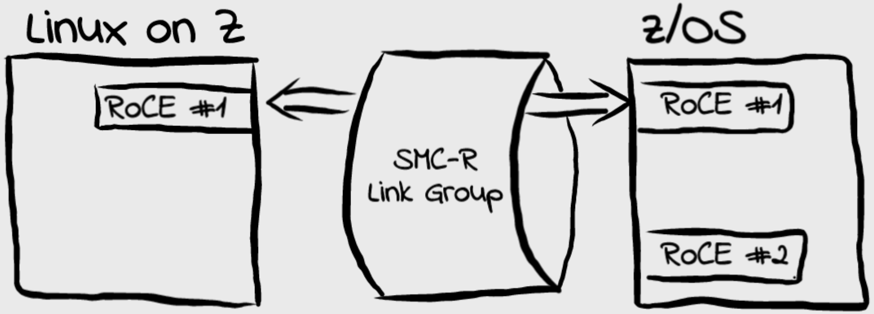


SMC-R Link Groups

- SMC-R **link groups** provide for load balancing and recovery
  - New TCP connection is assigned to the SMC-R link with the fewest TCP connections
  - Load balancing only performed when multiple RoCE Express adapters are available at each peer
- **Full redundancy** requires:
  - Two or more RoCE Express adapters at each peer
  - Unique system internal paths for the RoCE Express adapters
  - Unique physical RoCE switches
- **Partial redundancy** still possible in the absence of one or more of these conditions
- Linux on Z:
  - No failover support (yet)







Comparison

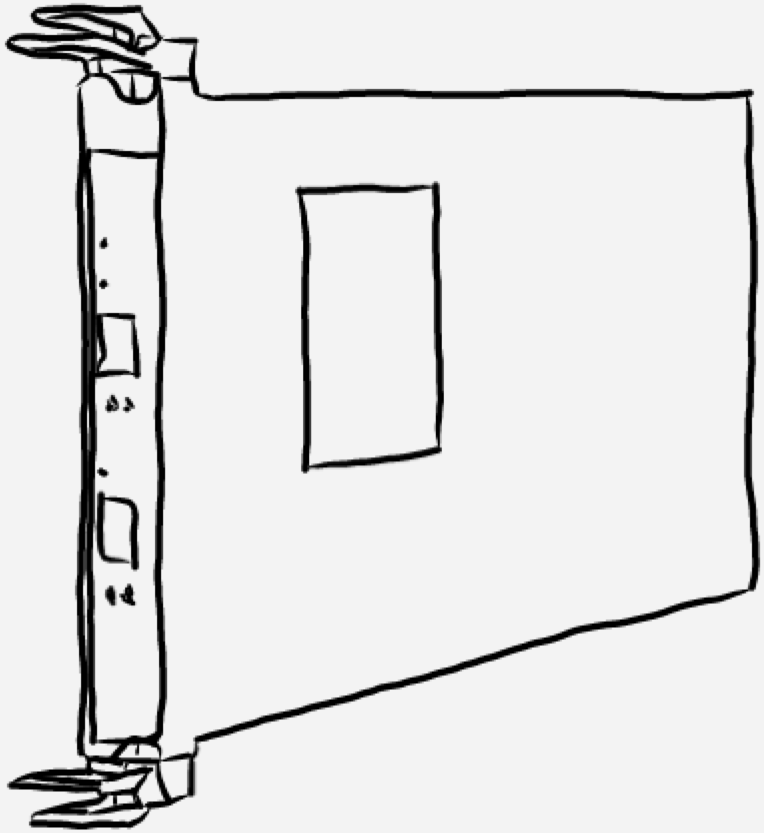
Feature	SMC-R	SMC-D
Intra-CEC	yes	yes
Cross-CEC	yes	no
RDMA Device	RoCE	ISM
Interface Type	PCI	PCI
Bus used	PCI	-
PNET ID Definition	IOCDS, or `smc_pnet`	IOCDS
Failover	tbd	N/a
Upstream Status	Initial code upstream in Linux kernel 4.18	Initial code upstream in Linux kernel 4.19 (anticipated)

smc-tools Package Overview

- Current version: v1.1.0
- smc-tools provides the following commands:
  - `smc_pnet`
    - Associate NICs via PNET ID in software
    - Does **not** modify/create IOCDS entries
    - Also works with bonding and VLAN devices
    - **Note**: PNET IDs defined in IOCDS always override
  - `smc_run`: Enable a binary application to use SMC.
  - `smcss`: Information about SMC-enabled sockets and link groups. Includes information on SMC mode used, as well as TCP fallbacks

Agenda

- SMC Basics
  - Motivation
  - The SMC Protocol
  - Benefits
- SMC for Linux on Z
  - SMC-D and SMC-R
  - smc-tools
- SMC in Action
  - Usage Examples
  - Deploying SMC
  - Tips & Tricks
- Miscellaneous



Prerequisites

- □**Direct connectivity** over same IP subnet. I.e. no routed traffic, no peers in different IP subnets, currently built on RoCE V1
- □(R)DMA device(s) attached and configured
- □PNET IDs assigned
- □Linux kernel supports SMC-R and/or SMC-D
- □**TCP only**, i.e. no UDP
- □No IPsec (SSL/TLS works)
- □No NAT (violates same subnet prerequisite)

Usage Example: SMC-D

Prerequisites: Applications in different LPARs on same CEC communicating through HiperSockets. ISM (FID: 80) and HiperSockets devices have the same PNET ID configured in IOCDS in each LPAR.

```
# Hotplug ISM device if not yet visible via 'lspci' command (see next step)
$ echo 1 > /sys/bus/pci/slots/00000080/power

# Verify presence of ISM device
$ lspci
0001:00:00.0 NonVGA unclassified device: IBM Internal Shared Memory (ISM) virtual PCI device

# Run application using smc_run
$ smc_run foo_socks

# Verify that SMC is really used
$ smcss a
State          UID   Inode  Local Address          Foreign Address         Intf Mode
ACTIVE         20000 115762 10.101.4.8:60594       10.101.4.49:3220        0000 SMCD
ACTIVE         20000 112844 10.101.4.8:60592       10.101.4.49:3220        0000 SMCD
ACTIVE         20000 112605 10.101.4.8:60590       10.101.4.49:3220        0000 SMCD
```

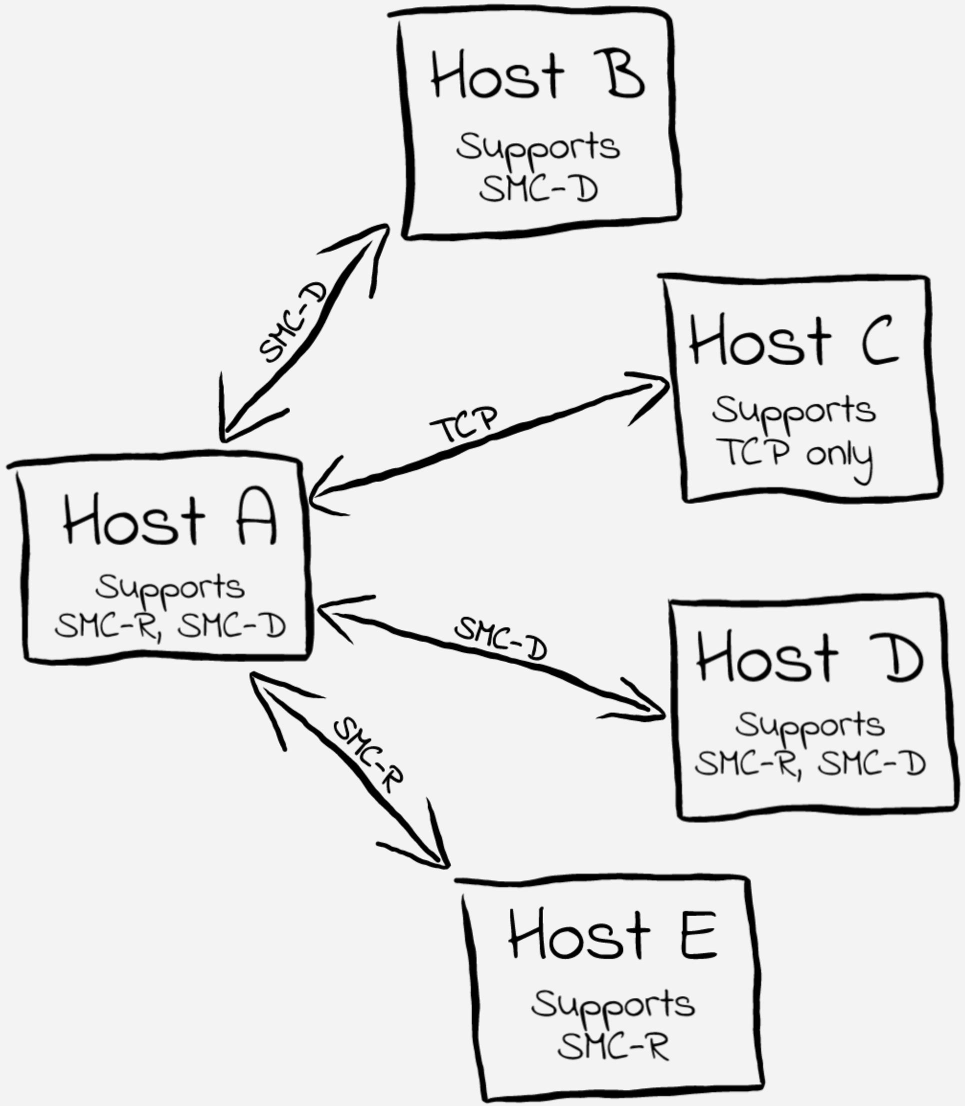
Usage Example: SMC-R

Prerequisites: Existing Applications in LPARs on separate CECs communicating through OSA card `enccw0.0.f500`.
RoCE Express adapter has network interface `ens2` and infiniband interface `mlx4_0` - we will use its 1st port. No PNET IDs configured.

```
# Verify presence of RoCE card
$ lspci
0000:00:00.0 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family [ConnectX3/ConnectX3 Pro Virtual Function]

# Set RoCE card interface UP, and verify
$ ip link set ens2 up
$ ip link show ens2
3: ens2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 82:03:14:32:f1:a0 brd ff:ff:ff:ff:ff:ff

# VLANs only: Define an interface, and assign an IP – interface does not need to be in state UP!
$ ip link add dev ens2.201 link ens2 type vlan id 201
$ ip addr add 192.168.23.42/24 dev ens2.201

# Configure PNET ID on OSA and RoCE device:
$ smc_pnet a PNET1 I enccw0.0.f500 D mlx4_0 P 1
$ smc_pnet s
PNET1 enccw0.0.f500 mlx4_0 1

# Run application using smc_run
$ smc_run foo_socks

# Verify that SMC is really used
$ smcss a
State          UID   Inode  Local Address         Foreign Address        Intf Mode
ACTIVE         20000 115762 10.101.4.8:60594      10.101.4.49:3220       0000 SMCR
```

Mixing SMC Usage

- Both variants of SMC can be used concurrently to provide an optimized solution
- Enable SMC independent of peers' capabilities; i.e. no commonality in SMC support on all peers required
- Use
  - SMC-D for local connections
  - SMC-R for remote connections
  - fall-back to regular TCP where neither SMC variant is supported



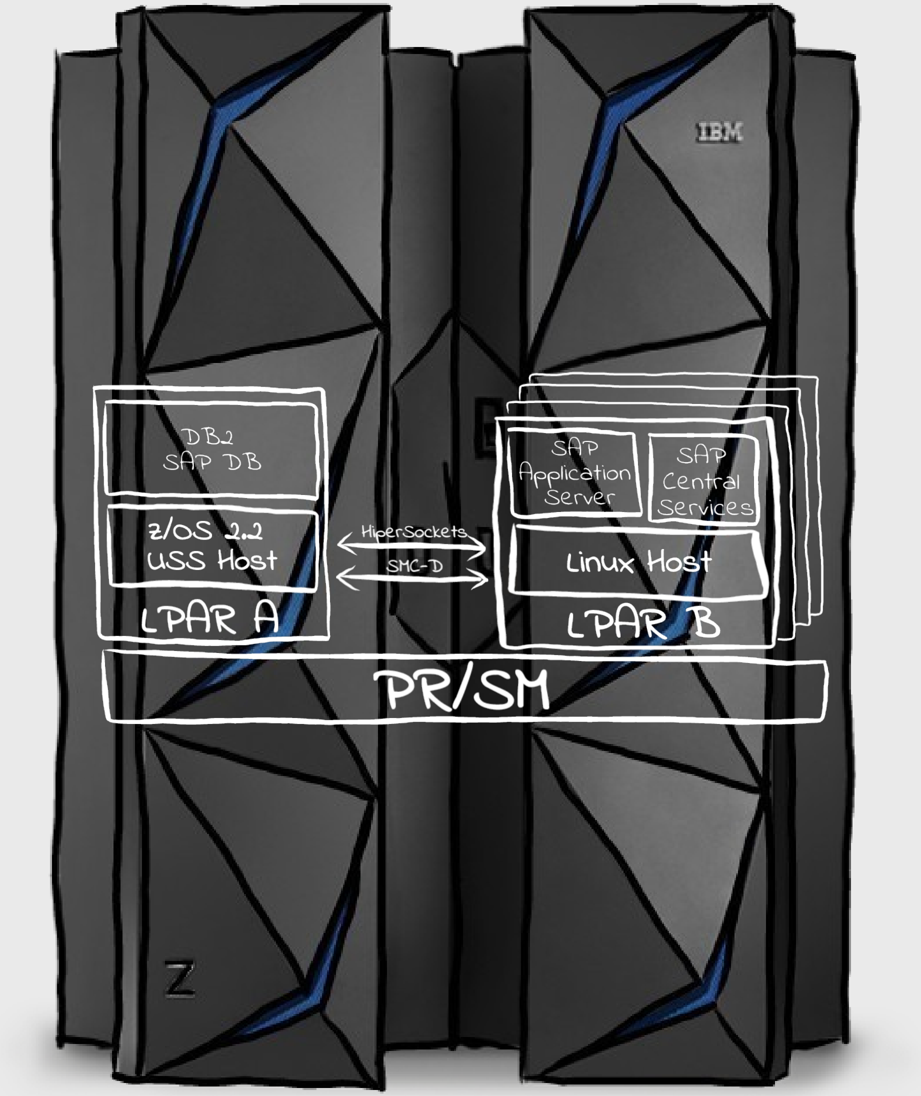
Performance Considerations

- **Expect**:
  - Reduction of CPU usage
  - Lower latency
  - Higher effective throughput
  - Higher maximum throughput (SMC-D only)
- **But consider**:
  - CLC handshake adds add’l round trips prior to actual traffic
    →Minimum number of transmits required to break even

Usage Example: SAP on IBM Z

- Deploy
  - DB2 SAP Database on z/OS
  - SAP Central Services and SAP Application Server on Linux on Z
- Provides lower latency, less CPU used

→ Higher transaction rates



Agenda

- **SMC Basics**
  - Motivation
  - The SMC Protocol
  - Benefits
- **SMC for Linux on Z**
  - SMC-D and SMC-R
  - smc-tools
- **SMC in Action**
  - Usage Examples
  - Deploying SMC
  - Tips & Tricks
- **Miscellaneous**

Summary

Key Attributes

- Leverages existing Ethernet infrastructure (SMC-R)
- Transparent to (TCP socket based) application software
- Preserves existing network addressing-based security models
- Preserves existing IP topology and network administrative and operational model
- Transparent to network components such as channel bonding and load balancers
- Built-in failover capabilities (SMC-R)

Typical Workloads To Benefit

- Transaction-oriented,
- latency-sensitive, and
- bulk data streaming, e.g. when running backups.

References

- **smc-tools Homepage**
  https://www.ibm.com/developerworks/linux/linux390/smc-tools.html
- **SMC on z/OS**
  https://www-01.ibm.com/software/network/commserver/SMC/
- **SMC-AT**
  https://www-01.ibm.com/software/network/commserver/SMC-AT/
- **RFC7609 (SMC-R)**
  https://tools.ietf.org/html/rfc7609
- **Linux on Z (technical):**
  https://www.ibm.com/developerworks/linux/linux390/
- **SMC for Linux on Z:**
  http://linux-on-z.blogspot.com/p/smc-for-linux-on-ibm-z.html
- **Blogs**
  - **Linux on z distributions new**
    http://linuxmain.blogspot.com/
  - **Linux on Z latest development news**
    http://linux-on-z.blogspot.com/
  - **KVM on Z**
    http://kvmonz.blogspot.com/
  - **Containers on Z, primarily Docker**
    http://containerz.blogspot.com/