<Network Working Group>
Internet Draft
Intended status: Experimental
Expires: June 2017

A. Y. Chen
R. R. Ati
Avinta Communications, Inc.

December 14, 2016

Adaptive IPv4 Address Space
draft-chen-ati-adaptive-ipv4-address-space-00.txt

Status of this Memo

Copyright Notice

Abstract

This document describes a solution to the Internet address depletion issue through the use of an existing Option mechanism that is part of the original IPv4 protocol. This proposal, named EzIP (phonetic for Easy IPv4), discusses the IPv4 public address pool expansion and the Internet system architecture enhancement aspects. It was originated by a study called ExIP (Extended IPv4) analyzing the use of the first available octet (eight bits) in the reserved private network pools (10/8, 172.16/12 and 192.168/16) to achieve a moderate address space expansion factor of 256 by each, while maintaining their familiar operation characteristics. Along the way, a parallel yet similar effort, called EnIP (Enhanced IPv4), was discovered. EnIP fully utilizes the same private network pools to increase the address space by a factor of 17.1M with end-to-end connectivity. EzIP is a superset that proposes one unified format for not only encompassing the considerations of both, but also identifying additional capabilities and flexibilities. For example, EzIP may expand an IPv4 address at least by a factor of 256 to as high as 256M without affecting the existing IPv4 public address assignments, while still keeping intact the current private networks for the 256M case if desired. The EzIP is in full conformance with the IPv4 protocol, and supports not only both categories of connectivity, but also their interoperability. The traditional Internet traffic and the IoT operations may coexist simultaneously without perturbing their existing setups, while offering end-users the freedom to choose one or the other. If the IPv4 public pool were reorganized, the assignable pool could be multiplied by 512M or even up to 2B times with end-to-end connectivity. EzIP may be deployed as a firmware enhancement to the Internet edge routers or private network gateways wherever needed, or simply installed as an inline adjunct module between the two, enabling a seamless introduction. The 256M case establishes a spherical layer of routers providing a complete interconnection between the Internet and end-users. This configuration enables the entire current Internet and private networks characteristics to remain intact. These proposed interim facilities would afford IPv6 more time to orderly reach the maturity and the availability levels required for delivering a long-term general service.

Table of Contents

1. Introduction

For various reasons, there is a large demand for IP addresses. It would be useful to have a unique address for each Internet device, such that if desired, any device may call any other. The Internet of Things (IoT) would also be able to make use of more routable addresses if they were available. Currently, these are not possible with the existing IPv4 facility.

By Year 2020, the population and number of IoTs are expected to reach 7.6 billion and 50 billion respectively, according to a recent Cisco online paper [1].

The IPv4 dot-decimal address format, consisting of four octets each made of 8 binary bits, results in the maximum number of assignable public addresses of 4.295 billion (calculated by 256 x 256 x 256 x 256, to be 4,294,967,296 - decimal exact). Using the binary / shorthand notation of 64K representing 256 x 256 (decimal 65,536), the full IPv4 address pool of 64K x 64K may be expressed as 4,096M, or 4.096B. Clearly, the demand is more than 13 times over the inherent capability available from the supply.

IPv6 with 128-bit hexadecimal address format offers a potential solution to this problem, but its global adoption appears to face certain challenges [2], [3]. Network Address and Port Translation (NAPT - commonly known simply as NAT) on private networks together with Carrier Grade NAT (CGNAT) over the Internet have been providing the interim solutions thus far. However, NAT modules slow down routers due to the state-table look-up process. As well, they only allow an Internet session be initiated by their respective own clients, impeding the end-to-end setup requests from remote devices that certain IoT operations desire.

If the IPv4 capacity could be expanded to eliminate this address pool deficiency while maintaining the familiar established operation conventions, and perhaps even offers reasonable reserve, the urgency will be relaxed long enough for the IPv6 to mature on its own pace.

To increase the Internet public address pool, there have been various proposals in the past. Among them, two recent efforts in particular

are referenced by this draft, namely ExIP and EnIP. The ExIP [4] study focuses on reclaiming part of a reserved private network address block, for example the third octet of 192.168/16, to be publicly routable at the edge of the Internet. By making use of this octet as semi-public address, the number of assignable public addresses is increased by a factor of 256 to become 1049B which is more than 20 times of the expected IoTs. This address expansion could be implemented in an inline module called Semi-Public Router (SPR) collocated with the Internet Edge Router (ER). Of course, the size of the resultant private networks will be reduced accordingly.

The Enhanced IP (EnIP) [5] project proposes to increase the available IPv4 public address space by a factor of 17.1M. Like IPv6, EnIP results in full end-to-end connectivity among the enhanced addresses. The EnIP implementation module, "NAT and EnIPNAT/translator", replacing existing private network gateway, is very similar to the SPR.

EzIP merges these two schemes into one uniform solution. Neither Internet Core (/ backbone) Router (CR), nor private network Routing Gateway (RG) needs to handle the Options added to the resultant IP header, since their designs recognize and preserve this Option mechanism, yet are not programmed to process the specific EzIP information. Even the Edge Routers (ER) may stay unchanged, if the SPR is deployed with the adjunct configuration during the introductory phase.

The assignable IPv4 compatible public address pool may be expanded significantly more upon incorporating other available IPv4 resources by the EzIP technique, as discussed in the latter part of this document.

1.1. Contents of this Draft

The rest of this draft begins with outlining the EzIP numbering plan. A modified IP header called EzIP header is introduced for carrying the EzIP address data in the Option words. The overview of the Internet architecture as the result of being expanded by the EzIP scheme, the EzIP header transitions through various routers and the operation considerations are discussed next, with details presented in Appendices A, B and C, respectively. Utilizing the EzIP approach, a range of possibilities of expanding the publicly assignable IPv4 address pool as well as enhancing the Internet operation flexibility are then described.

2. EzIP Overview

2.1. EzIP Numbering Plan

The ExIP technique which is the foundation of the EzIP plan began with making use of the reserved private network address pools in very much the same manner as Private Automatic Branch eXchange (PABX) telephone switching machines utilizing locally assigned “extension numbers” to expand the Public Switched Telephone Network (PSTN) capacity by replicating a public telephone line to multitudes of reusable private telephone numbers, each to identify a local instrument. At the first sight, this may seem odd, because the extension numbers of a PABX belong to a separate set from that of the public telephone numbers, while private network IP address is a specific subset reserved from the overall IPv4 pool that is otherwise all public. However, recognizing that neither of the latter two is allowed to operate in the other’s domain suggests that the proposed EzIP numbering system indeed may mirror the telephony case. In fact, the very basic form of the EzIP numbering is to make explicit the familiar subnetting process of 192.168/16 that has been performed routinely by consumer RGs (Residential / Routing Gateways) on residential premises for a long time.

2.1.1. To facilitate the following discussions, the 32 bits in a private network address notation are divided into three parts, namely Network, Extension and IoT No.’s as shown in Figure 1 below.

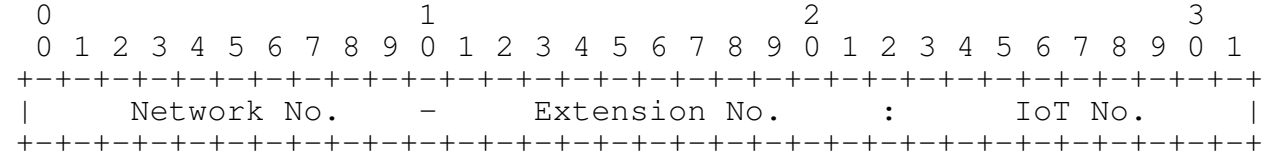


Figure 1 EzIP Address Notation (Generic)

The number of bits in the Extension No. part determines the multiplication factor to be applied by the EzIP process. The trailing IoT No. bits determine the size of the resultant private network. The Network No. part is the specific binary value of the remaining leading bits (the prefix) identifying an address block that will be reserved from the public IPv4 pool.

2.1.2. Following the general concept of subnetting, the unit for expanding an address does not need to be restricted to the boundary of an octet. This allows potentially finer grain resolution.

2.1.3. How to utilize the 32 bits leads to tradeoffs among EzIP operation characteristics. For example, maintaining the private network properties or establishing the end-to-end connectivity is just a matter of whether there are bits reserved for the IoT No.

2.1.4. This notation may be used to present two general categories of EzIP address types:

A. To retain the private network characteristics, the EzIP subnetting makes use of only the first available octet. For the common three private network address pools, we will have the following:

In Figure 2, 8 bits are available for IoT No., resulting in private networks each capable of 256 IoTs.

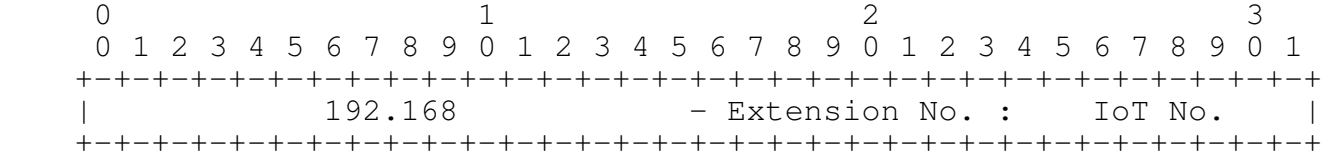


Figure 2 EzIP-1 (8 bits of 192.168/16 semi-publicly addressable)

In Figure 3, 12 bits are available for IoT No., resulting in private networks each capable of 4K IoTs.

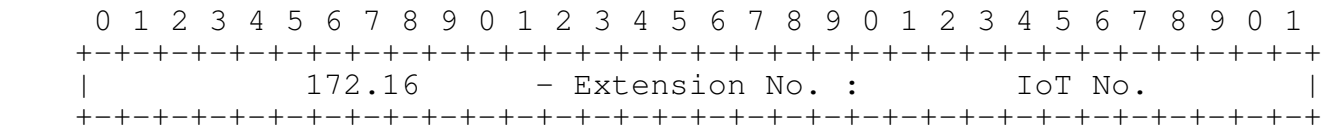


Figure 3 EzIP-2 (8 bits of 172.16/12 semi-publicly addressable)

In Figure 4, 16 bits are available for IoT No., resulting in private networks each capable of 64K IoTs.

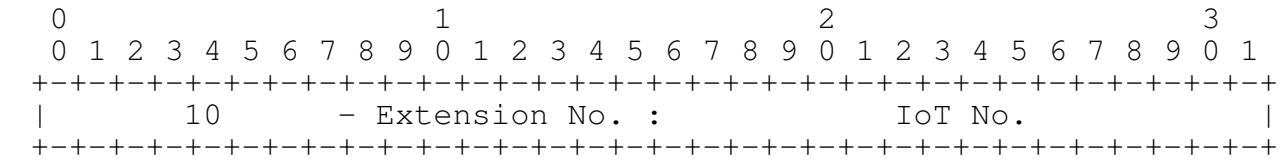


Figure 4 EzIP-3 (8 bits of 10/8 semi-publicly addressable)

B. To allow direct access from the Internet, EzIP makes use of all available bits in a reserved private network address as Extension No., leaving no bit for the IoT No. The resultant private network will have no RG, but only one IoT that is directly connected to the Internet:

In Figure 5, 16 bits are assigned for Extension No., resulting in 64K IoTs directly addressable from the Internet.

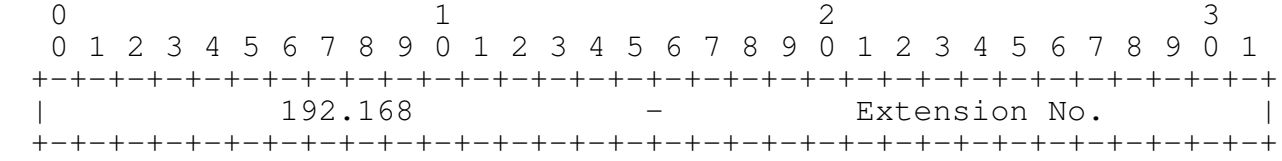


Figure 5 EzIP-4 (16 bits of 192.168/16 semi-publicly addressable)

In Figure 6, 20 bits are assigned for Extension No., resulting in 1M IoTs directly addressable from the Internet.

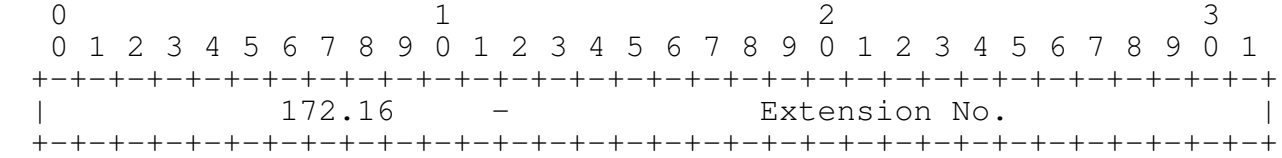


Figure 6 EzIP-5 (20 bits of 172.16/12 semi-publicly addressable)

In Figure 7, 24 bits are assigned for Extension No., resulting in 16M IoTs directly addressable from the Internet.

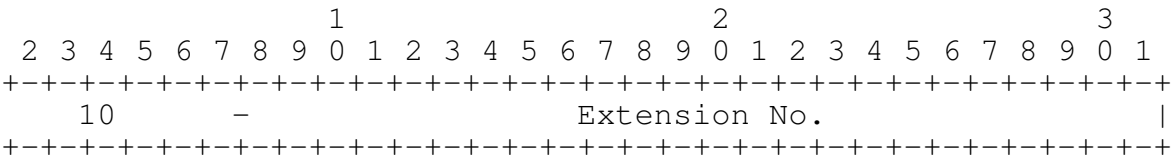


Figure 7 EzIP-6 (24 bits of 10/8 semi-publicly addressable)

For cross reference purpose, EzIP-1 through EzIP-3 are the same numbering types used by the ExIP study, while EzIP-4 through EzIP-6 are used by the EnIP project.

Figure 8 summarizes the number of possible publicly and privately assignable addresses for each original IPv4 public address under different configurations.

	192.168/16	172.16/12	10/8
Basic IPv4			
Address Bits*	32	32	32
Public	1	1	1
Private	64K	1M	16M
(ExIP)	EzIP-1	EzIP-2	EzIP-3
Address Bits*	40	40	40
Semi-Public	256	256	256
Private	256	4K	64K
(EnIP)	EzIP-4	EzIP-5	EzIP-6
Address Bits*	48	52	56
Public	64K	1M	16M
Private	1	1	1

Notes:

a. * -- Effective Overall Public Address Length

b. For each Public-Private pair, the numbers of addresses are multiplicative, not additive.

Figure 8 Basic IPv4 Address Expansion Configurations

2.2. EzIP System Architecture

With six basic EzIP expansion types, it is difficult to include them all in one single system architecture diagram. A complete set of system architectural diagrams is presented in Appendix A. To facilitate the presentation, a partial system diagram covering only the 192.168/16 (EzIP-1 and EzIP-4) portion as presented in Figure 9 below will be utilized for the discussions that follow.

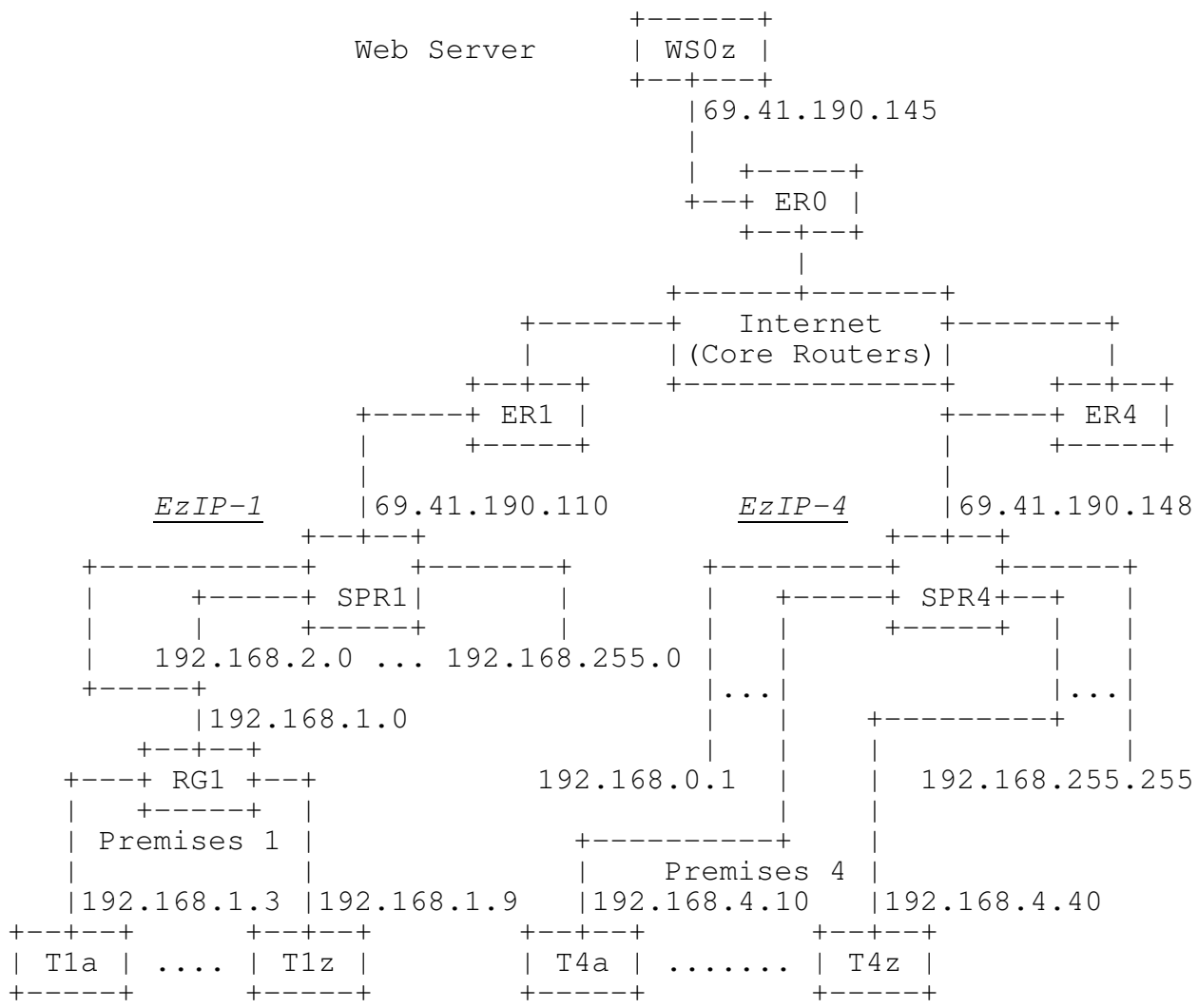


Figure 9 EzIP System Architecture-A (192.168/16 Portion)

	Basic IPv4	EzIP-capable
Internet Edge Router (ER)	ER0, ER1, ER4	----------
Internet of Things (IoT)	T1a, T4a	T1z, T4z
Routing Gateway (RG)	RG1	----------
Semi-Public Router (SPR)	----------	SPR1, SPR4
Web Server (WS)	----------	WS0z

Figure 10 EzIP-1 & EzIP-4 Components

2.2.1. Referring to the left portion labeled *EzIP-1* of Figure 9, instead of assigning each premises a public IPv4 address as in the current practice, an SPR like SPR1, is inserted between an Internet Edge Router (ER1) and its connections to private network Routing Gateways like RG1, for utilizing the third octet, such as 192.168.nnn/24 (nnn = 0 through 255) to identify respective entities. The RG1 serves either a LAN or a HAN. On each LAN / HAN, the fourth octet "mmm" of 192.168.nnn.mmm/32 continues to be used by the RG1 to identify the IoTs it serves. This is how common RGs are being configured today anyway (Factory default values of nnn are usually 0, 1, 2, 10, etc.)

2.2.2. The right portion of Figure 9 is labeled *EzIP-4*. Here SPR4 assigns the full range of the available 192.168/16 IP addresses (the third and fourth octets) individually to T4a through T4z. Consequently, these IoTs are directly accessible from any remote device on the Internet.

2.2.3. Since the existing physical connections to subscriber's premises do appear at the ER, it is natural to have SPRs be collocated with their ER. It follows that the simple routing function provided by the new SPR modules may be absorbed into the ER through a straightforward operational firmware enhancement. Consequently, the public - private demarcation line will remain at the RG where currently all utility services enter a subscriber's premises.

2.2.4. To identify each of these devices, we may use a three part address format "IPv4 - Semi-Public: TCP Port No.". The following is how each of the IoTs in Figure 9 may be identified.

RG1: 69.41.190.110-192.168.1.0

T1a: 69.41.190.110-192.168.1.0:3

T1z: 69.41.190.110-192.168.1.0:9

T4a: 69.41.190.148-192.168.4.10

T4z: 69.41.190.148-192.168.4.40

Note that to simplify the presentation, it is assumed at this juncture that the conventional TCP (Transmission Control Protocol) [6] Port Number, normally assigned to T1a and T1z by RG1's NAT module upon initiating a session, equals to the fourth octet of that IoT's private IP address that is assigned by the RG1's DHCP (Dynamic Host Configuration Protocol) [7] module as ":3" and ":9", respectively. Such numbers are unique within each respective private network. They are adequate for the discussion purpose here. However, considering security, as well as allowing each IoT to have multiple simultaneous sessions, etc., this direct correlation shall be avoided in actual practices by following the NAT operation conventions as depicted by the examples in **Error! Reference source not found.**.

2.3. IP Header with Option Word

To transport the EzIP Extension No., we will make use of the Option word in the IP header as defined in Figure 9 of [RFC791] [8]. This mechanism has been used for various cases in the past. Since they were mostly for utility or experimental purposes, however, their formats may be remote from the incident discussion.

2.4. Examples of Option Mechanism

The following two cases specifically deal with the address pool issues. They are referenced here to facilitate the appreciation of the Option mechanism.

A. EIP (Extended Internet Protocol) - [RFC1385] [9] (Assigned but now deprecated Option Number = 17) by Z. Wang: This approach attempted to add a new network layer on top of the existing Internet for increasing the addressable space. Although equipment near the end-user would stay unchanged, equipments around the Internet Core Routers (CR) apparently had to go through rather involved upgrade procedures.

B. EnIP (Enhanced IPv4) - Internet Draft [5] (temporarily utilizing Option Number = 26) by W. Chimiak: This work makes use of

the reserved private network addresses to extend the public pool by trading the private network operation for end-to-end connectivity. The EnIP and ExIP approaches closely resemble each other.

2.5. Basic EzIP Header

The basic EzIP header format uses the Option ID field to convey the value of the "Network No." as well as the length of the "Extension No.". This header has the capacity to handle up to two octets of the "Extension No." on either end of a connection.

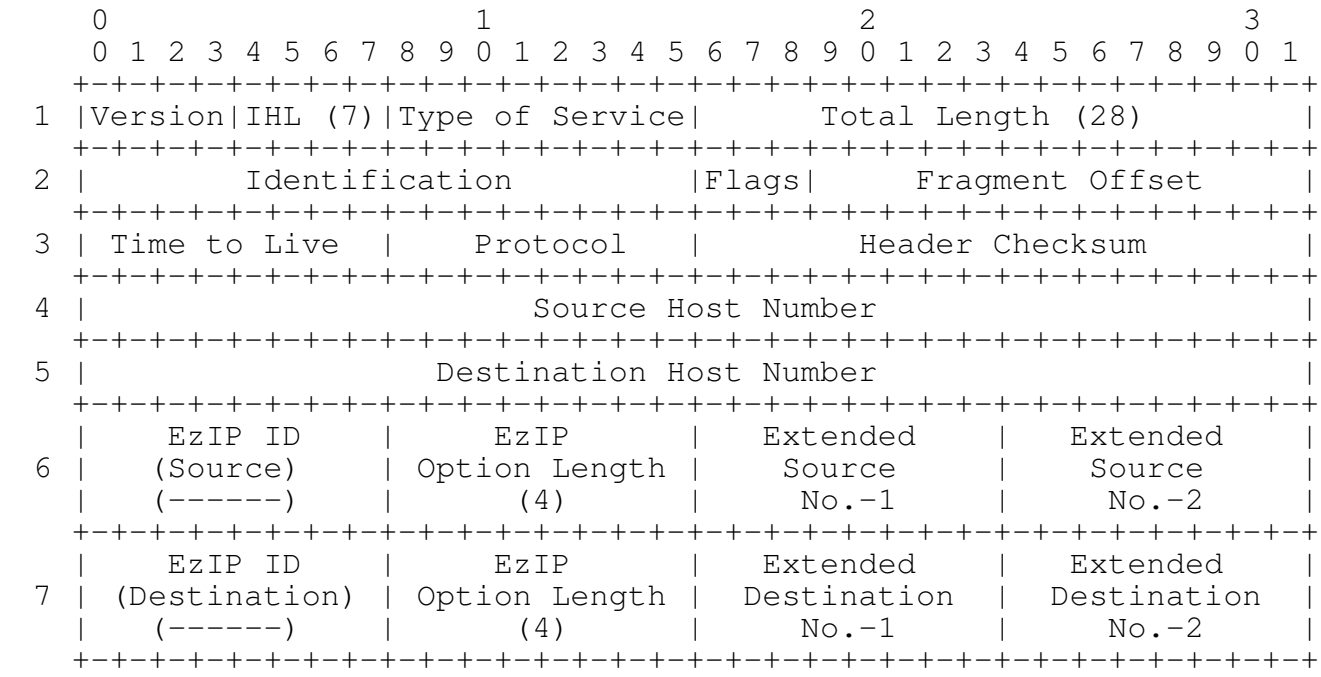


Figure 11 Basic EzIP Header (Two Octet)

To transport an IP header for T4z at the Source end and RG1 at the Destination end, Figure 12 depicts an EzIP header example:

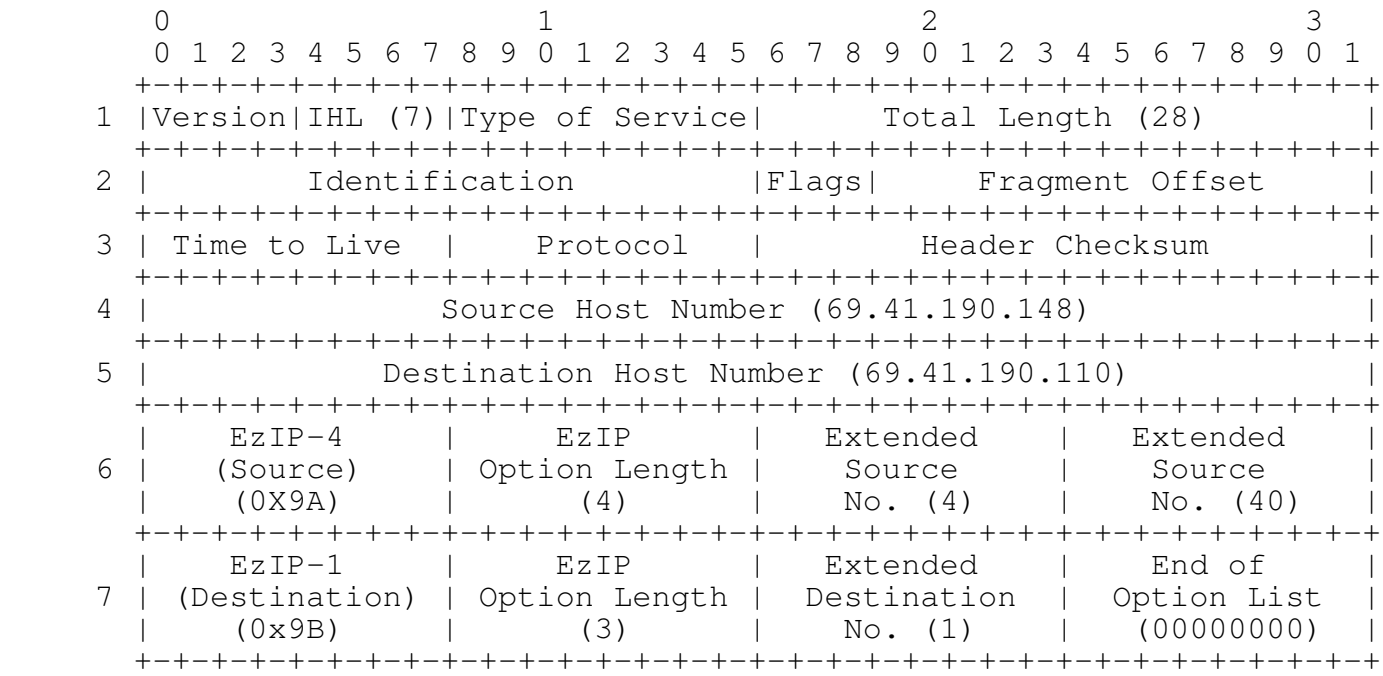


Figure 12 EzIP Header Example 1

Note that the Option IDs 0x9A (Option Number = 26) and 0x9B (Option Number = 27), both representing Network No. 192.168/16 while conveying the Extension No.'s being two and one octet, respectively, in the above figure, are arbitrarily chosen from the currently available Option Numbers list [10]. Since RG1 extension No. has only one octet, the "End of Option list" Option is used to fill up word 7.

If the transmission direction is reversed, types of EzIP extension used by the Source and the Destination will be interchanged as well. The unused octet will now be at the end of word 6. The "No Operation" Option should be used as the filler shown in Figure 13:

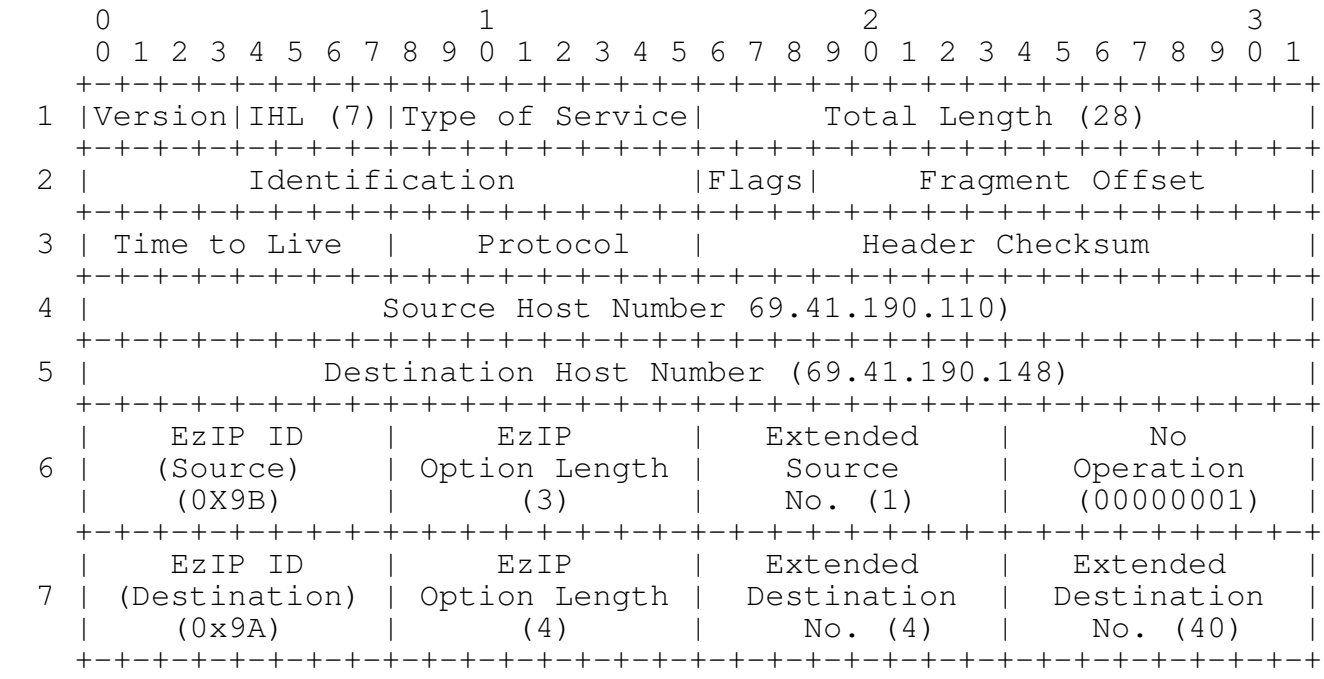


Figure 13 EzIP Header Example 2

2.6. EzIP Operation

With half a dozen of EzIP types, it would be very tedious and distracting to go through all combinations of IP header configurations and their transitions through the network. To convey the general scheme, **Error! Reference source not found.** presents examples of EzIP header transitions through routers among IoTs having EzIP-1 and EzIP-4 types of addresses, with and without EzIP capability.

To introduce the EzIP approach into an environment where EzIP-unaware IoTs like T1a and T4a will be numerous for a long time to come, a SPR must be able to follow certain decision rules to determine which type of service to provide for achieving a smooth transition. Appendix C outlines such logic and related considerations.

2.7. Generalizing EzIP Header

2.7.1. The basic EzIP header shown in Figure 11 with up to two octet Extension No. format is not capable of EzIP-5 and EzIP-6 types with 20 and 24 bit, respectively. One extra octet is needed on each end of such a connection. An additional word in the header, however, will have two octets unused. To take advantage of this spare resource, we might as well consider a header format shown in Figure 14 that can transport the full 4 octet (32 bit) extension addresses of both ends. This is similar as the EnIP header [5], except more flexible by allowing EzIP type being independent of that at the other end.

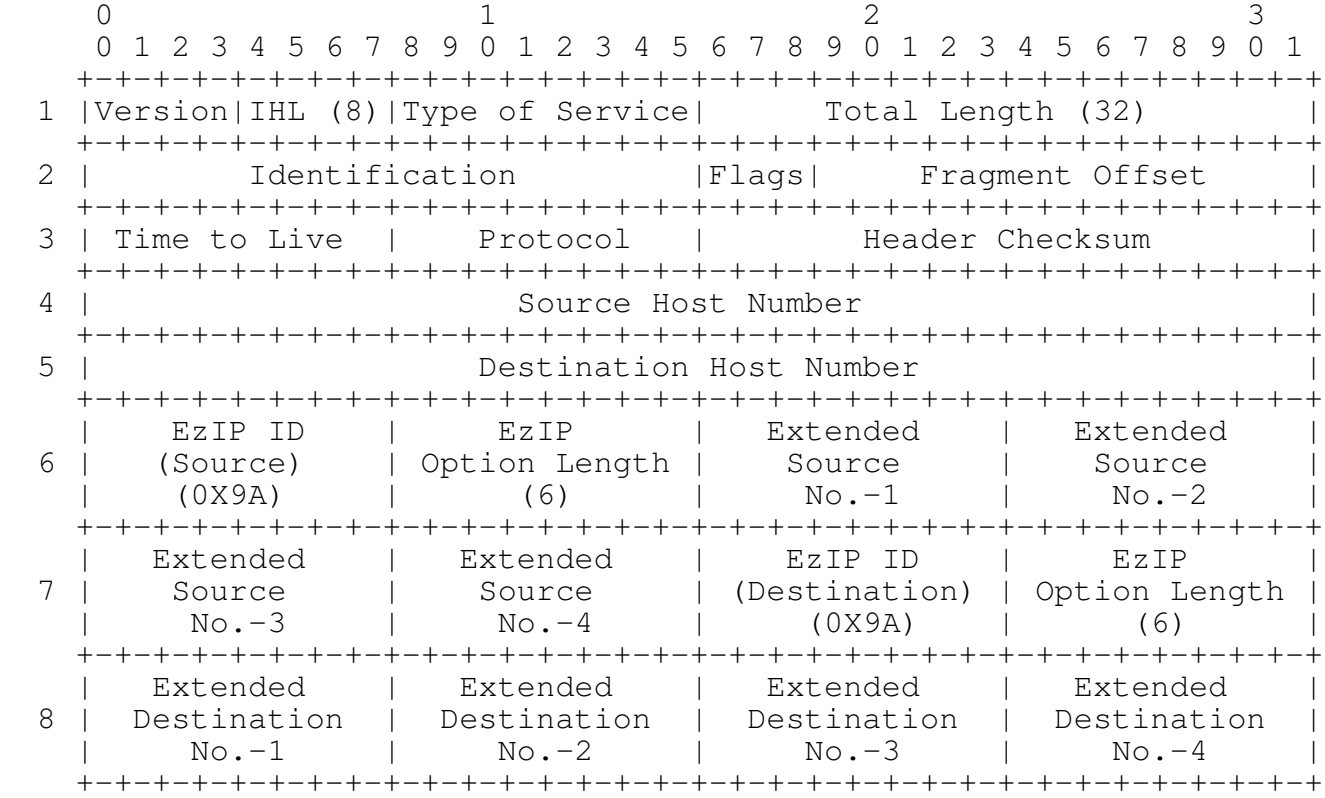


Figure 14 Full EzIP Header (Four octet)

2.7.2. In brief, Figure 12 or Figure 13 with seven words (40% overhead) having two octet capacity is suitable to transport EzIP-1 through EzIP-4 types consisting of one or two octet Extension No. EzIP-5 and EzIP-6 require the next IP header size which is eight words (60% overhead) as shown in Figure 14.

2.7.3. Being a superset, utilizing “No Operation” or “End of Option List” type of fillers, Figure 14 is capable of handling information for EzIP-1 through EzIP-4 just as well. The question then becomes; whether the extra 20% overhead when handling EzIP-1 through EzIP-4 headers is tolerable? If so, the single Figure 14 format may be used for all EzIP cases.

2.7.4. With the “Network No.” prefixes of the well-know private network addresses all explicitly carried by the IP header of every packet as shown in Figure 14, the Option Number only needs to identify the length of the “Extended No.”. Consequently, one Option Number is sufficient to represent EzIP-1 through EzIP-3 that only the first available octet is used for the Extension No. Similarly, one single Option Number representing EzIP-4 through EzIP-6 conveys the condition that all available bits are to be used for Extension No.

2.7.5. One potential drawback of the full four octet EzIP header is that it may cause Internet routers to intercept a packet for containing a disallowed (private network) IP address, although positioned at a location of the header normally not designated for address information.

2.7.6. By harmonizing EzIP-4 to -6 (EnIP) with EzIP-1 to -3 (ExIP) into one common (EzIP) format, enjoying which operating characteristics will simply be the result of a user subscribing to an EzIP address type appropriate for how he wishes to use his IoT.

3. EzIP Deployment Strategy

Although the eventual goal of the SPR is to support both web server access by IoTs from behind private networks and direct end-to-end connectivity between IoTs, the former application should be addressed first to immediately relieve the basic address shortage issue. Once the IoTs on both ends of an intended connection are served by SPRs, it will be natural to realize the latter.

A. Architecturally

Since the design philosophy of the SPR is an inline module between the Internet ER (Edge Router) and the private network RG (Routing Gateway), SPR introduction process may be flexible.

A.1. SPRs may be collocated with ERs to begin providing the CGNAT equivalent function. This may be done immediately without affecting the existing Internet (edge and core) routers. EzIP-capable IoTs will then take advantage of the faster bi-directional routing

services through the SPRs by initiating a communication session with an EzIP header.

A.2. Alternatively, a SPR may be deployed as an adjunct module before an existing RG to realize the same EzIP functions on private premises, even if the serving Internet Service Provider (ISP) has not enhanced ERs with the EzIP capability. This empowers individual subscribers to enjoy the new EzIP capability on their own.

B. Functionally

B.1. First, an ISP should install SPRs in front of business web servers so that new routing branches may be added to support the additional web servers for expanding business activities. Alternatively, this may be achieved by deploying new web servers with the SPR function built-in.

B.2. On the subscriber side, SPRs should be deployed to relieve the public address shortage issue, and to facilitate the access to new web servers.

C. Permanently

In the long run, it would be best if SPRs are integrated into ERs by upgrading the latter’s firmware to minimize the hardware.

Appendix C details the considerations in implementing these outlines.

4. Updating Servers to Support EzIP

Although the IP header Option mechanism utilized by EzIP was defined a long time ago as part of the original IPv4 protocol, it has not been used much in daily traffic. Certain current Internet facilities were thus optimized without considering the Option mechanism. They need be adjusted to provide the same performance to EzIP packets. There are also utility type of servers need be updated to support the longer EzIP address. For example;

A. Fast Path

Internet Core Routers (CRs) are currently optimized to only provide the “fast-path” (through hardware line card) routing service to packets without Option word in the IP header [11]. This puts EzIP packets in a disadvantage, because EzIP packets would be put through the “slow path” (processed by CPU’s software before giving to the correct hardware line card to forward), resulting in a slower throughput. Since the immediate goal of the EzIP is to ease the

address pool exhaustion issue, subscribers not demanding for high performance traffic may be assigned with the facility provided through EzIP. This gives time for Internet routers to update so that EzIP packets with authorized Option numbers will eventually be recognized for receiving the “fast-path” service.

B. Connectivity Verification

One frequently used utility for verifying baseline connectivity, commonly referred to as the “PING” function in PC terminology, needs be able to transport the full EzIP address that is longer than the standard 32 bit IPv4 address. There is an example of an upgraded TCP echo server in [RFC862] [12].

C. Domain Name Server (DNS)

Similarly, the DNS needs to expand its data format to transport the longer IP address created by EzIP. This already can be done under IPv6. Utilizing the experimental IPv6 prefix 2001:0101 defined by [RFC2928] [13], EzIP addresses may be transported as standardized AAAA records.

These topics are discussed in more detail under the IETF Draft RFC, Enhanced IPv4 – V.03 [5].

5. EzIP Enhancements

To minimize disturbing any assigned addresses, deployed equipment and current operation procedures, etc., the EzIP derivations so far are conducted under the constraint of utilizing only the existing three reserved private network address blocks. Beyond such, there are other possibilities. In the long run, EzIP may significantly expand the current IPv4 public address pool through the employment of such additional resources outlined below.

A. In reviewing the IP Option Number assignments [10], it is discovered that more than a dozen of them are currently available. That is, besides five numbers, 26, 27, 28, 29 & 31 that have never been assigned, there are eleven numbers assigned earlier but have been deprecated due to the end of associated experiments. If we take six such numbers, one to represent each of the six EzIP extension types, the EzIP-1 to EzIP-3 cases will multiply the IPv4 public address pool by a factor of 256, individually, or a combined factor of 768, resulting in 3,145.728B, or 3.146KB publicly assignable addresses. Similarly, we can use one Option Number for each of the EzIP-4, -5 and -6 cases to multiply IPv4 pool by 64K, 1M and 16M (a total of 17.1M) fold, respectively, to the combined total of 69.894MB

addresses. These capacities are over 63 and 1.4M times of the expected Year 2020 IoTs, respectively.

B. EzIP-8: If all Option numbers were made available, each representing one EzIP Network No. prefix, up to 32 private network address blocks, like the 10/8 could be utilized by EzIP. To determine the upper limit of this scenario, let's assume that we could employ 31 additional 10/8 type address blocks, say by re-designating 11/8 through 41/8 as private network blocks. These enable us to expand each existing IPv4 public address by 32 x 16M or 512M fold. Since this block of 512M addresses have to be removed from the basic public pool, the resulting total addresses will be (4.096B - 512M) x 512M, or 1,835MB. This is over 35M times of the predicted number of IoTs (50B) by Year 2020. It certainly has the capacity to deal with the short- to mid- term public IP address needs.

C. The above may be condensed for a more efficient operation. For example, a single 224/3 block contains the same amount of 512M addresses may be chosen upon re-allocation of currently assigned IPv4 public addresses so that just one Option Number may represent it. Now that we have freed up 31 Option numbers, we could allocate up to 31 more /3 address blocks for EzIP operation that provides even more extension address resource. However, this last step will exceed the total capacity of the IPv4 pool. On the other hand, this line of reasoning leads to the next observation.

D. EzIP-9: One interesting consequence of the EzIP header in Figure 144 capable of transporting the full 32 bit private network address is that the Extension No. may be as long as practical. That is, we can go to the extreme of reserving only one bit for the Network No., and leaving nothing for the IoT No. With these criteria, the current IPv4 pool may be divided into two halves, reserving one half of it (about 2B addresses) as a private network with prefix equal to "1" as the Network No., and all trailing 31 bits designated as Extension No. Each of the remaining 2B addresses (with prefix equals to "0") of the basic IPv4 pool may then be expanded 2B times through the EzIP process, resulting in a total of 4BB addresses that are IPv4 compatible and capable of full end-to-end connectivity. This is roughly 80M times of the Year 2020 IoTs.

E. EzIP-7: On the other hand, this full 32 bit EzIP addresses transport facility may be applied to the elusive IPv4 240/4 block (240/8 - 255/8) consisting of 256M addresses that has become "RESERVED for Future use" [14] as the result of the historical address assignment evolution. Since this block is not suitable for being used as public address, it might as well be re-classified as an additional (the fourth) reusable private network pool. Then, the SPR

may use this block as the extension address pool in the EzIP process. Following this approach, each current IPv4 public address may be multiplied by 256M times based on only one Option Number. Since the 240/4 block could not be used for public addressing, the size of the publicly assignable IPv4 pool has actually been only 3.84B (4.096B - 256M). So, the net public addressable pool created from this approach is 983MB (3.84B x 256M), which is over 19.6M times of the expected Year 2020 IoTs. This scheme is very close to EzIP-8. Although half of the capacity, this manifestation has the advantage of circumventing reassignment of public IPv4 addresses.

The following compares various IPv4 public address pool expansion configurations.

Extension Scheme	Option Used	Effect. AddBits	Expansion Factor	Assignable Pub Add	SUP/DMD	Connect-ivity
IPv4 Public Address Block Assignments Unchanged						
Ex I P	EzIP-1	1	40	256	978.69B	19.6	PrivNet
	EzIP-2	1	40	256	978.69B	19.6	PrivNet
	EzIP-3	1	40	256	978.69B	19.6	PrivNet
En I P	EzIP-4	1	48	64K	244.67KB	5K	EndToEnd
	EzIP-5	1	52	1M	3.82MB	77K	EndToEnd
	EzIP-6	1	56	16M	61.17MB	1M	EndToEnd
EzIP-7(240/4)	1	60	256M	978.69MB	20M	EndToEnd
IPv4 Public Address Block Assignments Adjusted						
EzIP-8 (224/3)	1	61	512M	1.84BB	37M	EndToEnd
EzIP-9 (Half of IPv4 Pool)	1	63	2B	4BB	80M	EndToEnd

Notes:

a. EzIP-1 through EzIP-7 Assignable Public Addresses calculated with the net basic IPv4 public address pool of 3.823B after removed the 240/4, 10/8, 172.16/12 and 192.168/16 blocks from the basic 4.096B

b. EzIP-8 and EzIP-9 Assignable Public Addresses calculation started from scratch based on the full IPv4 pool of 4.096B minus only the specific portion used for extension purpose

c. "SUP/DMD": Ratio of EzIP SUPplied publicly assignable addresses to IoT DeManD by Year 2020

d. Each group of EzIP-1 to -3 and EzIP-4 to -6 may use only one Option number if "four octet" EzIP headers are used.

Figure 15 IPv4 Address Multiplication Possibilities

F. It is important to note that schemes summarized in Figure 15 are not mutually exclusive but mostly complementary. Except the last two cases (EzIP-8 and EzIP-9) that are intend to demonstrate the potential public address sizes by starting from the full 4.096B IPv4 pool ignoring the current assignments and reservations, EzIP-1 through EzIP-7 may be applied to the same public IPv4 address since they are distinguished from one another by the Option Numbers representing the network prefix and the number of Extension No. bits. These enable an ISP to offer a rich mixture of addresses for the subscribers to choose from.

G. An address extended by EzIP-4 through EzIP-7 directly connecting an IoT to the Internet could nevertheless be replaced by a private network established through an RG as described at the end of Appendix B. The EzIP-7 can best take advantage of this approach, because the 240/4 address block is totally segregated from the three conventional private network pools, thus avoiding confusing the Internet routers. Essentially, the subscribers, appearing as private networks and directly connected IoTs, will interface with a complete spherical layer of secondary ERs (made of the SPRs) that wraps the entire existing Internet within by utilizing a never assigned address pool.

H. In summary, the EzIP technique may expand the current IPv4 public address pool with a wide range of multiplication factors. It may be 256 folds while maintaining the current private network properties except with reduced size, and from 64K to 256M folds while offering direct end-to-end connectivity. In addition, multiplication factor of 512M may be achieved with some re-assignments of the IPv4 blocks. Lastly, the address capacity could even become 1B times of the current 4B pool with fully direct end-to-end connectivity. However, these last two EzIP manifestations rely on significant realignments

of the current address blocks. In between, we could have an IPv4 based Internet that can simultaneously support private networks along with directly accessible IoTs for interconnectivity and interoperability.

I. Overall, EzIP-7 may be the optimum choice. It utilizes a block of IPv4 addresses that could not be assigned as public identifiers anyway. It needs only one Option Number. Furthermore, existing private network setups may remain intact. Essentially, EzIP-7 introduces a new layer of routers (made of the SPRs) that expands the Internet address capacity by 256M fold uniformly, with minimum disturbance to the current Internet operations.

6. Security Considerations

The EzIP solution is based on an inline module called SPR that intends to be as transparent to the Internet traffic as possible. Thus, no overall system security degradation is expected.

7. IANA Considerations

This draft does not create a new registry nor does it register any values in existing registries; no IANA action is required.

8. Conclusions

This draft RFC describes an enhancement to IPv4 operation utilizing IP header Option mechanism. Because the design criterion is to enhance IPv4 by extending instead of altering it, the impact on already in-place routers and security mechanisms is minimized.

To resolve the IPv4 public address pool exhaustion issue, a technique called EzIP (phonetic for Easy IPv4) making use of the reserved private network address blocks is proposed.

The basic EzIP intention is to maintain the existing private network configuration. If an Extension No. for EzIP is chosen from the very end of the 32 bit reserved private network address, leading to no address bit available to assign on the resultant network, the IoT being served is directly accessible from any remote device in the Internet. An IoT may communicate through the Internet with either type of the connectivity, depending on which type of extension address its owner wishes to subscribe and to utilize with.

The basic EzIP header uses two added words (or 40% overhead) to the IP header for transporting two octets of an Extension No. To carry the full four octet EzIP extension address, a third added word is

needed resulting in a 60% overhead. The latter, being a superset of the former, may be used for all EzIP cases if the extra 20% overhead is tolerable for cases when the larger capacity is not necessary.

At the extreme end of the spectrum, the EzIP scheme could be configured to support an IPv4 compatible pool of up to 4BB addresses with full direct end-to-end connectivity.

Last but not the least, the "RESERVED for Future use" 240/4 block may be re-classified as the fourth reusable private network pool, so that the SPR may use it as the EzIP extension address. This pool can multiply each current IPv4 public address by 256M times based on only one Option Number, while all existing subscriber premises setups (private networks and directly connected IoTs) may remain unchanged. This manifestation of EzIP technique may be the optimal solution to our needs.

9. References

9.1. Normative References

(None)

9.2. Informative References

[1] https://nishithsblog.files.wordpress.com/2014/04/internet-of-things-market-forecast.jpg

[2] http://stats.labs.apnic.net/ipv6

[3] https://ams-ix.net/technical/statistics/sflow-stats/ether-type

[4] http://www.avinta.com/phoenix-1/home/IETF-Draft-ExIP.pdf

[5] https://tools.ietf.org/html/draft-chimiak-enhanced-ipv4-03

[6] https://tools.ietf.org/html/rfc793

[7] https://www.ietf.org/rfc/rfc2131.txt

[8] https://tools.ietf.org/html/rfc791

[9] https://tools.ietf.org/html/rfc1385

[10] http://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml

[11] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.477.1942&rep=rep1&type=pdf

[12] https://tools.ietf.org/html/rfc862

[13] https://tools.ietf.org/html/rfc2928

[14] http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml

10. Acknowledgments

The authors would express their deep appreciation to Dr. W. Chimiak for the enlightening discussions about his team's efforts and experiences through the EnIP development.

This document was prepared using 2-Word-v2.0.template.dot.

Appendix A EzIP System Architecture

A.1. EzIP System Part A

The EzIP-1 and EzIP-4 portions of the EzIP system has already been shown as Figure 9 in the main body of this Draft document.

A.2. EzIP System Part B

The EzIP-2 portion maintains private network operation characteristics, while EzIP-5 portion delivers end-to-end connectivity.

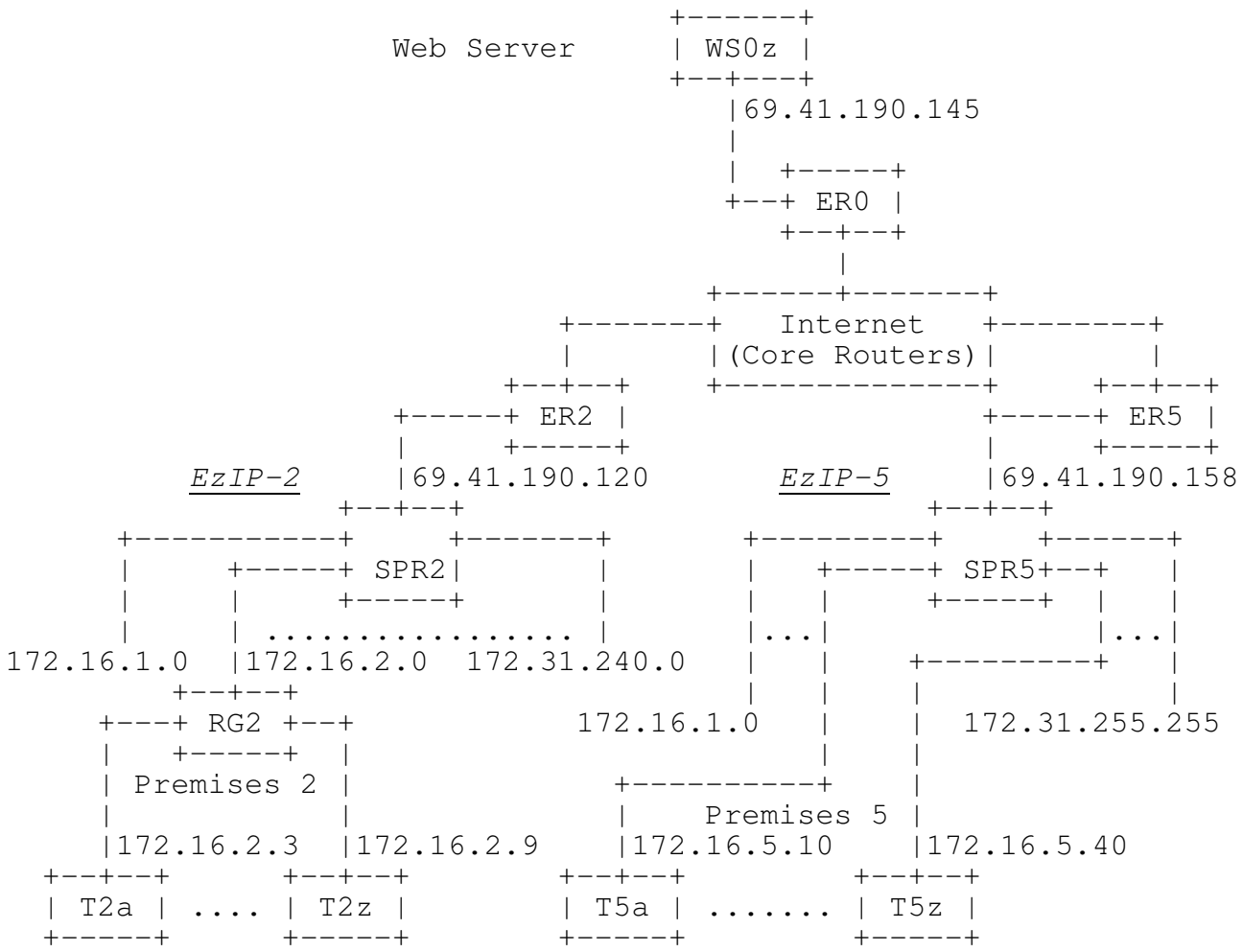


Figure 16 EzIP System Architecture-B (172.16/12 Portion)

A.3. EzIP System Part C

The EzIP-3 portion maintains private network operation characteristics, while EzIP-6 portion delivers end-to-end connectivity.

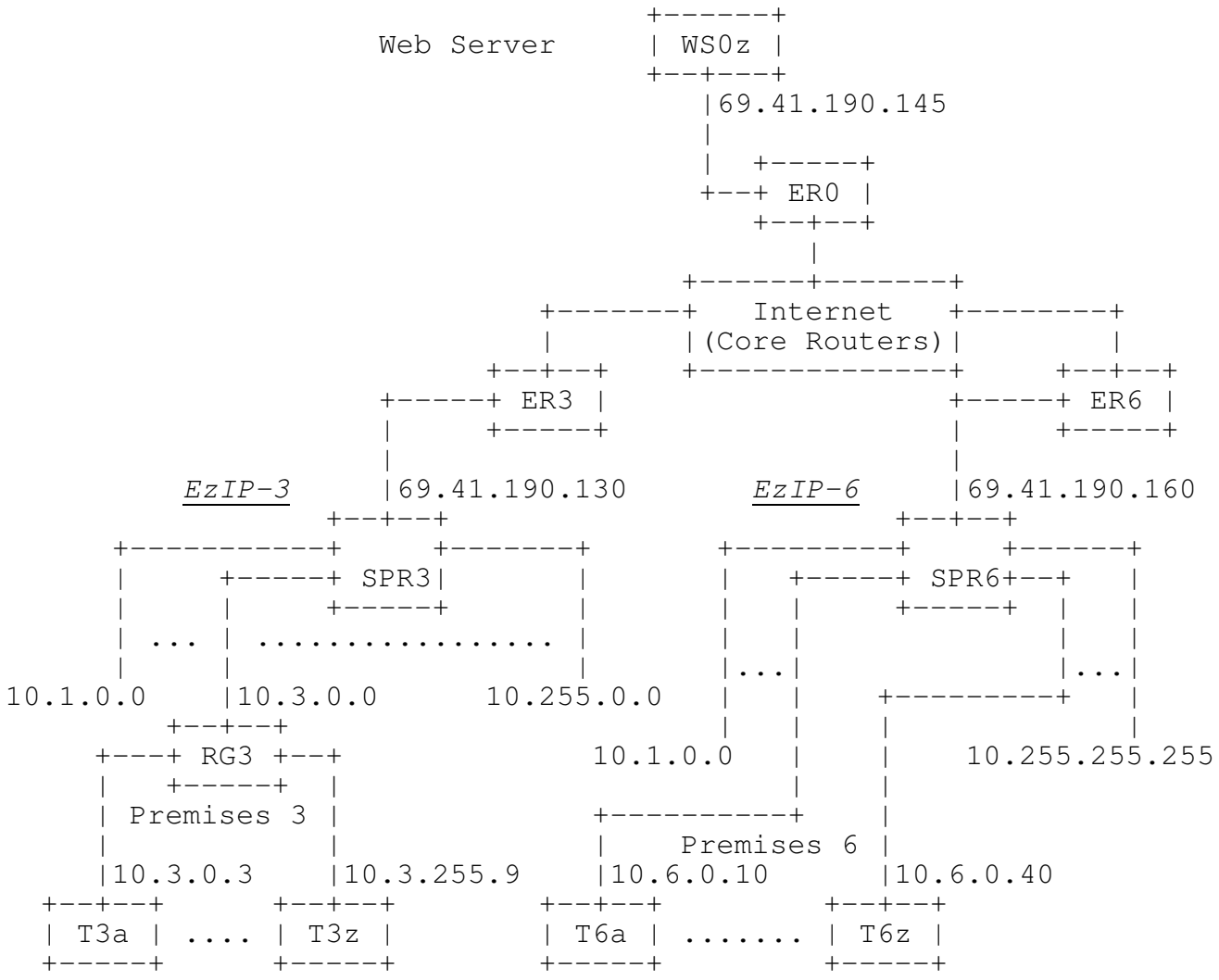


Figure 17 EzIP System Architecture-C (10/8 Portion)

A.4. EzIP System Part D

Utilizing 240/4, the EzIP provides a “spherical shell” of routable addresses wrapped around the entire current Internet (CRs and ERs), separating it from the subscribers’ IoTs that are either directly addressable from the Internet such as T7z, T8z, or behind existing private networks like RG7, RG8.

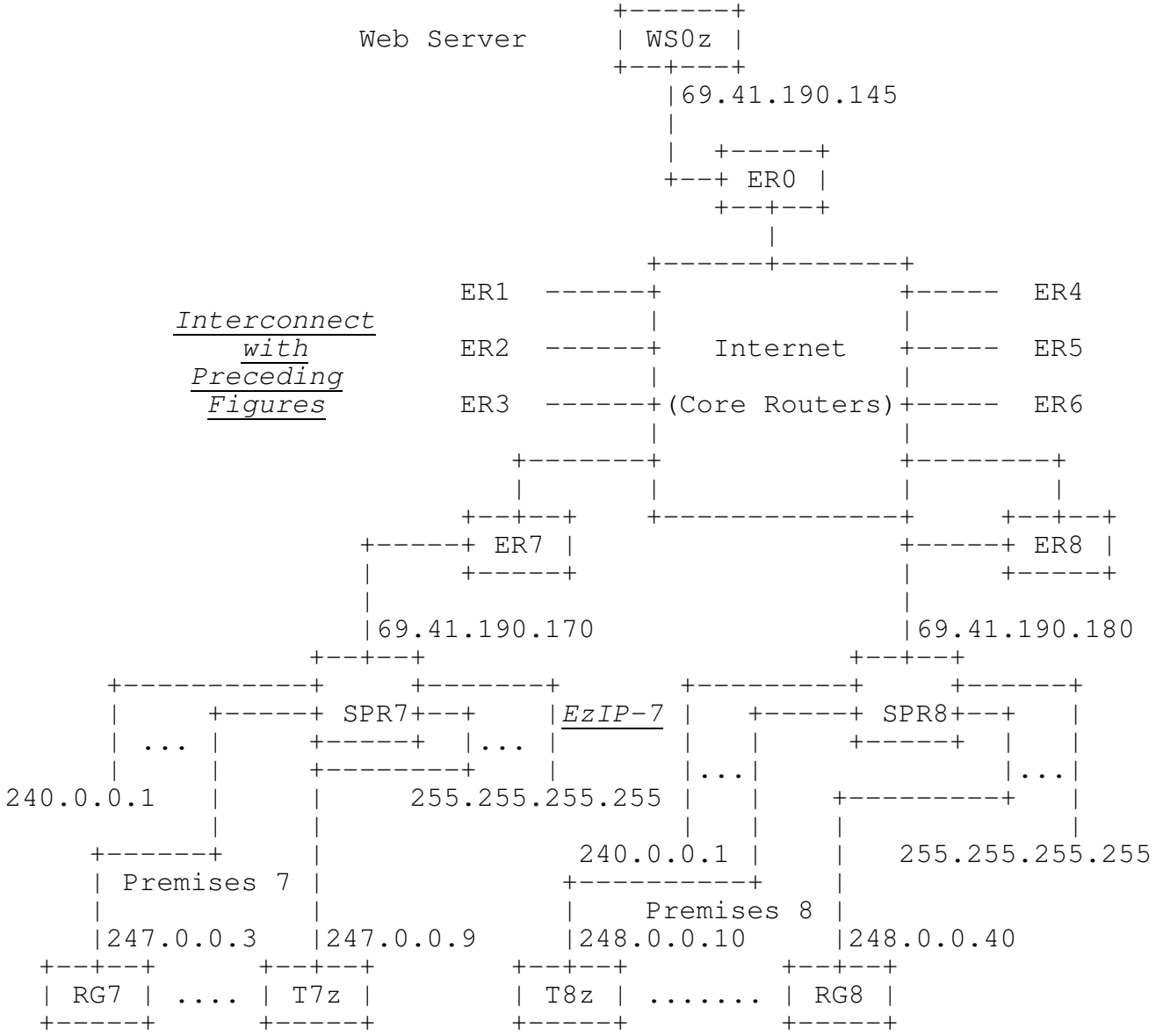
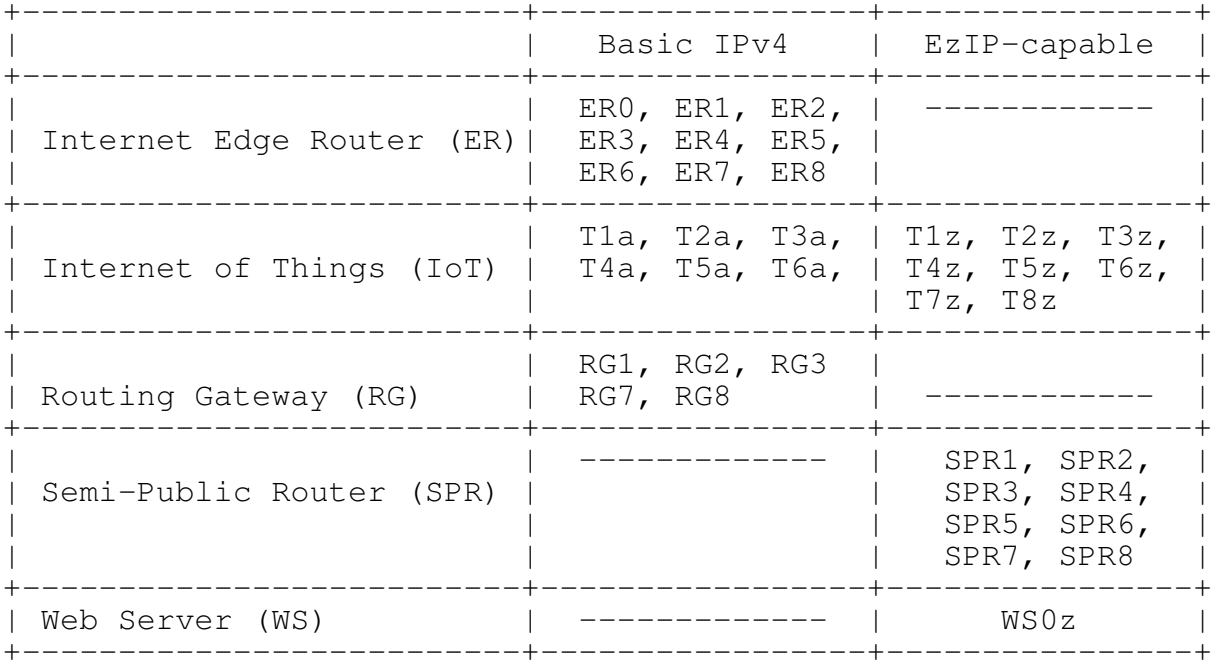


Figure 18 EzIP System Architecture-D (240/4 Portion)



Note: WS0z could be either a collection of conventional web servers connected to the Internet via a SPR, with message transfer capability among themselves, or a new web sever with multiple modules that recognize and re-direct packets depending on its header (conventional IP or EzIP) type. The main path functions the same as existing web servers. The secondary servers are on EzIP extension addresses that may be directly accessed by packets with EzIP header, or receive packets forwarded through the main module upon being qualified.

Figure 19 EzIP System Components

Appendix B EzIP Operation

To demonstrate how EzIP could support and enhance the Internet operations, the following are three connection examples that involve SPRs as shown in Figure 9. These present a general perspective of how IP header transitions through the routers may look like.

A. The first example is between EzIP-unaware IoTs, T1a and T4a. This operation is very much like the conventional TCP/IP packet transmission except with SPRs acting as an extra pair of routers supported by CGNAT. In addition, SPR4 may be viewed as a full-fledged RG minus DHCP and NAT support, because it assigns its IoTs with static addresses from the entire range of reserved 192.168/16, instead of the common much smaller pool of 192.168.nnn/24.

B. The second one is between EzIP-capable IoTs, T1z and T4z. Here, the SPRs process the extended public IP addresses in router mode, avoiding the delays due to the NAT type of operations.

C. The last one is between EzIP-unaware and EzIP-capable IoTs. By initiating and responding with a conventional IP header, T1z and T4z behave like an EzIP-unaware IoT. Thus, all packet exchanges use the conventional IP headers, just like case A. above.

B.1. Connection between EzIP-unaware IoTs

B.1.1. T1a Initiates a Session Request towards T4a

In Figure 20, T1a initiates a session request to SPR4 that serves T4a by sending an IP packet to RG1. There is no TCP port number in this IP header yet.

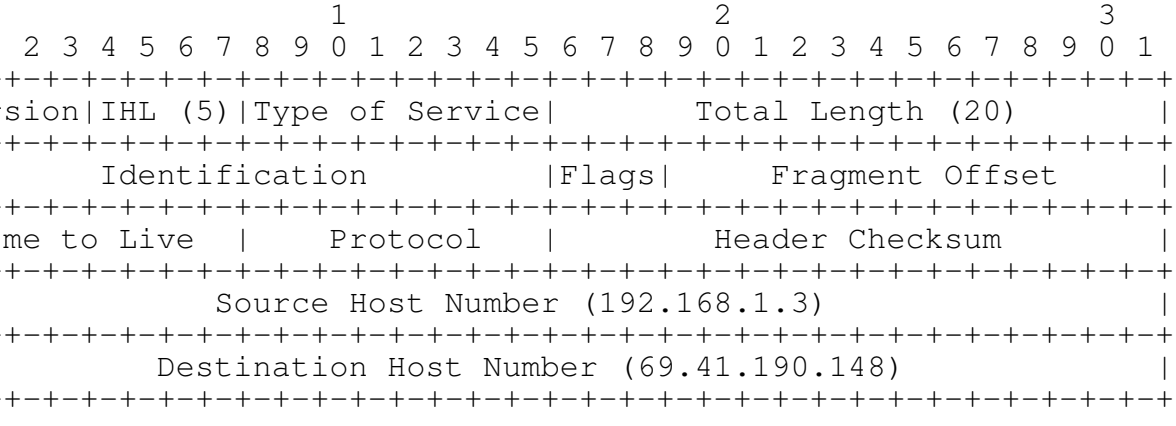


Figure 20 IP Header: From T1a to RG1

B.1.2. RG1 Forwards the Packet to SPR1

In Figure 21, RG1, allowing be masqueraded by T1a, relays the packet toward SPR1 by assigning the TCP Source port number, 3N, to T1a. Note that the suffix "N" denotes the actual TCP port number assigned by the RG1's NAT. This could assume multiple values, each represents a separate communications session that T1a is engaged in. A corresponding entry is created in the state table for handling the responding packet from the Destination site. Since T4a's TCP port number is not known yet, it is filled with all 1's.

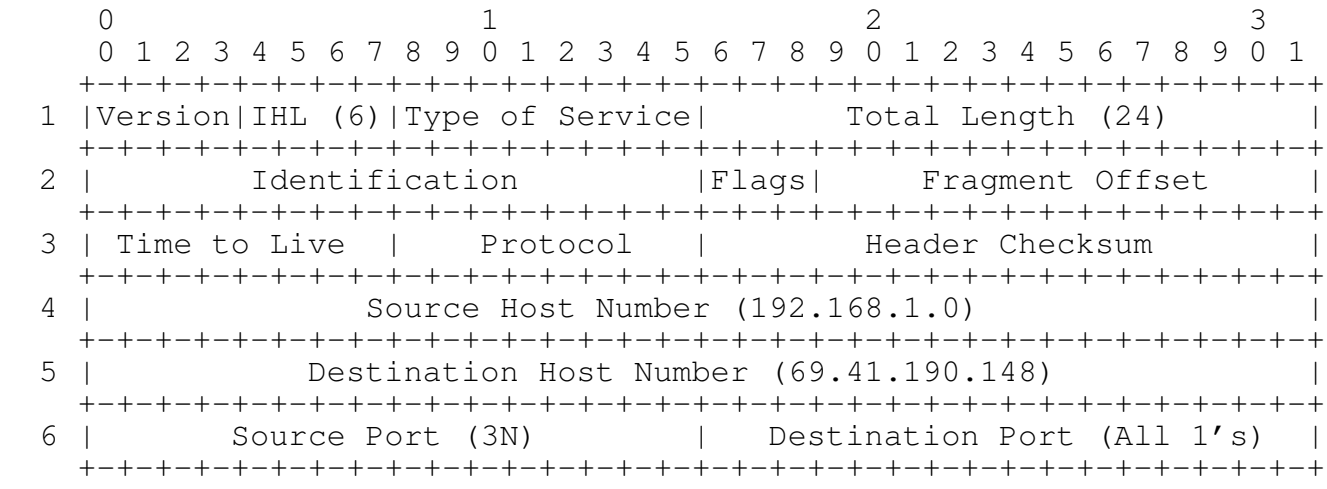


Figure 21 TCP/IP Header: From RG1 to SPR1

B.1.3. SPR1 Sends the Packet to SPR4 through the Internet

In Figure 22, SPR1 allowing masqueraded by RG1 (with the Source Host Number changed to be its own and the TCP port number changed to 1C, where "C" stands for CGNAT) sends the packet out through the Internet towards SPR4. The packet traverses through the Internet (ER1, CR and ER4) utilizing only the basic IP header portion of address information (words 4 & 5).

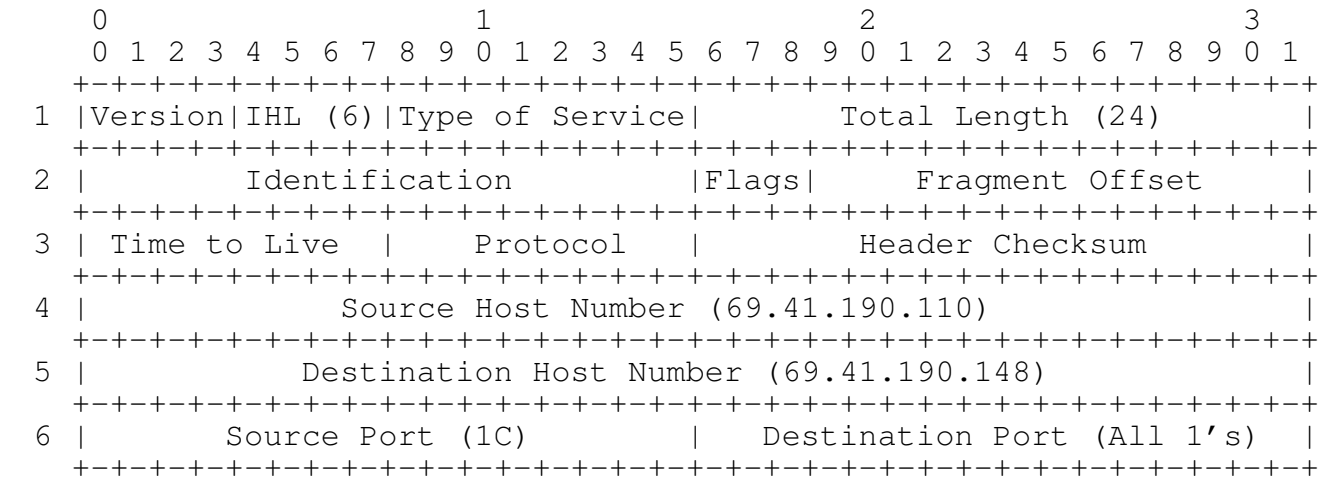


Figure 22 TCP/IP Header: From SPR1 to SPR4

B.1.4. SPR4 Sends the Packet to T4a

Since the packet has a conventional IP header without Destination TCP port number, SPR4 would ordinarily drop it due to the CGNAT function. However, for this example, let's assume that there exists a state-table that was set up by a DMZ process for redirecting this packet to T4a with a CGNAT TCP port number 410C (the composite of the third and the fourth octets, "4.10" of T4a's Extension No.). In Figure 23, SPR4 sends the packet to T4a by constructing the destination address accordingly.

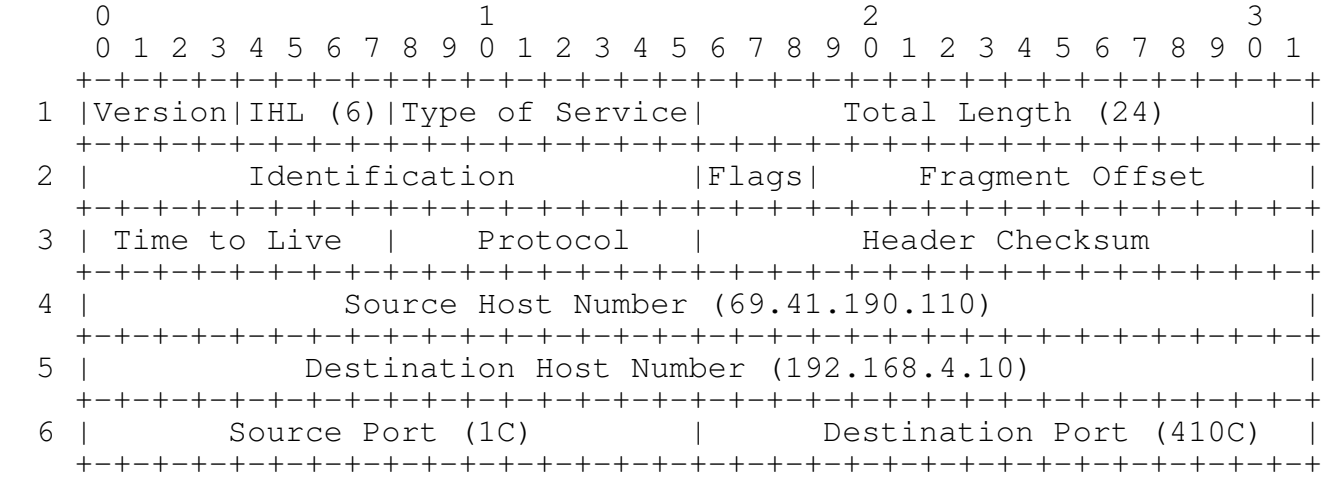


Figure 23 TCP/IP Header: From SPR4 to T4a

B.1.5. T4a Replies to SPR4

In Figure 24, when T4a replies to SPR4, it interchanges the Source and Destination identifications to create an IP header for the reply packet.

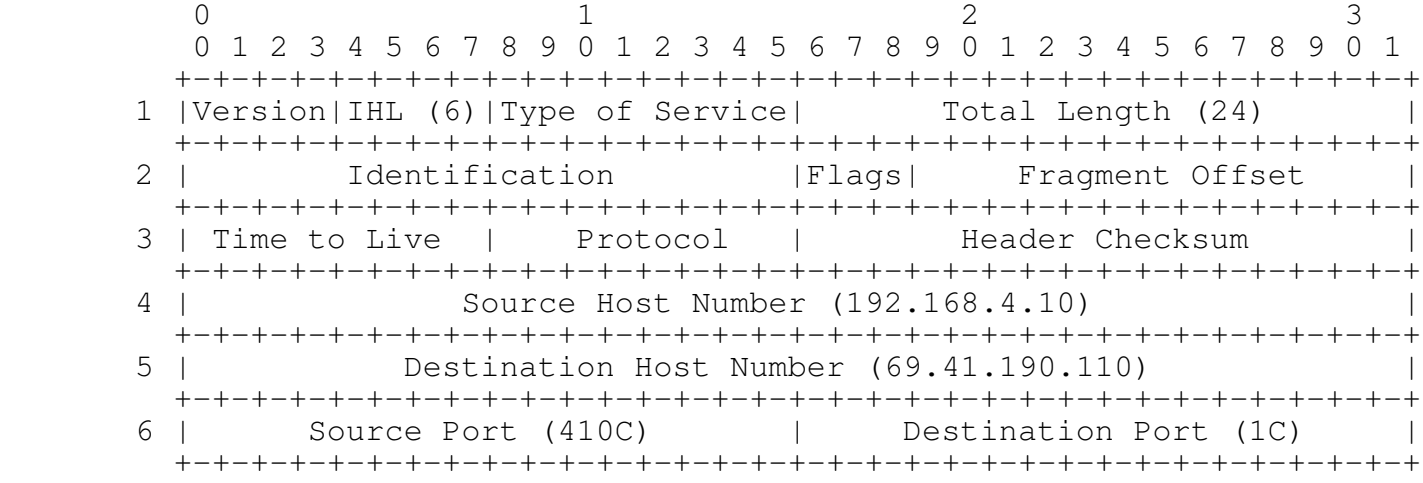


Figure 24 TCP/IP Header: From T4a to SPR4

B.1.6. SPR4 Sends the Packet to SPR1 through the Internet

In Figure 25, SPR4 sends the packet toward SPR1 with the following header through the Internet (ER4, CR and ER1) who will simply relay the packet according to the information in word 5 (Destination Host Number):

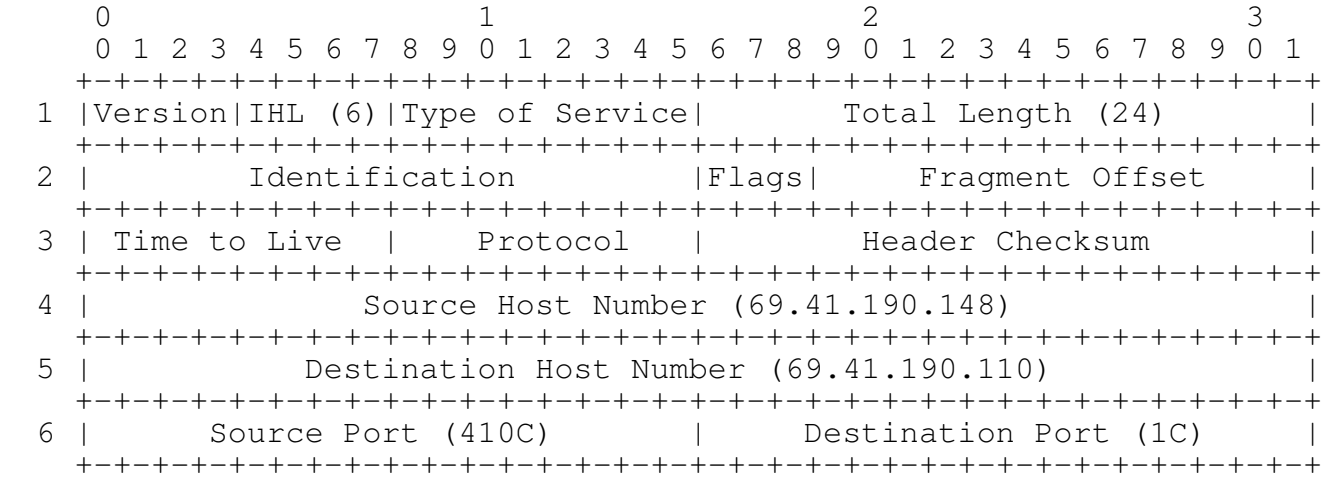


Figure 25 TCP/IP Header: From SPR4 to SPR1

B.1.7. SPR1 Sends the Packet to RG1

In Figure 26, RG1 address is reconstructed by using the information in the CGNAT state-table stored in SPR1.

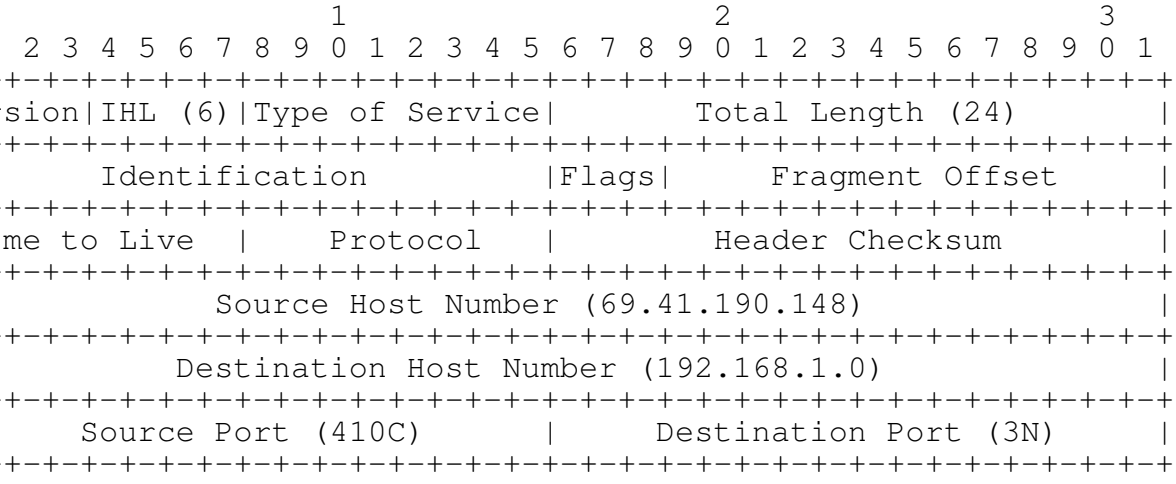


Figure 26 TCP/IP Header: From SPR1 to RG1

B.1.8. RG1 Forwards the Packet to T1a

In Figure 27, T1a address is reconstructed from that of RG1 and the state-table in the NAT based on Destination Port (3N).

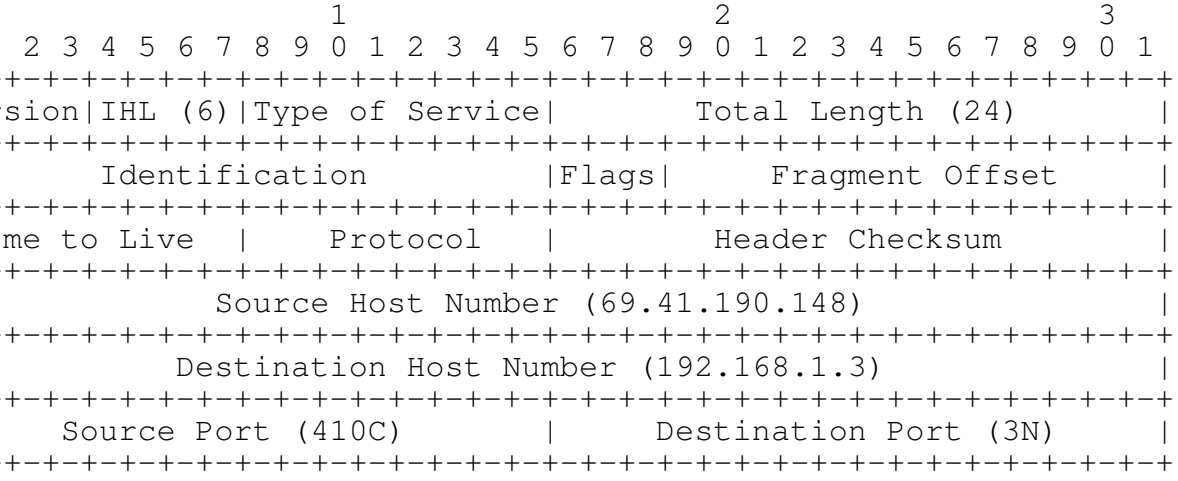


Figure 27 TCP/IP Header: From RG1 to T1a

B.1.9. T1a Sends a Follow-up Packet to RG1

To carry on the communication, T1a in Figure 28 sends the follow-up packet to RG1 with a full TCP/IP header.

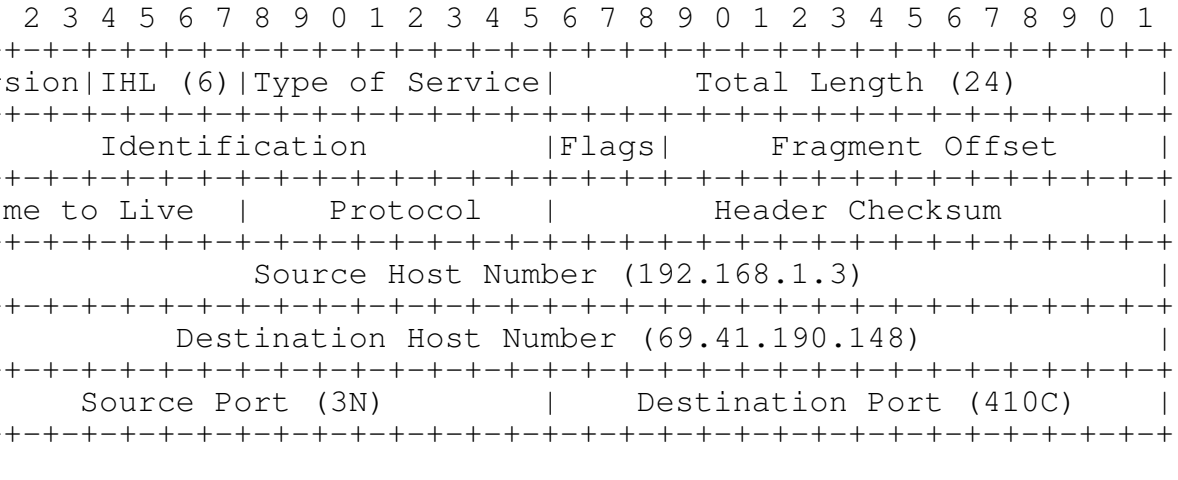


Figure 28 TCP/IP Header: Follow-up Packets From T1a to RG1

B.2. Connection Between EzIP-capable IoTs

The following is an example of EzIP operation between T1z and T4z shown in Figure 9. Each knows its own full "Public - EzIP : Private" network addresses, "69.41.190.110-192.168.1.0:9" and "69.41.190.148-192.168.4.40", respectively, as well as the other's. Note that T4z full address does not have the IoT No. portion. It is directly addressable from the Internet.

B.2.1. T1z Initiates a Session Request towards T4z

T1z initiates a session request to T4z by sending an EzIP packet to RG1. There is no TCP port number word, because T4z does not have such and that for T1z has not been assigned by the RG1's NAT.

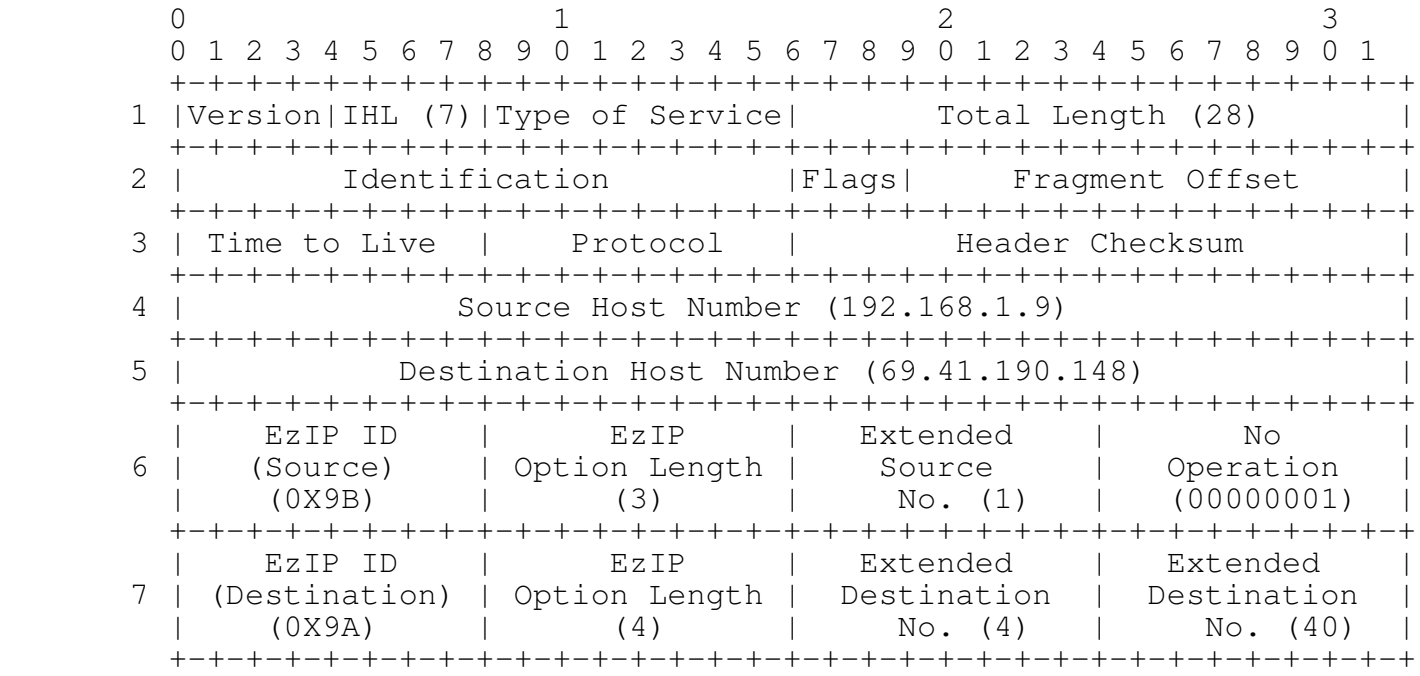


Figure 29 EzIP Header: From T1z to RG1

Note that 0X9A and 0X9B are temporarily selected from the available "IP Option Numbers" [10]. They were employed by prior efforts to facilitate the presentation of, EnIP and ExIP, respectively. These convey the concepts of transporting the value of the "Network No." as well as the number of octets needed in the "Extension No.". That is, both Option Numbers represent 192.168/16 as the EzIP Network No. prefix, while individually conveys two or one octets used in the Extension No., respectively.

B.2.2. RG1 Forwards the Packet to SPR1

In Figure 30, RG1, allowing to be masqueraded by T1z, relays the packet toward SPR1 by assigning the TCP Source port number, 9N, to T1z. Since T4z is directly connected to the Internet, there is no private network information to fill the Destination portion of the TCP word.

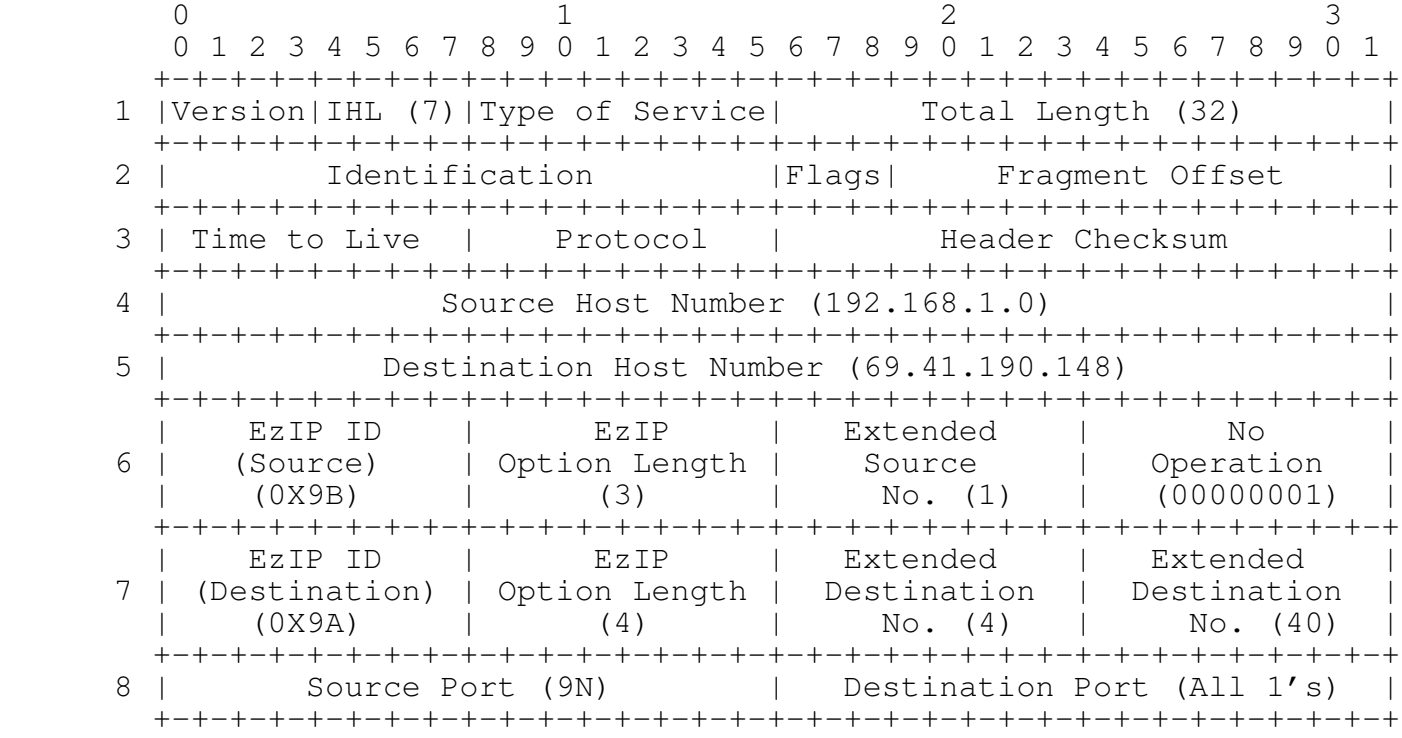


Figure 30 TCP/EzIP Header: From RG1 to SPR1

B.2.3. SPR1 Sends the Packet to SPR4 through the Internet

In Figure 31, SPR1 sends the packet out into the Internet towards SPR4. The packet traverses through the Internet (ER1, CR and ER4), utilizing only the basic IP header portion of address information. Note that the third octet of word 6 plus the first two octets of word 8 make up the subnet address of T1z. And, the last two octets of word 7 represent the extended address of T4z.

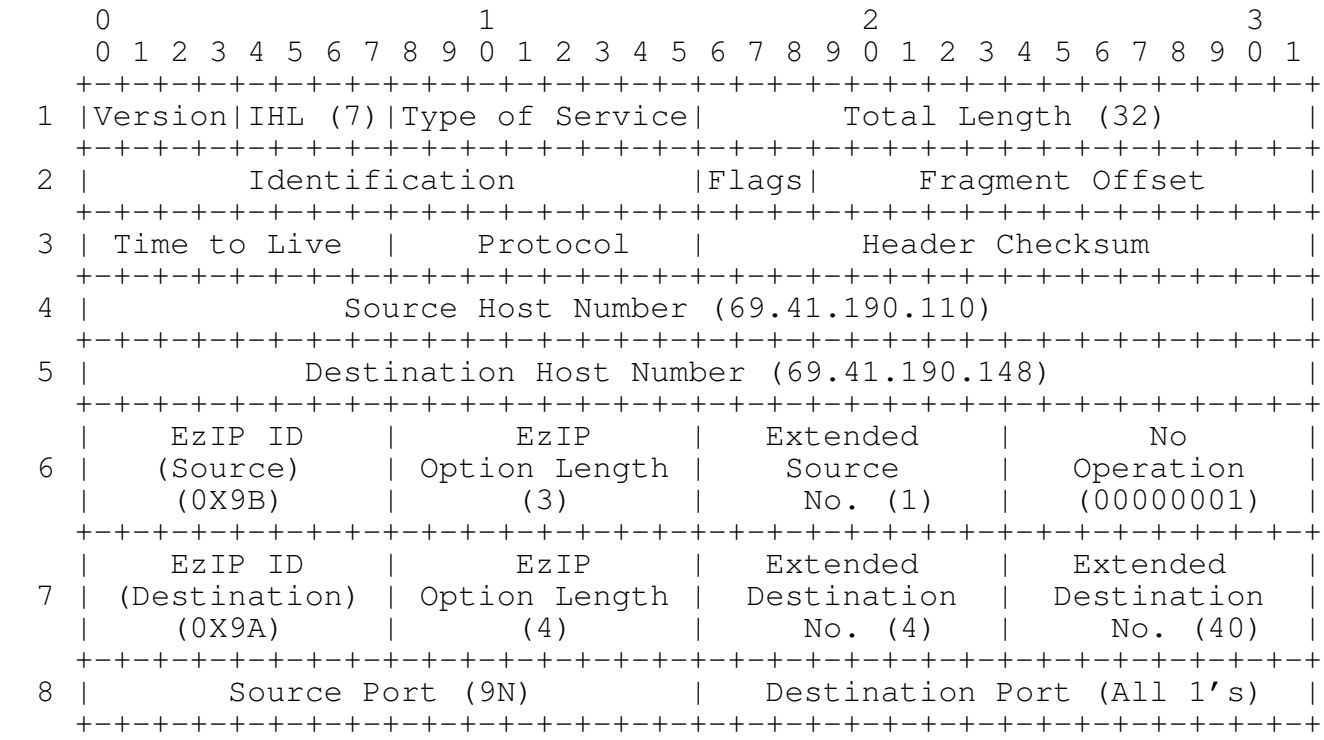


Figure 31 TCP/EzIP Header: From SPR1 to SPR4

B.2.4. SPR4 Sends the Packet towards T4z to RG2

In Figure 32, SPR4 sends the packet to RG2 by reconstructing its address from the Option number and the Extended Destination No.

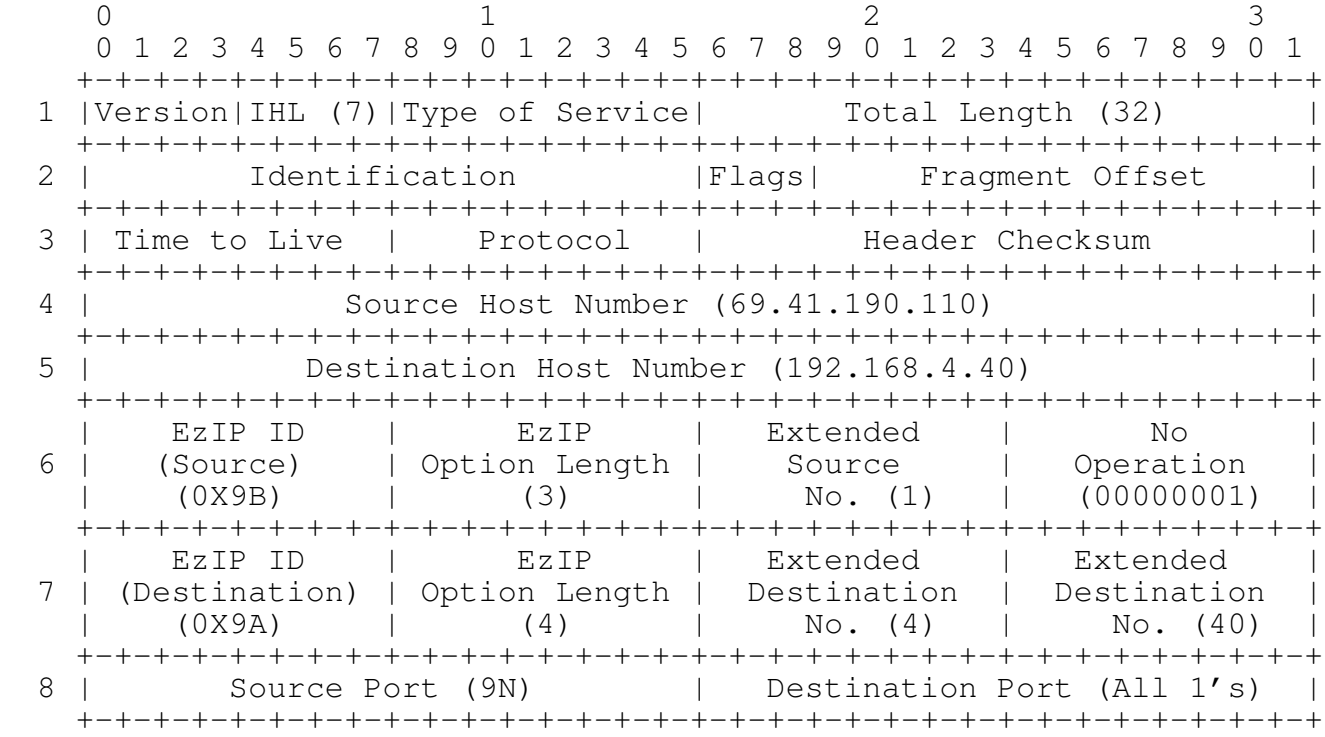


Figure 32 TCP/EzIP Header: From SPR4 to T4z

B.2.5. T4z Replies to SPR4

In Figure 33, T4z replies to SPR4 with the full T1z identification (69.41.190.110-192.68.1.0:192.168.1.9N conveyed by Option ID 0X9B together with the compact address string 69.41.190.110-1:9N) to create an EzIP header for the reply packet.

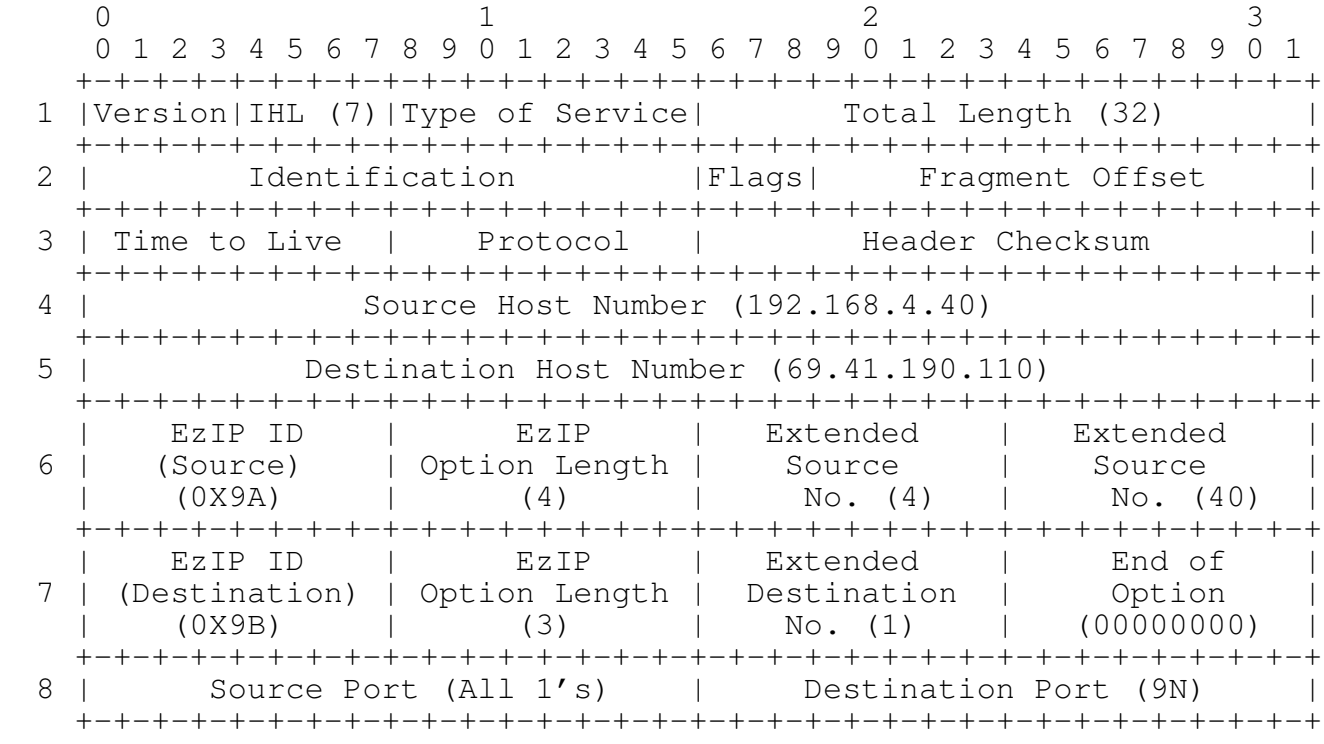


Figure 33 TCP/EzIP Header: From T4z to SPR4

B.2.6. SPR4 Sends the Packet to SPR1 through the Internet

In Figure 34, SPR4 sends the packet toward SPR1 with the following header through the Internet (ER2, CR, and ER1) who will simply relay the packet according to the information in word 5 (Destination Host Number):

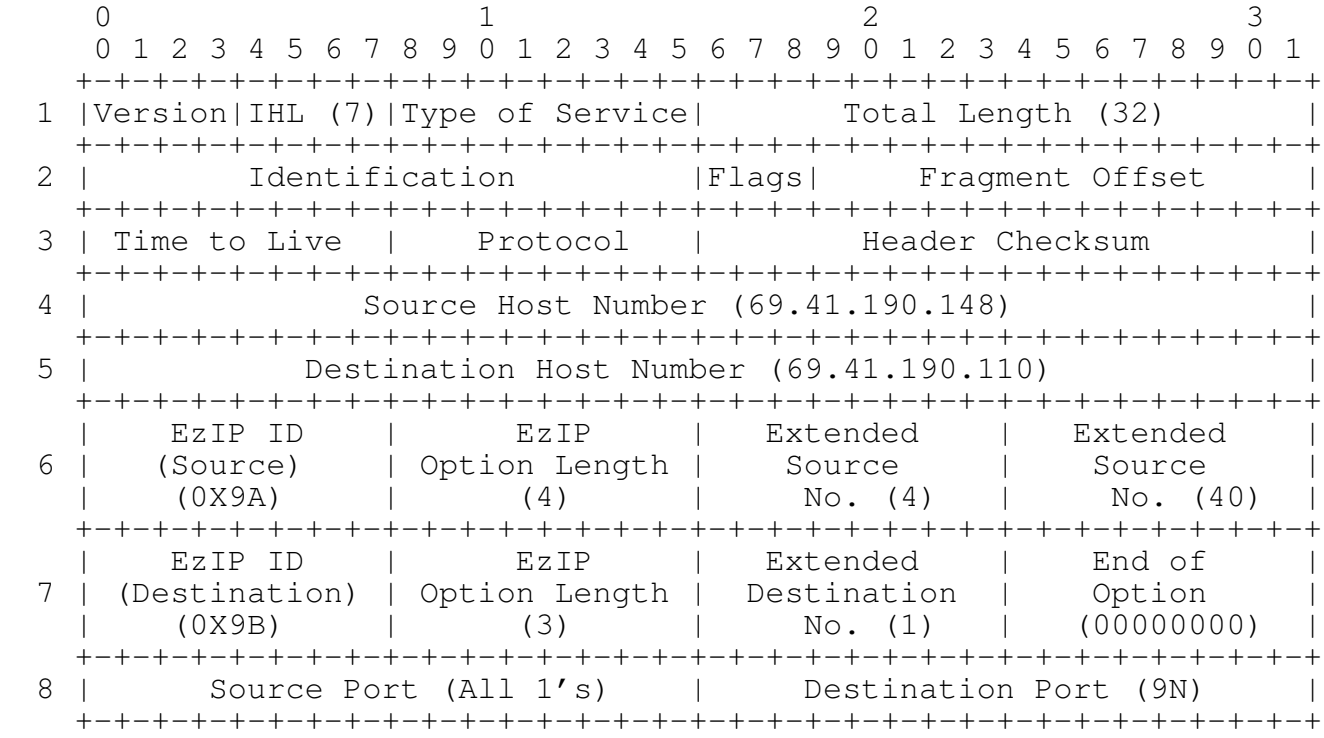


Figure 34 TCP/EzIP Header: From SPR4 to SPR1

B.2.7. SPR1 Sends the Packet to RG1

In Figure 35, RG1 address is reconstructed from the Option number and the Extended Destination No.

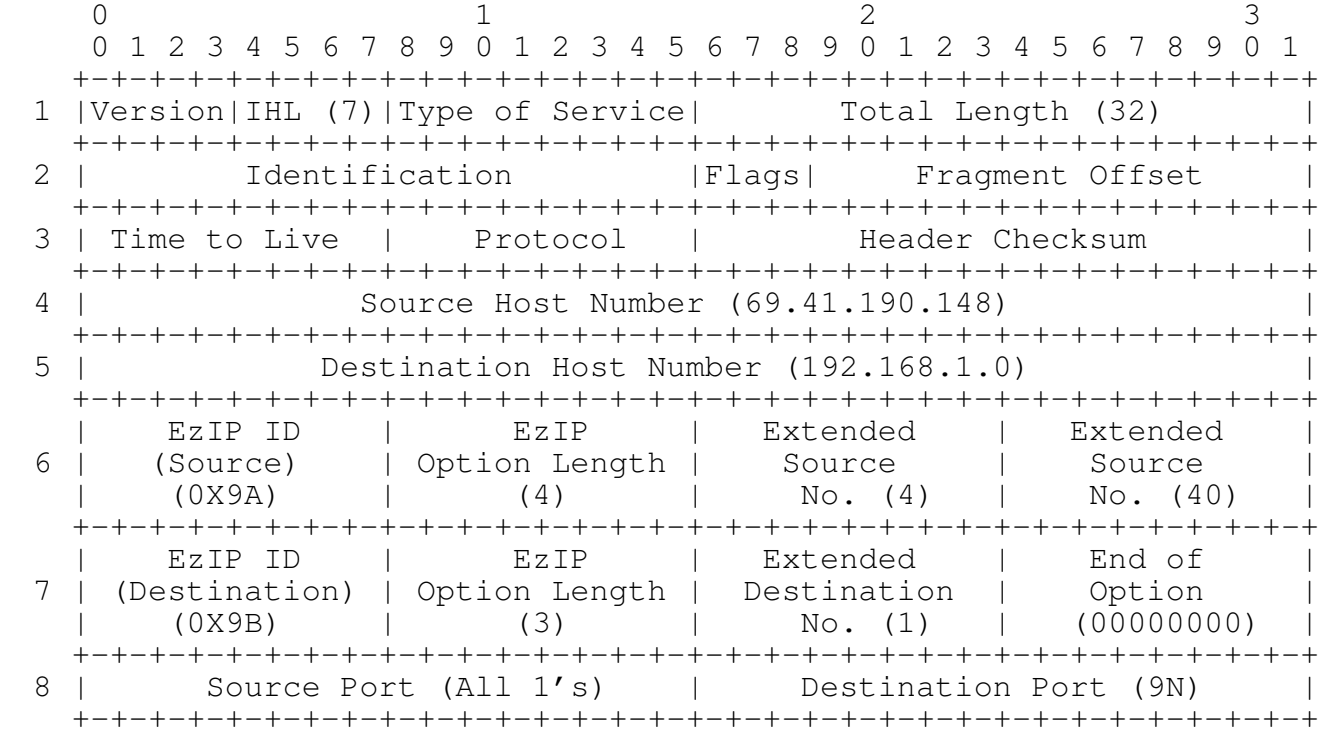


Figure 35 TCP/EzIP Header: From SPR1 to RG1

B.2.8. RG1 Forwards the Packet to T1z

In Figure 36, T1z address is reconstructed from that of RG1 and the NAT state-table based on Destination Port (9N).

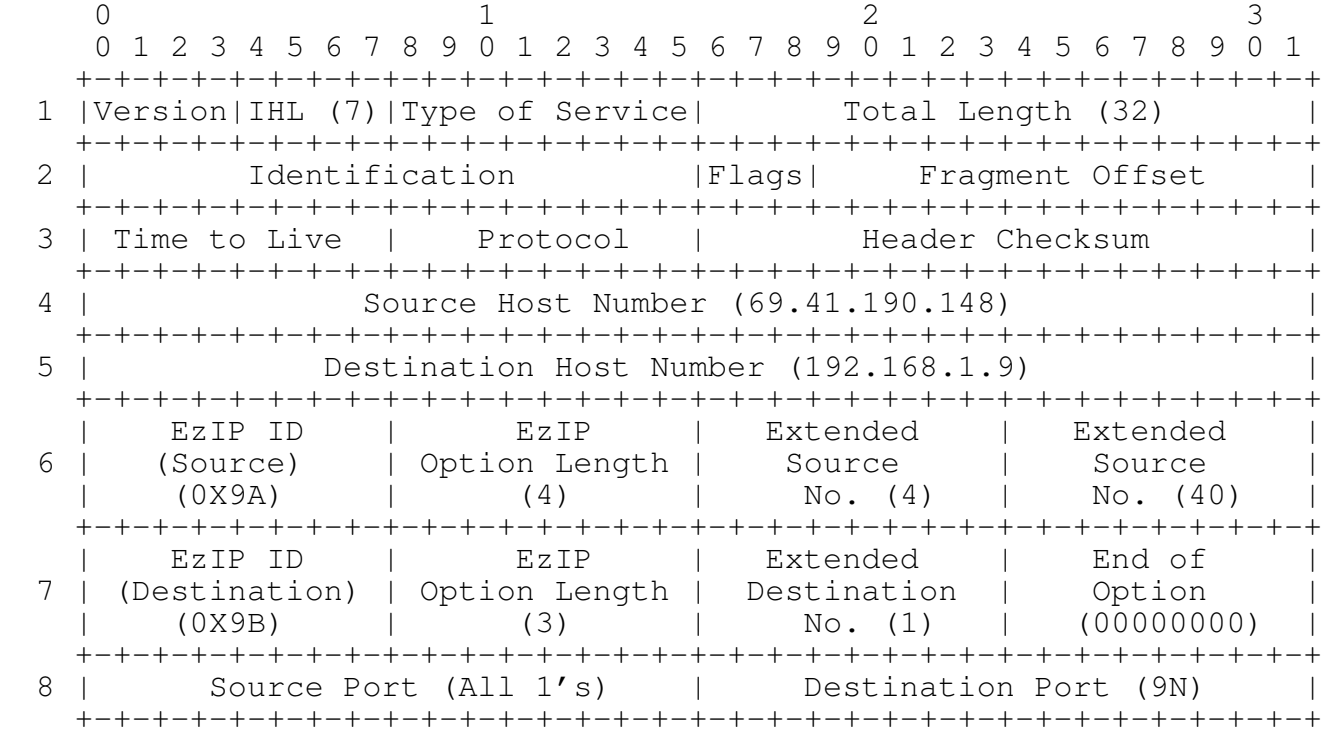


Figure 36 TCP/EzIP Header: From RG1 to T1z

B.2.9. T1z Sends a Follow-up Packet to RG1

In Figure 37, T1z sends a follow-up packet to RG1 with all fields filled with needed information.

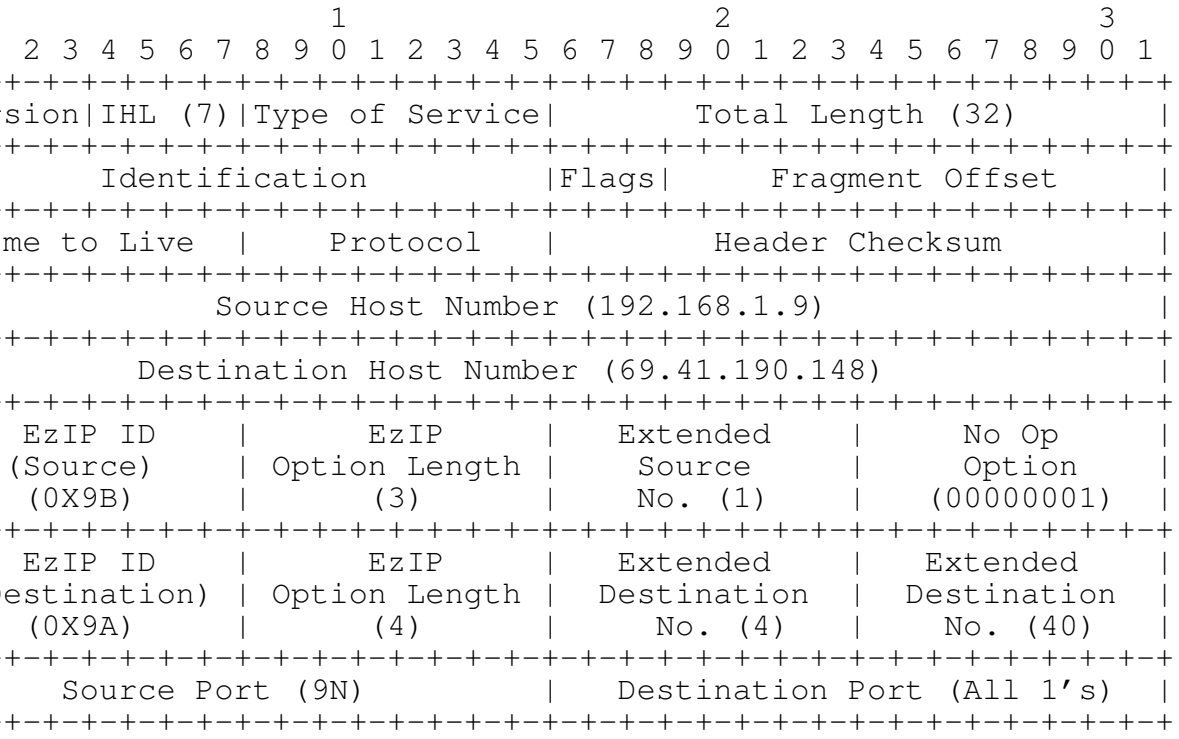


Figure 37 TCP/EzIP Header: Follow-up Packets from T1z to RG1

B.3. Connection Between EzIP-unaware and EzIP-capable IoTs

B.3.1. T1a initiates a request to T4z

Since T1a can create only IP header with conventional format, the SPRs will provide CGNAT type of services to the IP packets. And, assuming SPR4 has a state-table set up by DMZ for forwarding the request to T4z, the packet will be delivered to T4z. Seeing the incoming packet using conventional IP header, T4z should respond with the same so that the session will be conducted with conventional TCP/IP headers.

B.3.2. T1z initiates a request to T4a

Knowing T4a is not capable of EzIP header, T1z purposely initiates the request packet using conventional IP header. It will be treated by SPRs in the same manner as the T1a initiated case above and recognizable by T4a.

In brief, the steps outlined above are very much the same as the conventional TCP/IP header transitions between routers, except two extra steps in each direction are inserted to encode and decode the additional SPR provided EzIP routing process.

Note that when an IoT, such as T4a or T4z, is directly connected to a SPR, like SPR4, there is no RG in-between. There is no corresponding TCP port number in word 8 of the above TCP/EzIP headers. This spare facility in the header allows an RG be inserted if desired, thus re-establishing the private network environment.

When only its Extension No. portion of an EzIP extension address is transported in the EzIP header, the conventional private network address may be reused in this kind of added private networks. When extension address is transported by a full TCP/EzIP header with four octet format, proper precaution must be exercised to avoid confusing the routers along the way due to the appearance of a full private network address although at a location in the IP header not intended for ordinary IP address. When EzIP-7 is used, this is not of concern because the 240/4 block does not belong to the three conventional private network address blocks.

Appendix C Internet Transition Considerations

To enhance a large communication system like the Internet, it is important to minimize the disturbance to the existing equipments and processes due to any needed modification. The basic EzIP plan is to confine all actionable enhancements within the new SPR module. The following outlines the considerations for supporting the transition from the current Internet to the one enhanced by the EzIP technique.

C.1. EzIP Implementation

C.1.1. Introductory Phase:

A. Insert an SPR in front of a web-server that desires to have additional subnet addresses for offering diversified activities. For the long term, a new web server may be designed with these two functional modules combined.

. The first address of a private network address pool, e.g., 192.168.0.0, used by the SPR should be reserved as a DMZ (De-Militarized Zone) channel directing the initial incoming service requesting packets to the existing web server. This will maintain the same operation behavior projected to the general public.

. The additional addresses, up to 192.168.255.255 may be used for EzIP address extension purposes. Each may be assigned to an additional web server representing one of the business's new activities. Each of these new servers will then respond with EzIP header to messages forwarded from the main server, or be directly accessed through its EzIP address.

B. Insert an SPR in front of a group of subscribers who are to be served with the EzIP function. The basic service provided by this SPR will be the CGNAT equivalent function. This will maintain the same baseline user experience in accessing the Internet.

C. Session initiating packets with basic IPv4 header will be routed by SPRs to a business's existing server at the currently published IPv4 public address (discoverable by existing DNS). The server should respond with the basic IPv4 format as well. Essentially, this maintains the existing interaction between a user and a web server within an EzIP-unaware environment.

So far, neither the web-server nor any subscriber's IoTs needs to be enhanced, because the operations remain pretty much the same as today's common practice utilizing CGNAT assisted connectivity. See Appendix B.1. for an example.

D. Upon connected to the main web server, if a customer intentionally selects one of the new services offered by the primary web-server, the web-server will ask the customer to confirm the selection.

. If confirmed, implying that the customer is aware of the fact that his IoT is being served by an SPR, the web server forwards the request to a branch server for carrying on the communication via an EzIP address.

. The SPR at the originating side, recognizing the EzIP header from the web-server, replaces the CGNAT service with EzIP routing.

. For all subsequent packets exchanged, the EzIP headers will be used in either direction. See Appendix B.2. for an example. This will speed up the transmission throughput performance for the rest of the session.

C.1.2. New IoT Operation Modes:

A. EzIP-capable IoT will create EzIP header in initiating a session, to directly reach a specific web-server, instead of the lengthy steps of going through the DMZ port followed by manually making the selection from the main web server. This will speed up the initial handshake process. See Appendix B.2. for an example.

B. To communicate with an EzIP-unaware IoT, an EzIP-capable IoT should purposely initiate a session with conventional IP header. This will signal the SPRs to provide just CGNAT type of connection service. See Appendix B.3. for an example.

C.1.3. End-to-End Operation:

Once EzIP-capable IoTs become common for the general public, direct communication between any pair of such IoTs will be achievable. An EzIP-capable IoT, knowing the other IoT's full EzIP address, may initiate a session by creating an EzIP header that directs the SPRs to provide EzIP services, bypassing the CGNAT process. See Appendix B.2. for an example.

C.2. SPR Operation Logic

To support the above scenarios, the SPR should be designed with the following decision process:

C.2.1. Initiating a Session Request for an IoT or via a RG

If a session request IP packet contains EzIP Option word, it will be routed forward by SPR accordingly. Otherwise, the SPR provides CGNAT service by assigning a TCP port number to the packet and allowing the packet to masquerade with the SPR's own IP address while an entry to the state (port forward / look-up / hash) table is created in anticipation of the reply packet.

C.2.2. Receiving a Session Request from the ER

If a received IP packet includes a valid EzIP Option word or port number, SPR will utilize it to route the packet to an RG or an IoT. For a packet with plain IP header, it will be routed according to the Destination Host Number (IP header word 5).

C.3. RG Enhancement

With IPv4 address pool expanded by the EzIP schemes, there will be sufficient publicly assignable addresses for IoTs wishing to be directly accessible. The existing private networks may continue their current behavior of blocking session request packets from the Internet. In-between, another connection mode is possible. The following describes such an option in the context of the existing RG operation conventions.

C.3.1. Initiating Session request for an IoT

Without regard to whether the IP header is a conventional one or an EzIP type, a RG allows a packet to masquerade with the RG's own IP address by assigning a TCP port number to the packet and creating an entry to the state (port forward / look-up / hash) table. This is the same as current NAT practice.

C.3.2. Receiving a packet from the SPR

The "Destination Port" value in the packet is examined:

A. If it matches with an entry in the RG NAT's state-table, the packet is forward to the corresponding address. This is the same as the normal NAT processes in a conventional RG.

B. If it matches with the address of an active IoT on the private network, the packet is assigned with a TCP port number and then forwarded to that IoT.

Note that there is certain amount of increased security risk with this added last step, because a match between a guessed destination identity and the above two lists could happen by chance. To address

this issue, the following proactive mechanism may be incorporated in parallel:

If the “Destination Port” number is null or does not match with either of the above cases, the packet is dropped and an alarm state is activated to monitor for possible ill-intended follow-up attempts. A defensive mechanism should be triggered when the number of failed attempts has exceeded the preset threshold within a finite time interval.

In brief, if the IP header of a session requesting packet indicates that the sender knows the identity of the desired destination IoT on a private network, the common RG screening process will be bypassed. This facilitates the direct end-to-end connection, even in the presence of the NAT. Note that this process is very much the same as the AA (Automated Attendant) capability in a PABX telephone switching system that automatically makes the connection for a caller who indicates (via proper secondary dialing or the equivalent) knowing the extension number of the destination party. Such process can effectively screen out most of the unwanted callers.

Authors’ Addresses

Abraham Y. Chen
Avinta Communications, Inc.
142 N. Milpitas Blvd., #148, Milpitas, CA 95035-4401 US

Phone: _+1(408)942-1485
Email: AYChen@Avinta.com

Ramamurthy R. Ati
Avinta Communications, Inc.
142 N. Milpitas Blvd., #148, Milpitas, CA 95035-4401 US

Phone: _+1(408)458-7109
Email: rama_ati@outlook.com