The revival of the learning-sync bridgeport flag

HiperSockets Converged Interface

Alexandra Winter wintera@linux.ibm.com
Maintainer of linux/drivers/s390/net
## Trademarks

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

<table>
<thead>
<tr>
<th>Trademark</th>
<th>Company/TM</th>
<th>Product/Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>AX*</td>
<td>IBM*</td>
<td>PowerVM</td>
</tr>
<tr>
<td>BladeCenter*</td>
<td>IBM eServer</td>
<td>z/VM*</td>
</tr>
<tr>
<td>DataPower*</td>
<td>IBM (logo)*</td>
<td>z9*</td>
</tr>
<tr>
<td>DB*</td>
<td>InfiniBand*</td>
<td>z10 BC</td>
</tr>
<tr>
<td>FICON*</td>
<td>Parallel Sysplex*</td>
<td>z10 EC</td>
</tr>
<tr>
<td>GDSN*</td>
<td>POWER*</td>
<td>zEnterprise</td>
</tr>
<tr>
<td>HiperSockets*</td>
<td>POWER7*</td>
<td></td>
</tr>
</tbody>
</table>

* 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, at both and is used under license from there.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.

Windows Server and the Windows logo are trademarks of Microsoft Corporation in the United States and other countries.

InfiniBand is a trademark and service mark of the InfiniBand Trade Association.

Intel, Intel logo, Intel Inside logo, Intel Inside logo, Intel Centriente, Intel Centriente 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.

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

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

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.

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

* All other products may be trademarks or registered trademarks of their respective 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 multitasking in the user’s core, the configuration, and the workload processed. Therefore, no assurances can be given that an individual user will achieve throughput improvements equivalent to the performance ratio stated.

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 performance 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 availability of products 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 geographic area.
IBM zSystems aka Mainframes

Logical Partitions (LPARs)
HiperSockets

Provided by Hardware / Firmware (memory to memory moves)
Low latency / high throughput 😊
But:
Additional subnets
Routing rules, etc.

😊 “I want single-homed systems that can be deployed anywhere”
“use a bridge”

😊:
- Performance bottleneck
- Extra hop
- Single point of failure
HiperSockets Converged Interface - HSCI

- Single interface per system
- Choose HiperSockets if possible
- Chose default NIC otherwise

😊👍
Support virtualization

- Multiple target MACs on top of HSCI
- Dynamic add/del of instances
- (Live) migration
To consider:

• No (broadcast) loops

• Need a forwarding database (FDB) with (learned) source addresses and (learned) target addresses

What is reachable via HiperSockets?

• Chicken-egg problem with MAC learning on HiperSockets interface (risk that it is never used)

• Don’t rely on gratuitous ARP messages

• Ageing of obsolete entries

HiperSockets Firmware provides for:

• Query FDB of network segment

• Events in case of FDB changes
Options for implementation

• Hardware / Firmware
  -> additional buffer copy

• BPF
• ebtables
• Open flow
  -> working prototypes, but all required FDB implementation in user space

• Bridge and switchdev
  -> almost all required features exist already
Configure linux bridge as HSCI

- No loops:
  - stp off
  - ext dev: isolated on
  - hs dev: isolated on
  - hsci: isolated off

- ext dev as default:
  - ext dev: flood on
  - hs dev: flood off
  - hsci: flood on

- hs dev if possible
  - fdb entries
  - ext dev: learning off
  - hs dev: learning off
  - hsci: learning on
Switchdev

- Documentation/networking/switchdev.rst

- Device to bridge notifiers:
  - SWITCHDEV_FDB_ADD_TO_BRIDGE
  - SWITCHDEV_FDB_DEL_TO_BRIDGE

- Bridge to device notifiers:
  - SWITCHDEV_FDB_ADD_TO_DEVICE
  - SWITCHDEV_FDB_DEL_TO_DEVICE
Notifiers

Example:
- Add guest 2
- hsci port on left bridge learns source MAC
- Notification to HS channel
- Notification to right bridge: guest 2 is reachable via HiperSockets
How to turn notification on and off?

• Need to preserve legacy behaviour of HiperSockets interfaces (default)

• `learning_sync` attribute enables syncing of the learned/forgotten FDB entry to the bridge's FDB.

• `man bridge link set`:
  
  `learning_sync on` or `learning_sync off`

  Controls whether a given port will sync MAC addresses learned on device port to bridge FDB.

• `=> bridge link set dev $hsdev learning_sync on self`

• Controls subscription to and generation of notifications by HiperSockets interfaces

• No need to change bridge code (generation and subscription is always on for bridgeports)
Summary

HiperSockets Converged Interface:

• Efficiently converged the external network and an internal preferred network segment
• Used existing bridge and switchdev behaviour
• Additions to HiperSockets device driver:
  10a6cfc0fc82 s390/qeth: Translate address events into switchdev notifiers
  817741a8ea2 s390/qeth: Reset address notification in case of buffer overflow
  780b6e7db25e s390/qeth: implementndo_bridge_getlink for learning_sync
  521c65b64916 s390/qeth: implementndo_bridge_setlink for learning_sync
  60bb1089467d s390/qeth: Register switchdev event handler
  4e20e73e631a s390/qeth: Switchdev event handler
  f7936b7b2663 s390/qeth: Update MACs of LEARNING_SYNC device

• Addition to bridge code:
  d05e8e68b07c bridge: Add SWITCHDEV_FDB_FLUSH_TO_BRIDGE notifier
Open issues

• Bridge over bond:

Unlike MACVLAN, interfaces on bridgeports do not get notified in case of bond failover. So the attached guests do not send GratArps.

See

https://lore.kernel.org/netdev/20220329114052.237572-1-wintera@linux.ibm.com/

• HiperSockets support very large MTUs. How can HSCI benefit?

Bridge needs to settle on lowest MTU of all bridgeports. Investigate whether Segmentation Offload can help.