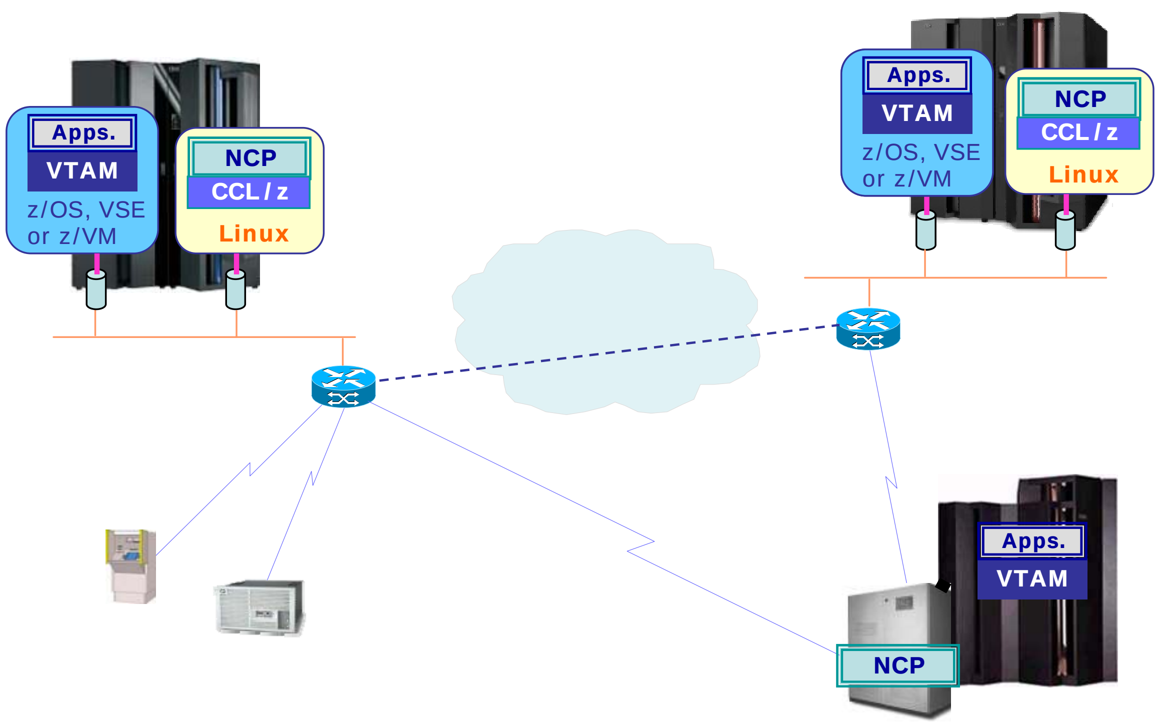
A WHITE PAPER

IBM COMMUNICATION CONTROLLER FOR LINUX ON SYSTEM Z V1.2.1

Extending the IBM 3745/46 Legacy for yet Another Decade



A MAINFRAME LINUX VIRTUAL 37XX FOR ACF/NCP & SNA MODERNIZATION

BY ANURA ['SNA'] GURUGÉ

SEPTEMBER 2006

Table of Contents

OVERVIEW 3

THE EVOLUTION OF THE CCL/Z 6

WHAT IS NEW IN CCL/Z V1R2 & V1.2.1 7

CCL/Z V1.2.1: WHAT IS POSSIBLE, WHAT IS NOT 14

JUSTIFYING THE CCL/Z AS A 3745 REPLACEMENT 16

CCL/Z: THE ARCHITECTURE 18

VM OR NOT TO VM – THE CCL/Z DEPLOYMENT OPTIONS 20

WHO NEEDS THE CCL/Z 21

THE BOTTOM LINE 21

ACRONYMS & SELECTED GLOSSARY 22

ABOUT THE AUTHOR 24

A Note on Acronyms

Given all the products and technologies involved, it is impossible to discuss the CCL/z without using a plethora of acronyms – some obscure and others with alternate meanings. Given the volume of acronyms in question, to spell out each, within the text, the first time it is used, as is my normal wont, was not practical with this White Paper. So please consult the 2.5-page, “Acronyms & Selected Glossary”, starting on page 18 if you come across a term that you are not sure of – in the context it occurs, keeping in mind that terms like CDLC can have IBM/SNA specific connotations.

SNAP-SHOT OF WHAT IS NEW IS CCL/Z V1R2 & V1.2.1

V1.2.1 (May 2006):

1. Integrated DLSw support
2. 3746 TIC3 emulation (referred to as “Native LAN” support)
3. IP Transmission Group (IP-TG) usability extensions
4. Additional performance and efficiency improvements

V1R2 (November 2005):

1. Support for all available OSA(-E) Ethernet variants
2. NPSI non-SNA X.25 connectivity using an XOT interface on an external router as well as IBM’s *“X.25 Over TCP/IP for CCL for Linux”* software product
3. ‘OSA-for NCP’ (OSN)
4. IP Transmission Groups
5. Significant performance enhancements

IBM Communication Controller for Linux on System z V1.2.1

A Mainframe Linux Virtual 37xx for ACF/NCP & SNA Modernization

Extending the IBM 3745/46 Legacy for yet Another Decade

The IBM Communication Controller for Linux on System z (CCL/z) is a *software-based* replacement for the now discontinued IBM 3745/3746 communications controllers – that in addition includes value-added SNA 'modernization' capabilities such as integrated DLSw. It is implemented as a mainframe Linux application that can run on a mainframe Linux LPAR or a z/VM Linux server image – co-residing, if necessary, with other Linux applications. Despite the 'System z' reference in its product name, the CCL/z will also run on z9s as well as S/390 G5 and G6 series machines that have the appropriate versions of Red Hat or SUSE Linux.

The CCL/z emulates a **3745 Model 31A** with 16MB of memory. It is a run-time environment, on a mainframe, for **ACF/NCP V7R5**(+), as well as **NPSI** and **NRF**. That, indeed, is the inescapable *raison d'être* for the CCL/z. It enables SNA customers to maintain much of the unique, value-added networking functionality provided by ACF/NCP, such as **SNI**, NCP Boundary Functions, **XRF**, and **SSCP Takeover** etc. – independent of aging 37xx hardware. It provides NCP with a whole new lease of life, on a highly strategic, mission-critical, hardware platform.

SNA Subarea Networking Still Prevails

Thanks to the CCL/z, mainframe customers can continue using established and proven SNA protocols, especially in WAN, b2b and disaster recovery scenarios, without running the risk of a prolonged disruption due to a 37xx crash. With spares becoming scarce, 37xxs, are, alas, no longer invincible. Hence the need for the CCL/z.

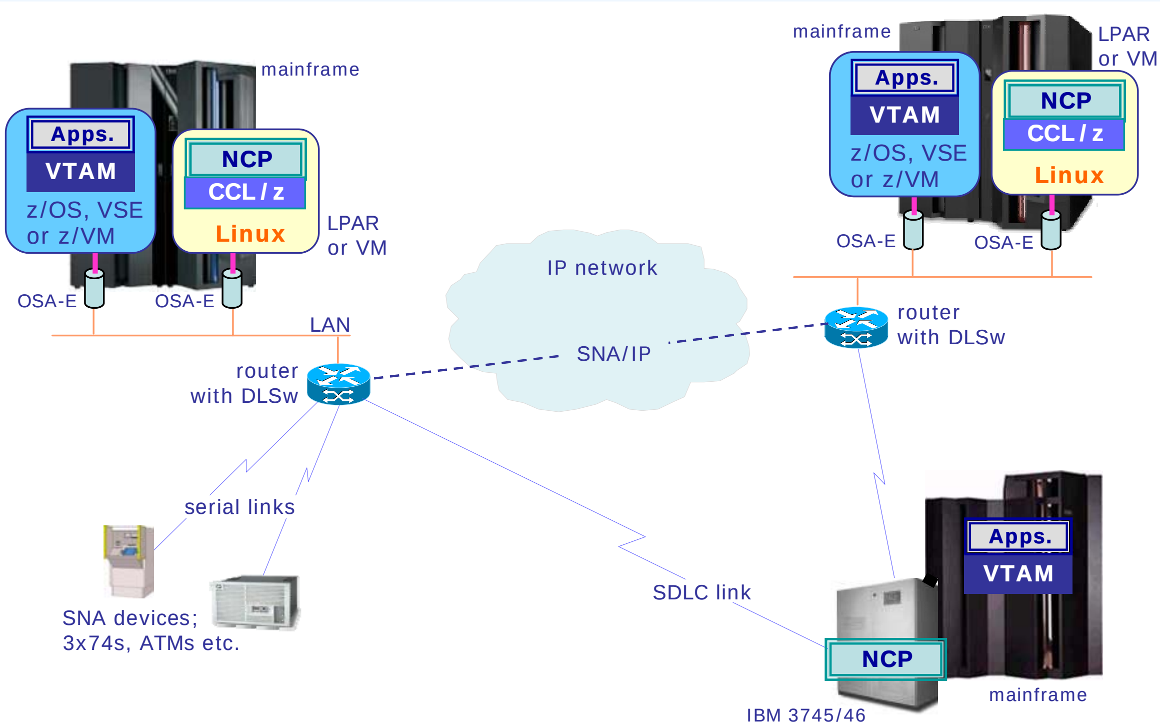
The indisputable popularity and prevalence of IP-centric enterprise networking, and the widespread use of SNA/IP integration technologies such as **tn3270(E)**, **Enterprise Extender** and **3270-to-HTML**, has, however, not resulted in the total demise of traditional SNA subarea networking. In mid-2005, there were still upwards of 20,000 37xxs in production, mission-critical use (though the 37xxs came to be over 30 years ago). True to what some of us have repeatedly said since the late 1980s, moving away from SNA is not a trivial endeavor – particularly if it requires coordinated changes across multiple, autonomous networks as is the case with SNI configurations. *NCP functionality will thus be required well into the next decade!* But going forward, NCP will be associated with CCL/z, rather than with 37xxs.

A Pragmatic Emulation

CCL/z does not require CCL/z-specific versions of 3745/46 software; e.g. NCP or **SSP**. There is also no need to create CCL/z-specific NCP definitions. Instead, ACF/NCP running on a CCL/z will, in general, accept and execute existing *NCP definitions* without the need for any major modifications. Any requisite changes are likely to be minor and easy to realize. This overall preservation of ACF/NCP's integrity also means that NCP running on a CCL/z will continue to work with key network management utilities such as **NetView**, **NPM** and **NTuneMON**.

The big difference between a CCL/z and a real 3745, as is to be expected, relates to physical I/O connectivity options – in particular serial ports (given that the ES/9370, c.1987, was the last IBM mainframe to offer serial ports for end-user networking). The CCL/z, consistent with IBM's strategic direction for mainframe networking, is OSA centric. It supports both Token-Ring and Ethernet OSA and OSA-Express ports – with CCL/z V1R2(+) being able to work with all the available Fast Ethernet, Gigabit and 10 Gigabit variants. V1R2(+), when implemented on a **z9-109** mainframe, will in addition provide an OSA-based, simulated **CDLC channel** connection between the LPARs or VMs running VTAM and the CCL – via a so called *'OSA for NCP'* (**OSN**) definition.

The absence of physical serial ports does not, however, preclude a CCL/z from accommodating all the widely used serial-mode SNA connections! CCL/z V1R2(+) supports SNA sessions traversing **SDLC**, **Frame Relay**, **TCP/IP [i.e. DLSw]**, **X.25** QLLC, **ISDN** and NPSI for non-SNA X.25 connectivity links – but does so in conjunction with the appropriate serial-to-LAN conversion technology [e.g. XOT] on contemporary 'multiprotocol' routers. Consequently, a CCL/z, in practice, should be able to handle at least 95% of all 37xx replacement needs.



A representative CCL/z configuration highlighting its SNI connectivity options and its support for serial links via DLSw-capable routers.

Mainframe Linux Makes Sense

Implementing CCL/z so that it runs on a Linux LPAR or VM, rather than being incongruous in anyway, is in reality, quite logical and rather convenient. The gravitation towards IP backbones and the availability DLSw (as well as APPN routing) on routers had, even by 1999, eliminated the need and justification for remote, link-attached 37xxs. Thus, the only real need was for a viable replacement for mainframe attached 37xxs.

Implementing a 3745 emulator outside of a mainframe would have meant the introduction of yet another sever platform into mainframe data centers – thereby increasing complexity and most likely overall operational costs. Any such platform would also have to measure up to all the stringent reliability, scalability and resilience criteria for mission-critical SNA networking. Plus this platform would have to have proven, high-performance Token-Ring and Ethernet adapters, à la OSA-E, with the appropriate support for SNA.

The more you think about it, a mainframe with OSA-E adapters is a great platform for 37xx emulation. It is already there within the data center, it meets all the mission-critical criteria, it offers various capacity on demand (**CoD**) options, and the SNA code for the OSA(-E) adapters is well proven. Implementing the emulator so that it runs on a Linux LPAR or VM, ensures that it can work with all the current mainframe OSs (including **TPF**) – and furthermore helps contain costs.

Mainframe Linux is targeted for cost-effective and convenient server consolidation, and the CCL/z sets out to exploit that. Rather than requiring a separate box, à la 3745, that takes up more floor space, the CCL/z becomes a shared resource, running on a mainframe alongside VTAM – albeit in a separate LPAR or VM. The ability to run Linux using the lower priced IFL processors, as an alternative to standard CPs, adds icing to the cake.

Little Time Left For Procrastination

The SNA community, ever cognizant of the exacting demands of mission-critical networking, is renowned for its circumspection when it comes to major infrastructure changes (with the dash to embrace SNA/IP being the one exception). But time, alack, is fast running out when it comes to the aging population of 3745/46s.

For those SNA customers that still rely on key ACF/NCP functionality, the CCL/z is the only viable strategic option – *going forward*. The CCL/z is a worthy replacement for the 37xx. It preserves the integrity of ACF/NCP – and thereby extends the life of the huge investment SNA customers have in ACF/NCP definitions and expertise.

The bottom line is that the CCL/z is a bit *less* than an actual 3745/46 – but at the same time also *more* than a 3745/46! Given that it doesn't support everything possible with a 3745/46 [e.g. BSC, NTO, ESCON] one has to admit to it being a subset of a real 37xx. But when it comes to performance and SNA modernization capabilities the CCL/z is definitely more than a 37xx. The CCL/z can significantly outperform a real 3745/46 while features like integrated DLSw and IP-TGs add a whole new dimension to SNA modernization possibilities.

THE EVOLUTION OF THE CCL/z

The table below sets out to provide a quick, easy-to-follow reference of how the capabilities and the performance of the CCL/z has evolved, somewhat rapidly, since the availability of initial release [i.e. V1R1] in March 2005. Where applicable the page number at which a description of a particular feature or enhancement can be found is included within "square braces" ([*'pp'*]).

CCL/z V1.2.1 — May 2006

New functionality:

1. Integrated **DLSw** support (in effect mainframe DLSw at long last) thus permitting SNA/APPN DLSw connections to be terminated within the CCL/z. [page 11]
2. **3746 TIC3 emulation** (referred to by IBM as "**Native LAN"** support), with the Token-Ring LLC2 protocol being processed using native System z instructions as opposed to being handled via the "NCP" – thus greatly improving performance and reducing mainframe CPU usage. [page 12]
3. **IP Transmission Group (IP-TG) usability extensions** to permit 'on-the-fly' modifications to configuration definitions and enable customers to specify which physical connections are to be used by specific IP-TGs. [page 12]

Ongoing **incremental performance improvements**, on top of those made in V1R2.
The overall performance, transactional throughput and response times are now *demonstrably better* than that of an actual 3745/46! At the same time the amount of System z CPU cycles required by the CCL/z continues to be minimized. [page 9]

CCL/z V1R2 — November 2005

Key new functionality:

1. Support for all available **OSA(-E) Ethernet** variants, irrespective of media-type and speed – with the added bonus of automatic **QDIO** support for all SNA LLC I/O performed across any OSA-E Ethernet port on z9s, z990s and z890s. [page 7]
2. NPSI non-SNA X.25 connectivity albeit using an **XOT** interface on an external router as well as IBM's *"X.25 Over TCP/IP for CCL for Linux"* software product that is sold separately from the CCL (since it is only required by XOT users). [page 8]
3. **'OSA-for NCP'** (OSN): Totally software-based, virtual channels within a z9-109 to enable VTAM-to-NCP and TPF-to-NCP communications across LPARs or VMs. [page 7]
4. **IP Transmission Groups (IP-TG)**: A new, optimized SNA-over-TCP/IP encapsulation scheme for high-throughput SNI or INN traffic between V1R2(+) CCL/z 'nodes'. [page 12]

Significant **performance enhancements** by rewriting the most frequently executed CCL emulation code in zSeries assembler and the introduction of IP-TGs.

CCL/z V1R1+ (& APAR #LI70826) — August 2005

First batch of **performance improvements** to the CCL/z emulation code.

CCL/z V1R1 [Initial Release] — March 2005

Key functionality:

1. Support for ACF/NCP V7R5 and above -- with boundary functions, INN & BNN
2. Support for NRF V1R9 and above
3. SNI support
4. XRF capability
5. Token-Ring TIC2 support via copper-based Token-Ring OSA(-E) ports

WHAT IS NEW IN CCL/Z V1R2 & V1.2.1

CCL/z was first previewed by IBM in May 2004 when the **Communications Server for Linux** was being announced. CCL/z V1R1 was formally unveiled in February 2005 and was ready for general availability in March 2005. CCL/z V1R1, which included support for SNI, boundary functions, INN, XRF, NRF, SSCP takeover etc., as well as support via external routers for SDLC, Frame Relay, X.25 etc., was, indubitably, feature-rich. It clearly demonstrated the viability and promise of this software-based 3745 emulator. But there were some distracting restrictions such as the OSA(-E) Ethernet support being restricted to copper-based connectivity options. V1R2 did away with nearly all of these restrictions.

V1R2

CCL/z V1R2 was available as of November 2005. V1R2, for a start, did away with many of the prior restrictions related to OSA(-E) Ethernet support. It also:

- Optimized CCL/z performance and adds **IP TGs** for faster inter-CCL/z interactions,
- Added NPSI non-SNA X.25 connectivity via an **XOT** interface on an external router plus IBM's "X.25 Over TCP/IP for CCL for Linux" software product, and
- Permitted internal VTAM-to-NCP interactions across a simulated channel on z9 mainframes.

V1R2, in addition to supporting OSA Token-Ring connectivity, embraced all available OSA-(E) Ethernet variants, irrespective of media-type or speed. **QDIO** mode I/O for SNA traffic across OSA-E Ethernet was also possible in some scenarios as described below. *However, fibre OSA-E ports cannot, even with V1.2.1, be used for VTAM to NCP-on-CCL/z interactions*. That is only possible with a copper LAN or OSN.

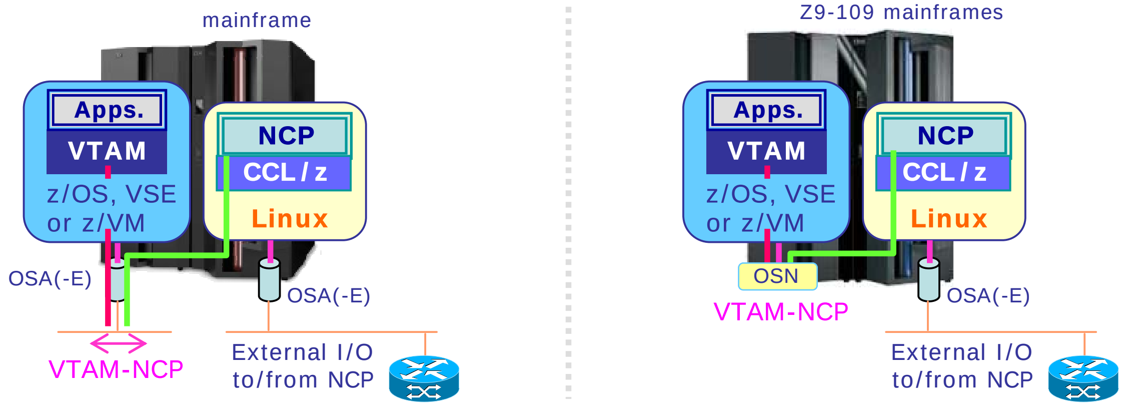
V1R2's universal Ethernet and QDIO support is made possible by what IBM refers to as its *"OSA Layer 2 support"*. Layer 2 support permits OSA-E Ethernet adapters on z9s, z990s and z890s *running Linux*, working in QDIO mode, to forward data frames to their destination using the frame's Ethernet MAC address – rather than the layer 3 addresses [e.g. IP address].

z9-109 customers, in addition, have the option of accelerating throughput and simplifying external LAN cabling through the use of the so called OSN definitions [page 4]. OSN definitions, as shown on page 8, create a totally software-based, virtual channel within a z9 to enable VTAM-to-NCP or TPF-to-NCP communications across LPARs or VMs. It is a kind of CCL/z V1R2 (on z9) specific HiperSockets mechanism for SNA traffic, given that HiperSockets can only be used with IP traffic.

V1.2.1

CCL/z V1.2.1 was available as of May 2006. It built upon V1R2, further enhanced performance and added the following 3 capabilities:

- integrated DLSw: This fulfills the well greeted statement-of-direction contained in the V1R2 announcement. It is also a long hoped for landmark in SNA modernization. Now, nearly 15 years after DLSw was famously pioneered by IBM on its star-crossed 6611 bridge/routers, one can have end-to-end DLSw between a mainframe end-point and a remote LAN.



STANDARD CCL/Z OSA REQUIREMENTS VS. THE NEW OSN FEATURE AVAILABLE WITH Z9-109S.

- TIC3 support to greatly enhance Token-Ring networking performance while reducing CPU utilization.
- Enhancements to the IP-TG feature introduced with V1R2.

With the Ethernet support and OSN having already been discussed above, the remainder of this section will focus on the other features introduced in V1R2 and V1.2.1 – including the performance enhancements.

NPSI NON-SNA X.25 VIA XOT / OEM MODULE

Though X.25, inexplicably, was never big in North America, it continues to be widely used in Europe and Japan. NPSI, which was introduced way back in the early 1980s, was IBM's strategic means for end-to-end SNA transport over X.25. The subsequent NPSI for non-SNA devices capability, raised the ante vis-à-vis X.25, even further. It performed, on-the-fly 'VT-to-SNA' protocol conversion, within a 37xx, to enable devices with no SNA cognizance [e.g. asynchronous mode terminals] to still have ready access to mainframe SNA applications.

NPSI running on a CCL/z V1R2(+) retains all of its X.25 capabilities, including the protocol conversion for non-SNA devices. As with the other serial-mode protocols, an appropriate router solution, this time with *'X.25 over TCP/IP'* (XOT) technology (rather than DLSw), will need to be used in conjunction with the CCL/z and the NPSI software. In the context of SNA and NPSI, XOT can be considered to be a 2nd-tier encapsulation. Whereas, DLSw is SNA-over-TCP/IP, XOT is SNA-over-X.25-over-TCP/IP!

So the X.25-over-TCP/IP has to be dealt with [i.e. terminated] in order to get to the SNA. This final XOT termination cannot be performed by a LAN-attached router as it will then necessitate the transports of X.25, i.e. a serial protocol, across OSA. Thus there is no option but to terminate XOT inside the mainframe – with a separate application running on the Linux LPAR or VM, alongside CCL/z; i.e. IBM's ***"X.25 Over TCP / IP for CCL for Linux"*** software product that is *sold separately* from the CCL/z (since it is only required by XOT users).

PERFORMANCE ENHANCEMENTS INCLUDING IP-TGS & TIC3 SUPPORT

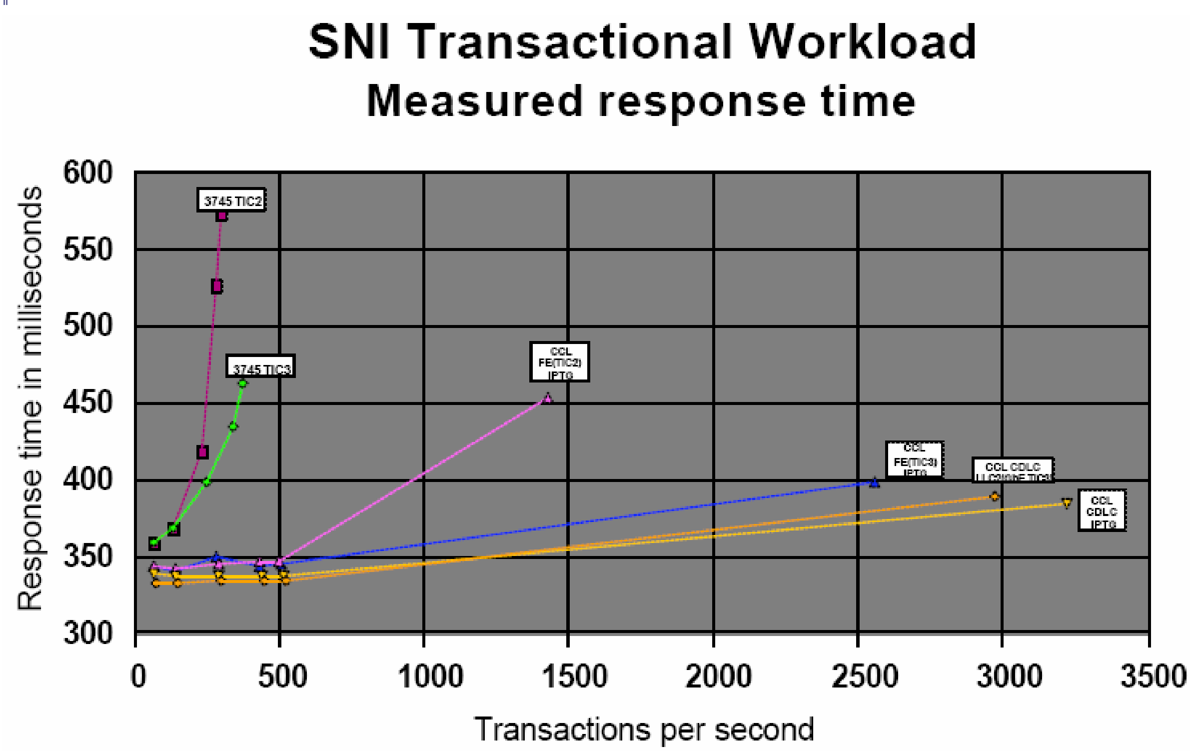
As of V1R2 CCL/z easily outperforms actual 3745/46 configurations in all networking scenarios – whether it be SNI, INN or boundary function processing! Furthermore, in marked contrast to what happens on real 37xxs there is no sharp degradation in response times as transaction loads increase.

Consequently a CCL/z V1.2.1 (with TIC3 support and IP-TGs) on a single z9 IFL engine can adroitly handle the workloads of 3 to 5 actual 3745-31As (running at 70% utilization) without in anyway compromising end-to-end networking throughput or performance. There has also been a marked reduction as of V1R2 in the amount of mainframe CPU cycles required by a CCL/z to perform a given 3745 workload. This mitigates concerns that CCL/z processing will require undue amounts of CPU cycles.

The performance gains and the processing efficiency of today's CCL/z can be attributed to 3 key factors:

1. Finely tuned code with the most frequently executed code segments now written in zSeries assembler,
2. IP-TGs that significantly speed up inter-CCL/z interactions while using less CPU resources, and
3. TIC3 emulation whereby Token-Ring related I/O is processed using native System z instructions as opposed to being processed via the "NCP".

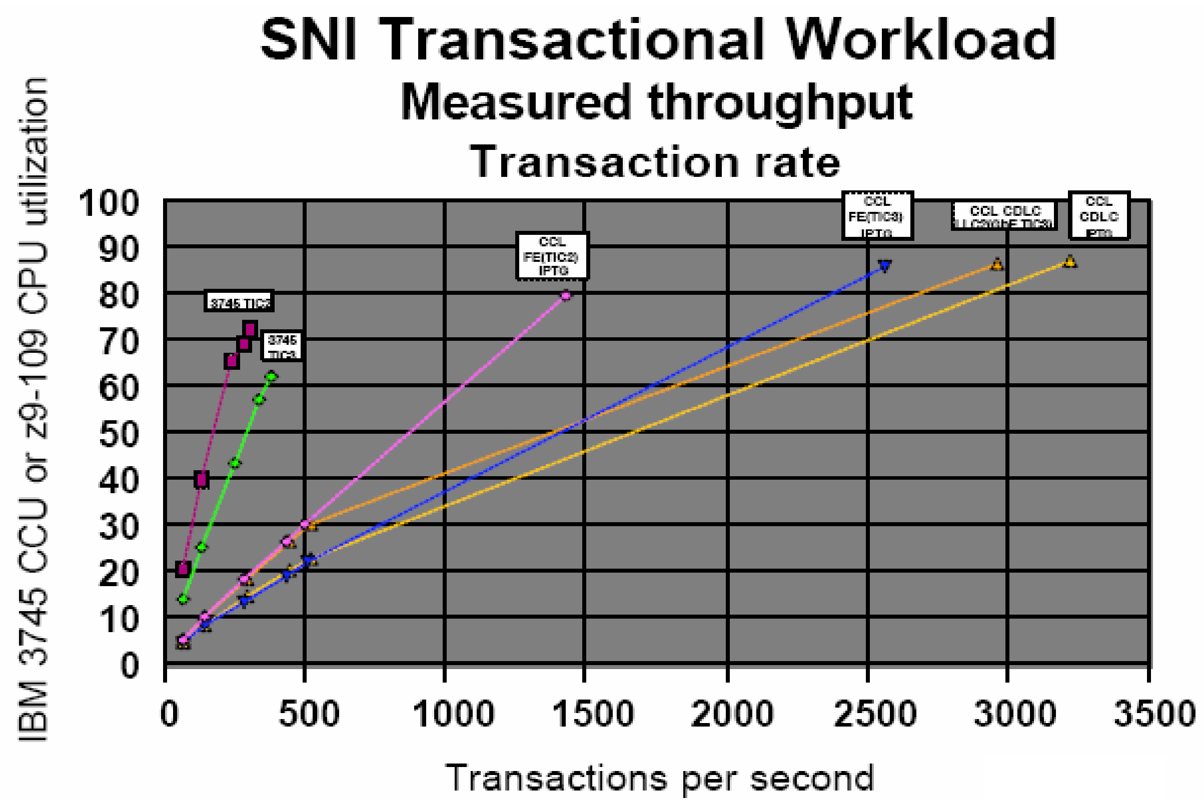
The tangible results of these performance enhancements and CPU utilization optimization can be seen in the following two IBM supplied graphs. The 1st graph shown below compares SNI response times for different transaction rates on actual 3745-31A TIC2/TIC3 configurations versus those achieved with different CCL/z V1.2.1 TIC2/TIC3 and IP-TG permutations.



The two leftmost, nearly vertical lines are that for the 3745 – with the leftmost for a TIC2 setup and the other with a TIC3. What this shows is that SNI response times on 3745-31A configurations increase extremely rapidly as the load reaches 250 to 300 transactions/sec. The four bottom lines show CCL/z performance with the topmost being that for the TIC2 IP-TG configuration.

This indicates is that even the least efficient of the CCL/z V1.2.1 configurations will still greatly outperform a real 3745 – the key point being that unlike with a 3745 there is no degradation in response times as the transaction load increases. The possibilities of 3745 SNI workload consolidation is thus attested to by the three bottom lines that demonstrate that a CCL/z IP-TG configuration can handle 10 to 12 times more transactions/sec. and do so without impacting response times.

The second graph, shown below, is again for an SNI setup but this time looking at 3745-31A CCU utilization versus that of z9-109 CPU utilization.



Again the two leftmost, nearly vertical lines are that for the 3745 – this time showing the rapid rise in CCU utilization as transaction rates increase. The other 4 lines are for CCL/z CPU utilization with the leftmost line, yet again, representing the TIC2 configuration. The bottom line here is that these actual measured results substantiate IBM's claims that the workloads of 3 to 5 actual 3745-31As can be easily handled by a single CCL/z without any degradation in response times or throughput.

INTEGRATED DLSW

DLSw, thought it unlike EE does not support SNA routing or LU 6.2 traffic prioritization, is by far the most widely used SNA-over-TCP/IP transport scheme. Given that prior to V1.2.1, DLSw was only available on routers (with 'Desktop DLSw' having being but a short-lived 'blip'), it is easy to forget that DLSw was a technology pioneered by IBM in 1982. IBM was also extremely instrumental in making DLSw into an industry standard via RFCs 1434, 1795 and 2166. So IBM is no stranger to DLSw.

It, however, also has to be noted that IBM in the mid-1990s abandoned DLSw in favor of EE. DLSw by then had been totally appropriated by the bridge/router vendors and with EE IBM was hoping to shift the goal posts so that it could get back into the game. Thus EE on mainframes became a reality with OS/390 Ver. 2 Rel. 6 way back in September 1998.

Having to terminate DLSw outside the mainframe became even more frustrating with the introduction of QDIO – given that it only supported IP traffic. Some customers even went as far to use a two-hop, hybrid DLSw-EE solution to get around this limitation; i.e. terminating DLSw at a router and then using EE to get in and out of the mainframe. Now at long last mainframe DLSw is finally here – albeit tacitly targeted at customers who are using DLSw in front of 37xxs (as opposed to those that have already migrated to a VTAM/OSA(-E) scheme).

As with implementing the 37xx replacement on Linux there is some justifiable logic in finally having mainframe DLSw within CCL/z – rather than say Comm. Sever (though DLSw on mainframe Comm. Server would, nonetheless, make a lot IT folks extremely happy). CCL/z sets out to support traditional subarea SNA, in situ, without the need for APPN/HPR. This also applies to DLSw – which in reality is a LLC2-based encapsulation methodology. Unlike EE, DLSw does not insist on APPN/HPR mode operation. It works with traditional SNA. Furthermore, EE is only supported by VTAM running on z/OS or OS/390, while CCL/z also works with VTAMs running on VSE, VM or TPF.

V1.2.1's integrated DLSw will, at a stroke, make CCL/z-based networking more efficient, resilient and scalable – not to mention simpler. For a start, DLSw-encapsulated subarea SNA traffic can exploit QDIO on any mainframe that supports QDIO. In contrast, the OSA-E Layer 2 support that also enables SNA in QDIO mode is only available with Ethernet, on the newer mainframes running Linux. There is also support for virtual SNA MAC addresses to simplify configuration needs.

With integrated DLSw, 37xx customers that rely on a large number of data center routers for DLSw termination can now replace their aging 37xxs and consolidate their router needs at the same time. This, at a minimum, holds the potential for reducing maintenance contract costs while at the same time bolstering overall networking resilience. The bottom line here is that integrated DLSw is indeed a significant milestone in SNA modernization.

IP Transmission Groups (IP-TGs)

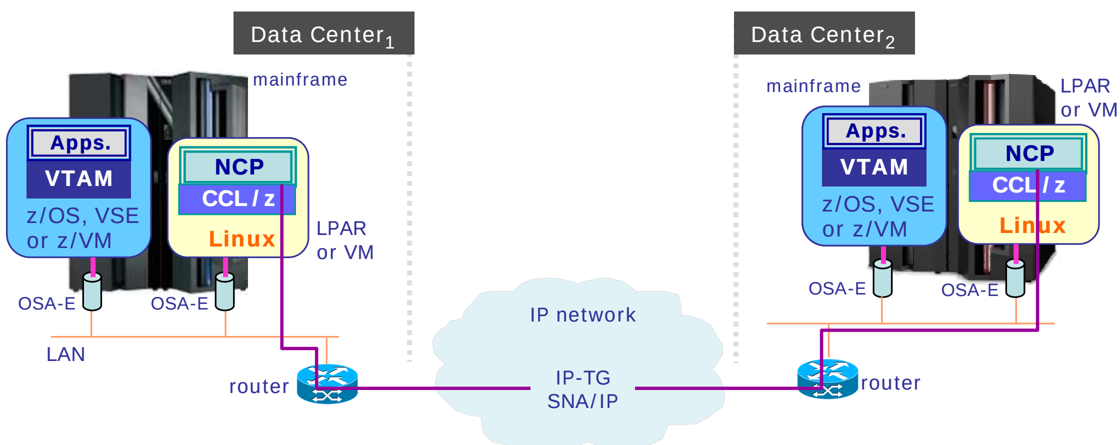
IP-TGs is a new, SNA-over-TCP/IP encapsulation scheme. At present it is only supported by, and as such can only be used between, CCL/z V1R2(+) nodes. It is recommended by IBM as the optimum SNA/IP transport mechanism for inter-data center connectivity, between CCL/zs for SNI or INN traffic.

IP-TG is meant to be an *optimized DLSw* for SNA – that is more efficient and is also able to properly prioritize traffic per SNA LU 6.2 COS (albeit contingent on the availability of appropriately configurable TCP ports or routers that support TOS). Introducing yet another SNA/IP encapsulation scheme, at this juncture, to optimize inter-data center SNA networking, is justifiable, though there will be those that, also justifiably, will decry the departure from the IBM endorsed DLSw and EE RFC 'standards'.

But then again as discussed above, IBM is no stranger to DLSw and if IBM says IP-TG is better than DLSw one should take note of it and not dismiss outright as just hype or sour-grapes. Furthermore the performance graphs shown earlier clearly demonstrate that when it comes to SNI, CCL/z with IP-TGs is fast and very efficient.

IP-TG, however, is an option, and customers can, as shown in the figure on page 4, indeed use DLSw for SNI or INN instead of IP-TG.

V1.2.1 includes usability enhancements to IP-TGs. These enhancements permit IP-TG configurations to be dynamically updated we well as providing customers with more specific controls over which physical connections are used by IP-TGs.



OPTION OF USING OPTIMIZED IP-TG FOR SNI OR INN, ACROSS AN IP NETWORK, BETWEEN CCL/Z V1R2 IN SEPARATE DATA CENTERS RATHER THAN DLSW.

3746 TIC3 Emulation (a.k.a. 'Native LAN')

The Toke-Ring interface adapter on 3745s is the TIC2 while that on 3746s is the TIC3. 3746 I/O adapters, such as the TIC3, benefit from getting additional processing help from 3746 resident 'Assist Processors'. These 3746 Assist Processors are used in particular to handle low-level, LLC processing. Thus with TIC3s, the NCP running on the 3745 frame did not have to contend with having to handle the Token-

Ring LLC. Instead this processing is offloaded to a 3746 Assist Processor via the 3745/3746 coupler.

Prior to V1.2.1 the CCL/z did not have explicit support for TIC3s. Instead, all NCP Token-Ring connections were supported as if they were TIC2s. V1.2.1 adds true TIC3 emulation replete with NCP offload as with a real 3745/3746. Now the NCP running in the CCL/z does not have to perform the LLC2 function. Instead TIC3 LLC2 processing is realized using native System z instructions. This significantly increases TIC3 related performance while at the same time reducing overall mainframe CPU utilization. The performance graphs shown earlier do indeed attest to this and demonstrate that TIC3-based networking is faster and more efficient than that possible with TIC2s.

TIC3 emulation is thus another major evolutionary step for the CCL/z.

CCL/z V1.2.1: What is Possible, What is Not

CCL/z, suffice to say, is a complex offering with considerable functionality, quite a few options and some important caveats. The table below attempts to capture and present as much of this *'yes, can do/no, can do'* aspects of the CCL/z V1.2.1 in a somewhat visual form. It is, however, recommended that you cross-check all items of interest with IBM to make sure that nothing is being overlooked.

	Supported	**Not Supported**
CCL implementation	on mainframe Linux, either in an LPAR or on a VM 'guest', in shared-mode if necessary – with z/OS, OS/390, VSE, z/VM or TPF on z9-109, z990, z900, z890, z800 or S/390 G5/G6	➢ on non-Linux mainframe OSs ➢ Linux on non-mainframe platforms
37xx emulation	3745 Model 31A (with 16MB) in particular – but in general 3705s, 3720s, 3725s, 3745s & 3745/3746-900s with one CCU [see below for more on CCU]	➢ 3746-950 ➢ 2216 MAE for the 3746 *[though some of the MAE functions, such as the tn3270(E) server, can be duplicated using an 'external' Comm. Server implementation.]* ➢ dual CCU 3745 Model 61A running two separate NCPs. *[ACF/NCP on a CCL only runs in single CCU mode.]*
3745 software and software features	1. ACF/NCP V7R5 and above [with boundary functions, INN & BNN] 2. NRF V1R9 and above 3. NPSI (with routers with XOT and a 3rd party mainframe Linux module) 4. SNI 5. XRF	➢ EP (and thus PEP) ➢ *NCP IP routing* ➢ NTO ➢ XI ➢ NSI ➢ NSF
3745 interfaces	1. **TIC2** & **TIC3** Token-Ring LANs via Token-Ring or (Giga/Fast) Ethernet OSA(-E) *with SNA transparently supported across any of the Ethernet options* 2. 'CDLC channel' via OSN on z9-109s (see page 4 & 6) 3. dual-TIC support – if necessary across Ethernet (using DLSw and VLANs to realize the simulation)	➢ ESCON or Bus-&-Tag channels ➢ serial ports *(other than via DLSw and XOT on external routers)*
CCL I/O interfaces	1. OSA(-E) Token-Ring 2. all OSA(-E) Ethernet variants, irrespective of speed or media type, with QDIO mode operation with Layer 2 support – with automated T-R emulation to permit SNA-over-Ethernet *[but copper or OSN is required for all VTAM interactions]* 3. OSN on z9-109s (see page 4 & 6)	➢ mainframe ESCON & FICON channels

	Supported	**Not Supported**
Serial connections *in conjunction with external routers (and 3rd party mainframe software in the case of NPSI)*	1. SDLC 2. X.25 QLLC & NPSI (non-SNA) 3. Frame Relay 4. ISDN	➢ BSC ➢ Start/Stop [i.e. asynchronous]
3745 Model 61A Dual CCU modes	The CCL/z is a single CCU emulation, but certain dual CCU mode configurations may be supported by running two CCL/zs; e.g. single mode [i.e. separate NCP on each CCU], standby mode and backup mode.	
SNA sessions	session types 0, 1, 2, 3, 6(.2) by any of the LU types resident in SNA/APPN node types 1, 2, 2.1, EN, NN, 4 or 5.	➢ now obsolete session type 4
IBM mainframe resident 37xx support software	1. ACF/SSP on z/OS, OS/390, VSE and z/VM	
Mainframe network management programs	1. IBM NetView 2. IBM NTuneMON 3. IBM NPM with some exceptions 4. NCP dump & restart (which now work faster than ever) 5. Tivoli OMEGAMON *family* *6. Most likely 3rd party programs that worked with ACF/NCP V7R5(+)*	
VTAM	VTAM V4R2 (or higher) running either on z/OS, OS/390, VSE or z/VM within the same CEC as the CCL/z, or on another, OSA(-E) *LAN attached*, mainframe	
Mainframe Linux	**31- or 64-bit:** 1. SUSE LINUX Enterprise Server for IBM zSeries and IBM S/390: 8 or 9 [i.e. SLES8 or SLES9] 2. Red Hat Enterprise Linux AS (RHEL4)	
z9-109 and zSeries PUs	1. CPs 2. IFLs *in shared or dedicated mode*	
Multiple CCL/zs per machine	Yes, with really no limitations, with CCL/zs being able to share CPs or IFLs – and even the possibility of running multiple CCL/zs on the same Linux LPAR or VM	
SNA-over-IP encapsulation schemes	1. IP-TG *(with potential SNA COS prioritization with TOS etc.)* 2. DLSw *3. EE – only in conjunction with Comm. Server for Linux*	➢ Multi-link TGs (MLTGs) with either IP-TG or DLSw

JUSTIFYING THE CCL/Z AS A 3745 REPLACEMENT

0. Only viable replacement option for the discontinued IBM 3745/3746 communications controllers for SNA customers that are still heavily dependent on 37xx-centric, SNA networking functions for their mission-critical business needs – especially for *high-performance*, inter-network interactions with other enterprises.

1. Realistic support for ACF/NCP V7R5 (and above), SSP, NRF and now NPSI – which include all of the really important (and unique) NCP functionality such as boundary functions, SNI, XRF, SSCP take-over and INN, and designed to work with z/OS, OS/390, VSE, z/VM and TPF.

2. Ability to implement it on mainframe Linux, on a LPAR or VM guest, obviates the need to introduce another box into the data center – simplifying installation, management and operations; saving valuable data center real estate, and eliminating concerns about RAS or scalability.

3. Support for all of the OSA(-E) Giga/Fast Ethernet options, with copper or fibre media, with automatic, transparent Token-Ring emulation, finally enables native SNA traffic to enjoy the throughput and efficiency of QDIO-mode, OSA-E data transfers.

4. Accepts and works with existing ACF/NCP definitions and SSP libraries, with minimum changes required to the NCP definitions, thereby extending the huge investment enterprises have in NCP software and expertise.

5. Does not require traditional SNA subarea networks to be updated to be APPN/HPR compliant and V1.2.1 even offers integrated DLSw.

6. Facilitates implementation of geographically dispersed DR configurations without concerns that aging 37xxs could be a potential liability – and furthermore enjoy the **GDPS** support for the automated recovery of Linux applications.

7. Unrestricted, transparent interoperability with ACF/NCPs running on real 3745/3746s in SNI or INN scenarios.

8. Uncompromised support for XRF, with continued support for existing network configurations, at a time when all enterprises are under increasing pressure to have in place proven, hot-standby disaster recovery (DR) schemes.

9. Enables the continued exploitation of serial link attached mission-critical, legacy devices [e.g. ATMs, 4700 ilk financial controllers and POSs], that use SDLC, X.25 or Frame Relay – albeit in conjunction with external routers in some cases.

A. Possibility of implementing multiple copies of CCL/z on a single mainframe, for workload partitioning or redundancy – with the CCL/z pricing, which is based on mainframe processors associated with running CCL/zs, as opposed to the number of CCL/z implemented, making this option somewhat attractive [though you will still have to added costs of paying for the multiple ACF/NCP licenses and the additional mainframe resources consumed].

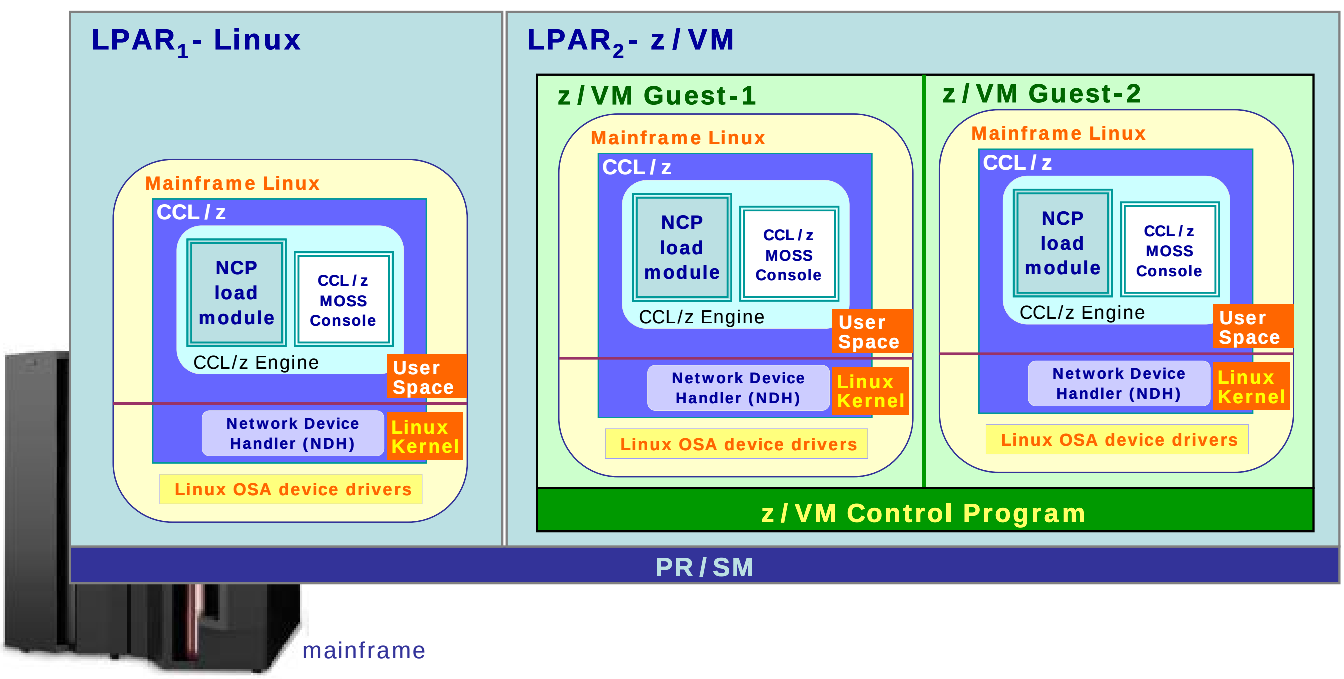
B. Retains compatibility with all of IBM's key 37xx related network management applications [e.g. NTuneMon], as well as providing a (Web browser accessed) 37xx MOSS console emulation, thereby minimizing retraining needs and preserving hard earned operational skills – thus making the transition from a physical 37xx to a CCL/z that much easier for operational and support staff.

C. CCL/z does not require IBM's Communications Server for Linux (CSL) as a prerequisite in order to realize any of the SNA, SNA/IP or LAN functionality it provides (though customers who wish to, do have the option of running both CCL/z and CSL on the same Linux image on a LPAR or VM).

D. A CCL/z can be run using either z9-109/zSeries CPs or IFLs, without the need for any dedicated processors – with the added bonus that IFLs are less expensive than CPs.

E. Possibility of updating and transferring existing ACF/NCP licenses to a CCL/z from a 37xx *being displaced* – particularly when coordinated with the 2 month testing allowance available on new NCP licenses.

F. Eliminate the now real danger of unacceptable, and potentially very costly, service disruptions caused by 37xx hardware failures – especially with spares and seasoned 37xx technicians getting scarce.



CCL/z DEPLOYMENT POSSIBILITIES – HIGHLIGHTING THAT A CCL/Z CAN BE IMPLEMENTED EITHER ON A NATIVE LINUX LPAR OR ON A VM GUEST WITH Z/VM. IT IS ALSO POSSIBLE, AS SHOWN ON PAGE 16, TO IMPLEMENT MULTIPLE CCL/ZS ON A SINGLE MAINFRAME LINUX IMAGE ON EITHER A SINGLE LINUX LPAR OR VM.

CCL / z: The Architecture

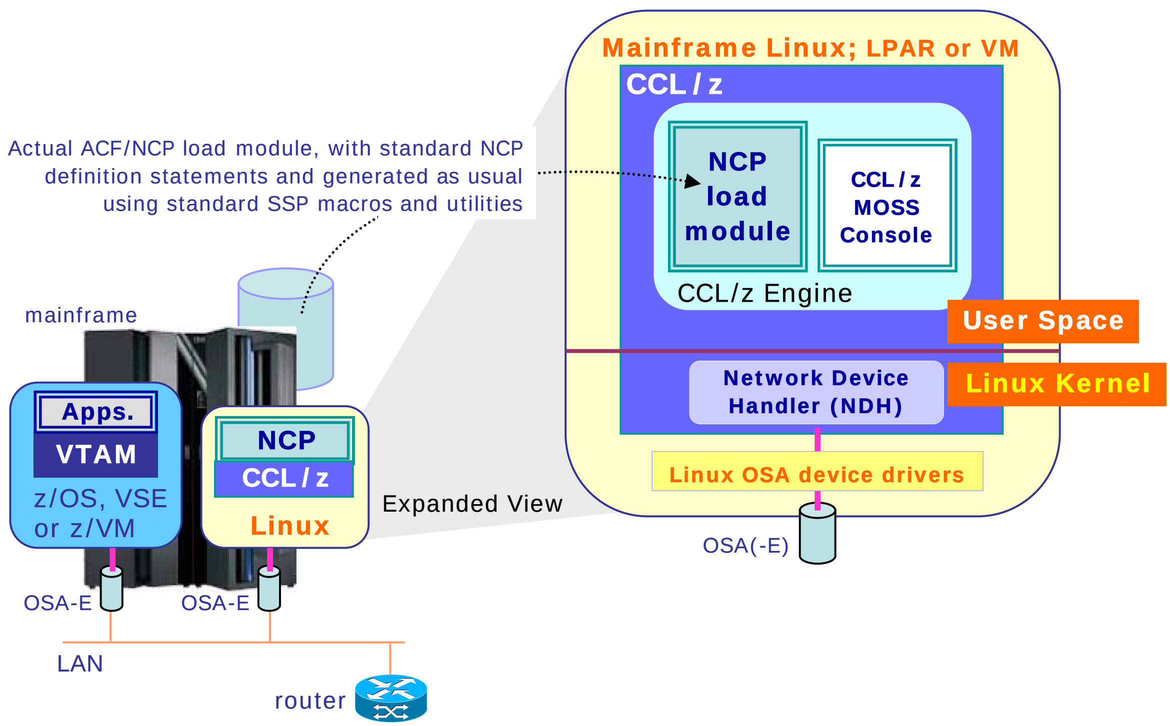
The CCL/z is made up of software functionality written in a combination of C and zSeries Assembler. This software was designed, expressly, to run on mainframe Linux, either on a Linux LPAR or a Linux image on a z/VM guest machine. The CCL/z functionality, as shown in the diagram below, consists of two primary components:

1. CCL/z **Engine**, and
2. **Network Device Handler (NDH)**.

The CCL/z functionality spans both the Linux user space and the kernel space – irrespective of whether it is running on a LPAR or VM. The CCL/z engine always runs in the user space, while the NDH, given the 'hooks' it needs into the Linux OSA device drivers, can only run within the kernel space.

The CCL/z engine is what emulates the core hardware functionality of a 3745; a 3745 Model 31A with 16MB of memory to be precise (where the Model 31A was a single CCU controller). It includes a MOSS console emulation that can be accessed via a standard Web browser. This MOSS console enables system operators to perform many of the familiar 37xx operational functions previously executed using the actual MOSS console of a 3745.

It is the CCL/z engine that also executes the crucial ACF/NCP load module. Though existing ACF/NCP definitions created to run on 37xxs will have to be updated to reflect the changes necessitated by the CCL/z *OSA-centric* configuration, the key here is that a CCL/z executes what in reality is still a standard ACF/NCP load module.



A HIGH-LEVEL ARCHITECTURAL VIEW OF THE CCL/Z SHOWING ITS TWO MAIN COMPONENTS AND HOW THESE ARE SPREAD ACROSS BOTH THE LINUX USER SPACE AND KERNEL SPACE.

The NCP load module executed by the CCL/z engine is created, per the standard ACF/NCP conventions of the last 3 decades, using NCP definitions and the usual SSP macros and utilities. The NCP creation, with SSP, would typically be performed on the OS running VTAM (though this is more of a convention, SSP licensing and convenience issue, as opposed to a CCL/z prerequisite). So from an NCP generation standpoint, creating a load module for the CCL/z is no different than that used for an all-LAN 3745. There are some new definitions and conventions but the overall process is still the same. *This is indeed good news for Systems Programmers*.

The process for loading a newly created NCP module into a CCL/z, as is to be expected, is, however, somewhat different to that associated with physical 37xxs. A new NCP load module has to be first transferred, using **FTP**, from the non-Linux OS [e.g. z/OS] into the Linux image that will be running the target CCL/z. A Linux ***"cclengine"*** command is then used to start the subject CCL/z and have the new NCP load module loaded into it. Once the NCP has been loaded, VTAM can contact it and activate it as with a real 37xx – with the CCL/z appearing to VTAM either as a LAN-attached *'XCA Major Node'* with a non-OSN connection, or a channel-attached 3745 if OSN is used.

NETWORK DEVICE HANDLER (NDH)

NDH is a Linux kernel extension. It is dynamically loaded into the kernel space. *Thus the NDH does not require a kernel rebuild or reboot*.

The NDH is logically positioned between the Linux OSA device driver and one *or more* CCL/z engines running on that Linux image. It serves as a kind of 3745 TIC2 emulator between the **NCP Token-Ring Interface (NTRI)** code running in the CCL/z engine and the actual LCS-mode OSA(-E) port being used by a CCL/z for its external communications. The key here being that NCP, hitherto always running on a 37xx as opposed to a mainframe, has never dealt directly with OSA(-E) ports. NTRI is the only LAN I/O interface that is recognized by NCP. Thus, NDH acts as a 'translation layer' between NTRI and OSA(-E) device drivers.

It is the NDH that ensures that NCP on a CCL/z will work, transparently and interchangeably, with either OSA(-E) Token-Ring or (Giga/Fast) Ethernet *(even though 3745s never supported SNA over Ethernet)*. The NDH automatically performs all the necessary layer 2 transformations between Ethernet and Token-Ring frame formats, so that the NCP never sees any Ethernet frames even when all of its communications is across OSA(-E) Ethernet.

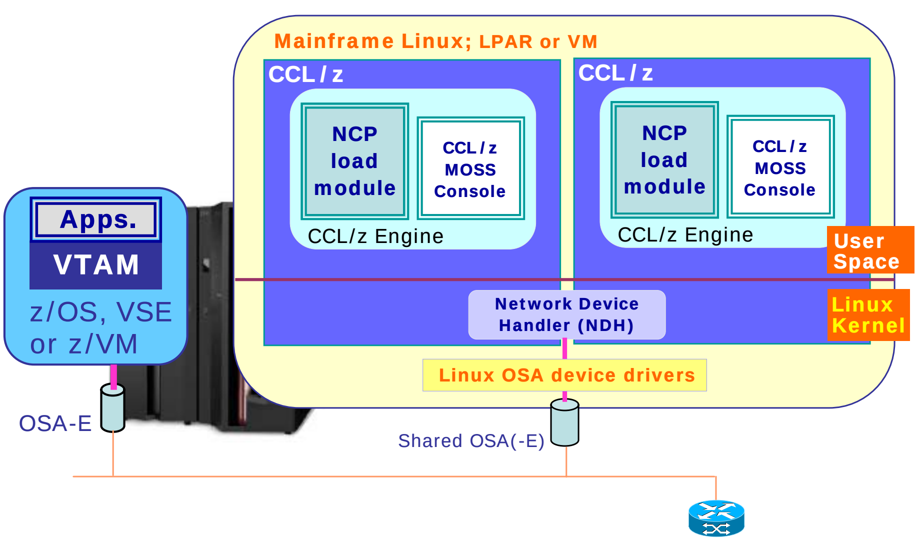
A copy of the NDH '*rpm'* is shipped with CCL/z but installed and compiled separately via the rpm install process. Only one instance of the NDH is required (and thus loaded) per Linux image – independent of the number of CCL/z engines that are to run on that Linux image. In other words, multiple CCL/z engines running on the same Linux image will always share a single instance of the NDH.

VM or Not to VM – The CCL / z Deployment Options

With mainframe Linux still somewhat nascent, there is, understandably, some uncertainty (and possibly even some confusion) as to how and where a CCL/z, or for that matter even multiple CCL/zs, should be implemented. Ironically, a part of the problem would appear to be the conspicuous lack of limitations when it comes to deployment options! IBM does not insist that CCL/zs can only be deployed in one particular way on one specific kind of Linux image. Hence customers have the ability to freely select the option that best meets their needs, expectations and budgetary goals.

The following criteria attempts to summarize the pertinent deployment guidelines that relate to CCL/z deployment possibilities:

- Independent of CCL/z: only one Linux image per (native) Linux LPAR, or multiple Linux images, as VM guests, on a z/VM LPAR.
- CCL/z can be deployed, *without any prejudice*, on either a Linux image running on a Linux LPAR, or on a Linux image running as a z/VM guest – *with the same CCL/z functionality in each case, though a CCL/z running on a Linux LPAR is likely to be about 10% faster than that implemented on a VM*.
- Multiple CCL/zs can be implemented on a single Linux image – independent of whether the Linux image is running on a Linux LPAR or VM guest.
- Linux workload, including CCL/zs, whether running on a Linux LPAR or a VM guest, can be powered by either shared or dedicated CPs or IFLs – with IFLs being less expensive than CPs.
- Each CCL/z, runs its own, dedicated NCP load module (and has its own MOSS console emulation) – independent of other NCPs, irrespective of whether CCL/z are co-located within the same Linux image or running on separate Linux images.
- OSA(-E) ports, *may be* shared between multiple CCL/zs – albeit with some potential (and recondite) limitations (that IBM can elaborate on when one is closer to thinking about exact hardware requirements for a specific CCL/z scenario).



Multiple CCL/zs can be implemented on a single mainframe Linux image – whether that image is running on a Linux LPAR or as a VM guest. Note that the NDH is shared between the CCL/z engines.

Given the above criteria, one should be able to quickly determine that enterprises can (and should) base their CCL/z deployment preferences contingent on their needs for running *other* mainframe Linux workloads. This is particularly germane vis-à-vis opting to pursue a VM Linux approach. *Unless you already have VM Linux, or intend to use VM Linux in the near future to run other Linux applications, there is no redeeming justification for implementing VM Linux just to run one or two CCL/zs*. Instead, it would be simpler and more cost effective just to run the CCL/zs in Linux LPARs.

WHO NEEDS THE CCL/z

The CCL/z is not an attempt by IBM to resurrect a new wave of SNA subarea-based enterprise networking. IBM, better than anybody else, knows that there is no merit or mileage in that. Instead, the CCL/z is *bona fide* attempt to truly help out SNA customers that are still dependent on aging 37xxs for some, or all, of their mission-critical networking. It enables them to maintain their networking status quo, without undue pain, and systematically migrate towards new networking schemes.

It eliminates the pressure to quickly find non-NCP based alternate schemes just to avoid being dependent on 37xxs. When dealing with changes to mission-critical networking, that involves multiple autonomous entities, as is inevitably the case in SNI scenarios, one has to have time on their side. *Haste is not an option*. Hence the need for the CCL/z.

It is highly unlikely that any customer that is not currently using a 37xx for production use will be interested in the CCL/z. It is not meant, in any shape or form, to be retrofit option for SNA customers that have already migrated away from 37xxs and are now using SNA/IP technologies such as tn3270(E), DLSw, EE and Web-to-host, across LAN interfaces. It is also not an alternative to IBM's Comm. Server.

The CCL/z is a sound, practical migration option for aging 37xxs. However, with integrated DLSw, IP-TGs, XOT etc. it is also offers significant value-added SNA modernization capabilities.

THE BOTTOM LINE

The CCL/z is a mainframe Linux-based replacement for the now discontinued IBM 3745/3746 controllers – replete with some powerful SNA modernization features. Its express goal is to enable SNA customers to maintain much of the unique, value-added networking functionality provided by ACF/NCP, NRF, and NPSI, in particular SNI, boundary functions, XRF, SSCP takeover and INN, independent of aging 37xx hardware.

SNA mission-critical networking has always been about steadfastness – slow, deliberate, measured progress. The CCL/z enables SNA customers to maintain this legacy. Now there is no need to scramble trying to find new, non-37xx related solutions. *The CCL/z, quite simply, gives 37xx customers the gift of time!*

Selected Acronyms/Glossary

ACF	Advanced Communications Functions: IBM product name prefix appended to post 1976 VTAM, NCP and SSP to denote that they supported cross-domain [i.e. multi-host] operations
APPN	Advanced Peer-to-Peer Networking; IBM's non-hierarchical mode of SNA introduced in 1986
BNN	Boundary Network Node: an SNA/APPN node reliant on a subarea or APPN network node for routing services
Boundary Functions	Set of services, including address and transmission header (TH) conversion, provided by a subarea node to the peripheral nodes attached to it
BSC	Binary Synchronous Communications: pre-SNA terminal control program that was widely used by 3270 and RJE terminals
CCU	Central Control Unit: 37xx processor
CDLC	Channel Data Link Control: original, c. 1970, channel protocol used between mainframes and 37xxs
CP	Central Processor: general purpose processors on contemporary IBM mainframes for handling business workloads
DLSw	Data Link Switching: extremely popular, LLC 2-over-TCP/IP transport and LAN-to-LLC 2 conversion technology initially developed by IBM and now available on routers
DR	Disaster recovery
EE	Enterprise Extender: end-to-end, mainframe-to-client, SNA transport across an IP network using HPR
EN	End Node: APPN/HPR peripheral node
EP	Emulation Program: NCP functions that emulated the pre-37xx, hard-wired 270x transmission control units
ESCON	Enterprise Systems Connectivity: IBM's fibre-based, 17MB/sec mainframe channel attachment scheme introduced in 1991
FICON	Fibre Channel Connectivity: IBM's fibre based, now up to 2Gbps channel attachment scheme originally introduced in 1998
GDPS	Geographically Dispersed Parallel Sysplex: DR at a distance
HiperSockets	TCP/IP-based, inter-LPAR communications scheme
HPR	High Performance Routing: the final iteration of SNA and APPN, essentially representing APPN Mk. II
IFL	Integrated Facility for Linux: specialized, lower-priced processors on contemporary IBM mainframes for handling Linux-only workloads
INN	Intermediate Network Node: essentially another name for SNA subarea nodes, especially 37xxs running NCP
ISDN	Integrated Services Digital Network: digital, circuit switching standard for voice and data
LCS	LAN Channel Station: IP-oriented channel protocol also initially developed for the 3172

LLC (2)	Link Layer Control (2): upper portion of the OSI Layer 2 data link control layer, with LLC 2 dealing with connection oriented traffic
LPAR	Logical Partition: an independent, mainframe 'image' (or a mainframe VM) created by PR/SM rather than by z/VM
LSA	Link Services Architecture: SNA-oriented channel protocol initially developed for use with the 3172
LU	Logical Unit: SNA's intelligent 'ports' for end user integration
MAE	Multiaccess Enclosure: IBM 2216 bridge/router internally Token-Ring attached to a 3746-900
NCP	Network Control Program; the pivotal control software that runs on a 37xx (and now CCL/z) and performs the SNA related functionality
NN	Network Node: APPN/HPR nodes capable of providing directory, configuration and routing services
NPM	NetView Performance Monitor
NPSI	NCP Packet Switching Interface: X.25-to-SNA
NRF	Network Routing Facility: NCP-based, LU-to-LU traffic routing that does not require the traffic to flow through mainframe VTAM
NSF	Network Supervisory Function: network management for X.25
NSI	Non-SNA Interconnection: BSC RJE support across SNA backbones
NTO	Network Terminal Option: NCP provided ASCII terminal-to-SNA protocol conversion (that is not supported by the CCL/z)
OSA(-E)	Open Systems Adapter(-Express): strategic, LAN adapters on new mainframes
OSN	OSA-for-NCP: z9-109 specific, software-based, virtual channel for VTAM to NCP-on-CCL/z interactions
PEP	Partitioned Emulation Mode: 37xx running both NCP & EP
PR/SM	Processor Resource/System Manager: IBM's mainframe partitioning, i.e. LPAR creating, mechanism
PU	Mainframes – processor unit [e.g. CP or IFL]; SNA – physical unit
QDIO	Queued Direct Input/Output: strategic and highly efficient mechanism for OSA-based mainframe I/O
QLLC	Qualified Logical Link Control: IBM's 'SDLC-across-X.25'
RJE	Remote Job Entry: 'cluster' with card reader/punch & printer
SDLC	Synchronous Data Link Control: IBM's initial, and subsequently near ubiquitous, serial data link control protocol for use with SNA
SNA	Systems Network Architecture (in case we forget)
SNA/IP	Generic term for any SNA-over-IP integration scheme [e.g. DLSw]
SNI	SNA Network Interconnection: ACF/NCP gateway-based mechanism for SNA interactions across autonomous SNA networks
SSCP	System Services Control Point: SNA's centralized control point, within VTAM, that provides directory, configuration and management services

SSP	System Support Program: mainframe-resident software for creating NCP load modules, loading 37xxs, and performing 37xx diagnostic functions such as traces and dumps
Subarea	Mainframe or 37xx node vis-à-vis SNA
TG	Transmission Group: a logical connection between two SNA nodes that could consist of one or more links
TIC	Token-Ring Interface Coupler: 37xx Token-Ring interface comprising of both hardware and software
tn3270(E)	Widely used, client-server-based mechanism for accessing SNA applications, across a TCP/IP network, using 3270 data stream
TPF	Transaction Processing Facility: IBM's high-performance, transaction processing monitor
VM	Virtual Machine: a mainframe image created by z/VM
VT	Virtual Terminal: a standard that describes a generic terminal
VTAM	Virtual Telecommunications Access Method: mandatory, mainframe software, now included with Comm. Server, that implements all of the requisite SNA control functions
XI	X.25 SNA Interconnect: not widely used scheme that enabled X.25 interactions across an SNA backbone (not available on CCL/z)
XOT	X.25-over-TCP: DLSw like router technology for transporting X.25 traffic, end-to-end, across a TCP/IP network
XRF	Extended Recovery Facility: IBM's SNA-centric, DR mechanism

THE AUTHOR: ANURA ['SNA'] GURUGÉ

Anura Gurugé [www.guruge.com] is an ex-IBMer (at Hursley, UK) from the 1970s. In addition to being a systems programmer he was involved with the 3270 program. His 1st book, "*SNA: Theory and Practice*" [which is still in print] was published in 1984, five years after he left IBM. For the next 15 years he was "Mr. SNA", and was heavily involved with Token-Ring switching, Frame Relay and Web-to-host. He was associated with the Token-Ring switching pioneer Nashoba Networks, which was acquired by Cisco Systems.

These days he is a consultant, a teacher, and writer. He is the Editor at Large for "*IT In-Depth*" [www.itindepth.com]. He is also the author of four other books, with his latest being "Web Services: Theory and Practice". In addition, he has published over 350 articles. In a career spanning 30 years, he has held senior technical and marketing roles in IBM, ITT, Northern Telecom, Wang and BBN. He can be contacted at (603) 455-0901 or anu@guruge.com.