

Appendice B

ESEMPI DI PDU

Questa appendice contiene una serie di pacchetti catturati da diverse reti locali con un analizzatore di protocollo Sniffer v.3.0 della Network General. L'analizzatore è stato programmato per fornire la decodifica dei vari protocolli a partire dai livelli più alti del modello di riferimento OSI, sino a giungere al livello 2 (Data Link), oltre che il pacchetto in esadecimale.

B.1 ESEMPIO DI DECODIFICA

La figura B.1 rappresenta un pacchetto LLC di tipo supervisory frame. Il pacchetto ha unicamente la busta 802.3 e la busta 802.2. I primi 6 byte del pacchetto contengono l'indirizzo di destinazione a livello MAC (MAC-DSAP) e i secondi 6 byte l'indirizzo di mittente a livello MAC (MAC-SSAP).

I due byte che seguono possono essere interpretati come protocol type (nel caso che il pacchetto sia Ethernet, e allora il valore contenuto deve essere maggiore di 1500) oppure come length (nel caso che il pacchetto sia 802.3, e allora il valore contenuto deve essere minore uguale a 1500). In questo caso il valore è 4 quindi si tratta di un pacchetto 802.3 e il significato è length.

Segue il pacchetto LCC che ha LLC-DSAP e LLC-SSAP uguali a 4. Tale valore indica dei SAP LLC di tipo user defined ed in particolare la codifica 4 indica che si tratta di un pacchetto di SNA Path Control.

Segue il campo control che nei pacchetti di tipo supervisory è su due byte (se si fosse trattato di un pacchetto unnumbered sarebbe stato su un byte solo).

Non vi sono dati trattandosi di un supervisory frame.

Il pacchetto non raggiunge la dimensione minima di 64 byte richiesta per la

trasmissione su 802.3 e quindi vengono aggiunti 42 byte di padding.

Analizzando più nel dettaglio il campo control, vediamo che si tratta di un supervisory frame di sottotipo receiver ready (RR), che trasporta un NAK per il pacchetto numero 92.

DLC: ----- DLC Header -----
DLC:
DLC: Frame 1 arrived at 17:46:30.5814; frame size is 60 (003C hex) bytes.
DLC: Destination = Station IBM 1A60BE
DLC: Source = Station 020000000040
DLC: 802.3 length = 4
DLC:

Busta 802.3

LLC: ----- LLC Header -----
LLC:
LLC: DSAP = 04, SSAP = 04, Command, Supervisory frame: RR, N(R) = 92, POLL
LLC:
DLC: Frame padding= 42 bytes

Busta 802.2

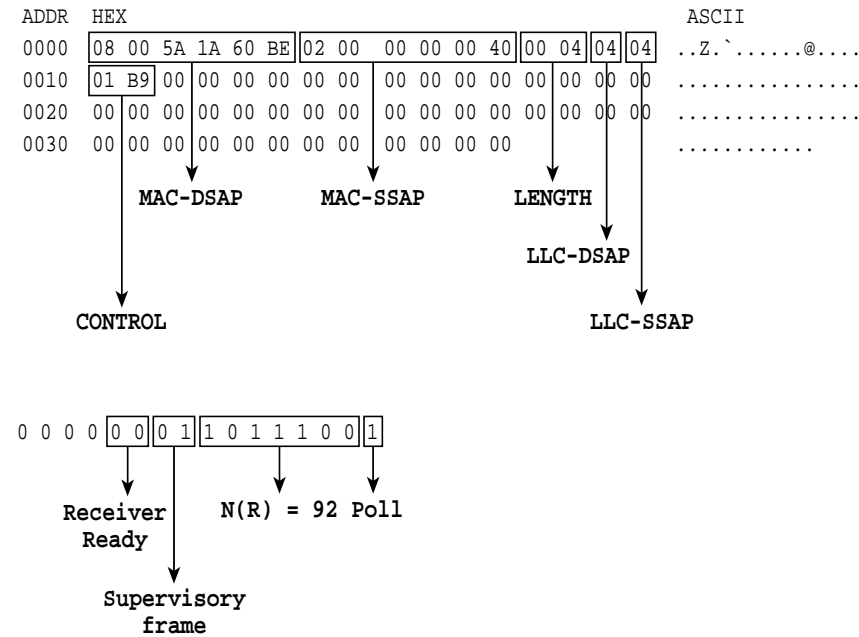


Fig. B.1 - Esempio di decodifica.

B.2 IBM

In questo sottoparagrafo sono riportati alcuni pacchetti appartenenti all'architettura di rete IBM SNA.

B.2.1 SNA

Il pacchetto seguente è un pacchetto di bind di una sessione SNA di tipo LU 6.2, imbustato in un pacchetto SNA path control, imbustato in un pacchetto LLC di tipo information (LLC connesso o di tipo 2), imbustato in un pacchetto 802.3 con un campo dati di 94 byte.

```
DLC: Destination = Station IBM 1A60BE
DLC: Source      = Station 020000000040
DLC: 802.3 length = 94
DLC:
LLC: ----- LLC Header -----
LLC:
LLC: DSAP = 04, SSAP = 04, Command, I frame, N(R) = 92, N(S) = 34
LLC:
SNA: ----- SNA Transmission Header -----
SNA:
SNA: Format identification (FID) = 2
SNA:
SNA: Header flags = 2F
SNA:   0010 .... = Format identification
SNA:   .... 11.. = Only segment
SNA:   .... ..1. = Address field negotiation flag
SNA:   .... ...1 = Expedited flow
SNA: Destination address = 01
SNA: Origin      address = 01
SNA: Sequence number = 0
SNA:
SNA: ----- SNA Request Header (RH) -----
SNA:
SNA: RH byte 0          = 6B
SNA:   0... .... = Command
SNA:   .11. .... = RU category is 'session control'
SNA:   .... 1... = FM or NS header follows
SNA:   .... .0.. = Sense data not included
SNA:   .... ..11 = Only RU in chain
SNA: RH byte 1          = 80
SNA:   1.00 .... = Definite response requested
SNA:   .... ..0. = Response bypasses TC queues
SNA:   .... ...0 = Pacing indicator
```

```

SNA: RH byte 2          = 00
SNA:      0... .. = Begin bracket indicator
SNA:      .0... .. = End bracket indicator
SNA:      .... ...0 = Conditional end bracket indicator
SNA:      ..0. .... = Change direction indicator
SNA:      .... 0... = Character code selection indicator
SNA:      .... .0.. = Enciphered data indicator
SNA:      .... ..0. = Padded data indicator
SNA:
SNA: ----- SNA SC-RU (Session Control) -----
SNA:
SNA: SC code = 31 (BIND: Bind Session)
SNA: Type = 0 (Negotiable)
SNA: FM profile = 19
SNA: TS profile = 7
SNA: Primary flags = B0
SNA:      1... .. = Multiple RU chains allowed
SNA:      .0... .. = Immediate request mode
SNA:      ..11 .... = Definite or exception chain response
SNA:      .... 0... = 2-phase commit not supported
SNA:      .... ..0. = Compression will not be used
SNA:      .... ...0 = Will not send end bracket
SNA: Secondary flags = B0
SNA:      1... .. = Multiple RU chains allowed
SNA:      .0... .. = Immediate request mode
SNA:      ..11 .... = Definite or exception chain response
SNA:      .... 0... = 2-phase commit not supported
SNA:      .... ..0. = Compression will not be used
SNA:      .... ...0 = Will not send end bracket
SNA: Common flags #1 = D1
SNA:      1... .. = Receipt of segments not supported
SNA:      .1... .. = FM headers allowed
SNA:      ..0. .... = Brackets are not used or reset states are INB
SNA:      ...1 .... = Conditional bracket termination will be used
SNA:      .... 0... = Alternate code set will not be used
SNA:      .... .0.. = Sequence numbers not available for sync points
SNA:      .... ..0. = BIS not sent
SNA:      .... ...1 = BIND can be queued
SNA: Common flags #2 = B1
SNA:      10... .. = Half-duplex flip-flop
SNA:      ..1. .... = Symmetric responsibility for recovery
SNA:      ...1 .... = Primary is contention winner
SNA:      .... 00.. = Process alternate code as ASCII-7
SNA:      .... ..0. = Control vectors are not included after the SLU
                        name
SNA:      .... ...1 = HDX-FF reset state is SEND for the primary
SNA: Secondary send window size = 1 (One-stage pacing)
SNA: Secondary receive window size = 0 (Adaptive pacing not
                        supported)

```

```

SNA: Maximum RU size sent by secondary LU = 480
SNA: Maximum RU size sent by primary LU = 480
SNA: Primary send window size = 0 (Two-stage pacing)
SNA: Primary receive window size = 1
SNA: LU type = 6
SNA: LU level = 2
SNA: PS flags #1 = 10
SNA:   ...1 .... = Access security info field will be accepted
SNA:   .... ..0. = Already verified indicator will not be accepted
SNA: PS flags #2 = 23
SNA:   .01. .... = Synchronization confirm supported
SNA:   .... 00.. = Operator controlled session reinitiation
SNA:   .... ..1. = Parallel session supported
SNA:   .... ...1 = Change number of sessions GDS variable flow
                  supported
SNA: PS Flags #3 = 0X
SNA:   .0.. .... = Contention-winner will not deactivate session
SNA: Cryptography options = 0X
SNA:   00.. .... = Private cryptography not supported
SNA:   ..00 .... = Session-level cryptography not supported
SNA:   .... 0000 = Length of cryptography options = 0
SNA: Primary LU name = "....K.....@@@"
SNA: Length of user data = 29
SNA: User data key = 0 (Structured subfields follow)
SNA: Structured data subfield number = 2 (Mode name)
SNA: Mode name = "....."
SNA: Structured data subfield number = 3 (Session instance
                  identifier)
SNA: Format = 0
SNA: Session instance identifier = 07
SNA: Structured data subfield number = 4 (Network-qualified PLU
                  network name)
SNA: Network-qualified PLU network name = "....K.....@@@"
SNA: Length of user request correlation field = 0
SNA: Length of secondary LU name = 8
SNA: Secondary LU name = ".....@"
SNA:

```

ADDR	HEX	ASCII
0000	08 00 5A 1A 60 BE 02 00 00 00 00 40 00 5E 04 04	..Z.`.....@.^..
0010	44 B8 2F 00 01 01 00 00 6B 80 00 31 00 13 07 B0	D./.....k..1....
0020	B0 D1 B1 01 00 F5 F5 80 01 06 02 00 00 00 00 00
0030	00 00 10 23 00 00 0D C1 D7 D7 D5 4B D7 C3 D4 C7	...#.....K....
0040	F1 40 40 40 1D 00 08 02 D8 D7 C3 E2 E4 D7 D7 03	.@@@.....
0050	03 00 07 0E 04 C1 D7 D7 D5 4B D7 C3 D4 C7 F1 40K.....@
0060	40 40 00 08 D7 D6 D3 C9 C1 E2 40 40	@@.....@@

B.2.2 IBM Netbios

Il pacchetto seguente è un pacchetto SMB (Server Message Block) generato probabilmente da un personal computer, imbustato nel protocollo Netbios IBM (Netbeui), imbustato in un pacchetto LLC information (LLC connesso o di tipo 2), imbustato in un pacchetto 802.3 generato da un elaboratore con indirizzo MAC 02-60-8C-74-11-78 (OUI = 02-60-8C, cioè 3Com). Il pacchetto SMB richiede l'apertura di un file remoto. La codifica del SAP LLC user defined del Netbios IBM è F0. Il pacchetto LLC ha numero di sequenza 109 e porta un NAK per piggybacking del frame numero 29.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 1 arrived at 09:18:37.3543; frame size is 143 (008F hex)
    bytes.
DLC: Destination = Station 3Com 676974
DLC: Source       = Station 3Com 741178
DLC: 802.3 length = 129
DLC:
LLC: ----- LLC Header -----
LLC:
LLC: DSAP = F0, SSAP = F0, Command, I frame, N(R) = 29, N(S) = 109
LLC:
NETB: ----- NETBIOS Data Only Last -----
NETB:
NETB: Header length = 14, Data length = 111
NETB: Delimiter = EFFF (NETBIOS)
NETB: Command = 16
NETB: Flags = X0
NETB: .... 0... = No Acknowledge_Included
NETB: .... .0.. = No Ack_with_data_allowed
NETB: .... ..0. = No NO.ACK indicator
NETB: Re-synch indicator = 0
NETB: Response correlator = 166B
NETB: Remote session number = 25
NETB: Local session number = 1
NETB:
SMB: ----- SMB Open & more Command -----
SMB:
SMB: Function = 2D (Open & more)
SMB: Tree id   (TID) = 0818
SMB: Process id (PID) = 0005
SMB: Multiplex id (MID) = 0794
SMB: File pathname = "\\3COM\\MSBENCH\\SYNC.CMD"
SMB: Additional information = 0001
SMB: .... .... .... .0.. = Notify about another file open
SMB: .... .... .... ..0. = Do not lock file
SMB: .... .... .... ...1 = Return additional information
```

```

SMB: File open mode = 0040
SMB:  .0.. .... = Not write through mode
SMB:  .... .... .100 .... = Shared open (allow others to read/write/
                           execute)
SMB:  .... .... .... .000 = Open file for reading
SMB: Search attributes = 0016
SMB:  .... .... ..0. .... = File(s) not changed since last archive
SMB:  .... .... ...1 .... = Directory file(s)
SMB:  .... .... .... 0... = No volume label info
SMB:  .... .... .... .1.. = System file(s)
SMB:  .... .... .... ..1. = Hidden file(s)
SMB:  .... .... .... ...0 = No read only file(s)
SMB: Attribute flags = 0020
SMB:  .... .... ..1. .... = File(s) changed and not archived
SMB:  .... .... ...0 .... = No directory file(s)
SMB:  .... .... .... 0... = No volume label info
SMB:  .... .... .... .0.. = No system file(s)
SMB:  .... .... .... ..0. = No hidden file(s)
SMB:  .... .... .... ...0 = No read only file(s)
SMB: Creation date: none supplied
SMB: Open function = 0001
SMB:  .... .... ...0 .... = If file doesn't exist, fail
SMB:  .... .... .... ..01 = If file exists, open it
SMB: Bytes to reserve on create or truncate = 0
SMB: Time to wait for completion: default
SMB:
SMB: ----- SMB Read & more Command -----
SMB:
SMB: Function = 2E (Read & more)
SMB: File handle = 0000
SMB: Offset in file = 0
SMB: Maximum read count = 4096 bytes
SMB: Minimum read count = 0 bytes
SMB: Time to wait for completion: no delay
SMB: Bytes remaining to read = 0
SMB:
SMB: ----- End of SMB chain -----
SMB:

```

ADDR	HEX	ASCII
0000	02 60 8C 67 69 74 02 60 8C 74 11 78 00 81 F0 F0	`.`.git.`.t.x....
0010	DA 3A 0E 00 FF EF 16 00 00 00 00 00 6B 16 19 01	.:.....k...
0020	FF 53 4D 42 2D 00 00 00 00 08 00 00 00 00 00 00	.SMB-.....
0030	00 00 00 00 00 00 00 00 18 08 05 00 00 00 94 07
0040	0F 2E 00 58 00 01 00 40 00 16 00 20 00 00 00 00	...X...@... ..
0050	00 01 00 00 00 00 00 FE FF FF FF 00 00 00 00 17
0060	00 5C 33 43 4F 4D 5C 4D 53 42 45 4E 43 48 5C 53	.\3COM\MSBENCH\S
0070	59 4E 43 2E 43 4D 44 00 0A FF 00 00 00 00 00 00	YNC.CMD.....
0080	00 00 00 00 10 00 00 00 00 00 00 00 00 00 00

B.2.3 IBM Network Management

Il pacchetto seguente è un pacchetto di network management (LLC-DSAP = F4). Si tratta di un unnumbered frame cui corrisponde un campo LLC control di un byte contenente il valore 03. Il pacchetto è trasmesso in multicast a livello MAC all'indirizzo 03-00-00-00-00-10.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 45 arrived at 00:03:39.1305; frame size is 109 (006D hex)
      bytes.
DLC: Destination = Multicast 030000000010
DLC: Source       = Station 02000000805E
DLC: 802.3 length = 95
DLC:
LLC: ----- LLC Header -----
LLC:
LLC: DSAP = F4, SSAP = E4, Command, Unnumbered frame: UI
LLC:
IBNM: ----- IBM Network Management -----
IBNM:
IBNM: Unknown major vector code point = 9006
IBNM:   Unknown complex subvector code point = 8001
IBNM:   Ring number = 000
IBNM:   (Warning: Next vector should be length 6 but is 10)
IBNM:   Number of alternate LAN Managers = 49152
IBNM:   (Warning: Next vector should be length 6 but is 8)
IBNM:   Ring status = 0000 (Operational)
IBNM:   MAC address = IBM 000000
IBNM:   Port Information
IBNM:   (Warning: Next vector should be length 12 but is 6)
IBNM:   Password key = ""
IBNM:   Unknown atomic vector type C00C
IBNM:   Status code = 0000 (Control lost)
IBNM:   Reason code = 0000 (Normal termination)
IBNM:   Unknown atomic vector type C00F
IBNM:   (Warning: Next vector should be length 14 but is 8)
IBNM:   Microcode level = ""
IBNM:   (Warning: Next vector should be length 10 but is 8)
IBNM:   Status = 00000010
IBNM:
IBNM: [Normal end of "IBM Network Management".]
IBNM:
```

ADDR	HEX	ASCII
0000	03 00 00 00 00 10 02 00 00 00 80 5E 00 5F F4 E4^._..
0010	03 00 5C 90 06 00 26 80 01 00 06 40 05 00 00 00	..\...&....@....
0020	0A C0 04 C0 00 00 00 00 00 00 08 C0 05 00 00 00


```

0030  00 00 0A 40 02 10 00 5A 00 00 00 00 32 80 03 00  ...@...Z....2...
0040  06 C0 0B 00 00 00 06 C0 0C 00 00 00 06 C0 0D 00  .....
0050  00 00 06 C0 0E 00 00 00 06 C0 0F 00 00 00 08 C0  .....
0060  10 00 00 00 01 00 08 C0 11 00 00 00 10  .....

```

B.3 TCP/IP

I pacchetti riportati nei seguenti sottoparagrafi appartengono tutti all'architettura di rete TCP/IP che comprende non solo i protocolli TCP e IP, ma anche molti altri quali UDP, ARP, BOOTP, ecc., e che più in generale andrebbe definita "internet technology".

B.3.1 Telnet

Il seguente pacchetto è generato dall'applicazione telnet che fornisce la funzionalità di terminale virtuale di un host remoto. Il pacchetto telnet è imbustato nel protocollo TCP, imbustato nel protocollo IP, imbustato in un pacchetto MAC di tipo Ethernet. Infatti i due byte che seguono i MAC address contengono il valore esadecimale 0800 che è maggiore di 1500 e quindi indica un pacchetto Ethernet ed assume il significato di protocol type (800 è il protocol type dell'IP). Il pacchetto ha due indirizzi MAC di tipo singolo, uno corrispondente ad una scheda IBM (OUI = 08-00-5A) e l'altra ad una scheda su cui opera anche il protocollo DECnet fase IV (OUI = AA-00-04). Che il pacchetto provenga da un'applicazione telnet risulta evidente poiché la source port nell'header TCP è 23, che è quella associata al telnet.

```

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 2 arrived at  16:55:17.7676; frame size is 60 (003C hex)
      bytes.
DLC:  Destination = Station DECnet009A7D
DLC:  Source      = Station IBM   1A60BE
DLC:  Ethertype   = 0800 (IP)
DLC:
IP:    ----- IP Header -----
IP:
IP:  Version = 4, header length = 20 bytes
IP:  Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:  Total length = 44 bytes

```

```

IP:  Identification = 22755
IP:  Flags = 0X
IP:  .0.. .... = may fragment
IP:  ..0. .... = last fragment
IP:  Fragment offset = 0 bytes
IP:  Time to live = 60 seconds/hops
IP:  Protocol = 6 (TCP)
IP:  Header checksum = 1853 (correct)
IP:  Source address = [130.192.4.18]
IP:  Destination address = [130.192.4.4]
IP:  No options
IP:
TCP:  ----- TCP header -----
TCP:
TCP:  Source port = 1044
TCP:  Destination port = 23 (Telnet)
TCP:  Initial sequence number = 866657824
TCP:  Acknowledgment number = 1449600001
TCP:  Data offset = 24 bytes
TCP:  Flags = 12
TCP:  ..0. .... = (No urgent pointer)
TCP:  ...1 .... = Acknowledgment
TCP:  .... 0... = (No push)
TCP:  .... .0.. = (No reset)
TCP:  .... ..1. = SYN
TCP:  .... ...0 = (No FIN)
TCP:  Window = 16384
TCP:  Checksum = 722D (correct)
TCP:
TCP:  Options follow
TCP:  Maximum segment size = 1451
TCP:

```

ADDR	HEX	ASCII
0000	AA 00 04 00 9A 7D 08 00 5A 1A 60 BE 08 00 45 00}..Z.`...E.
0010	00 2C 58 E3 00 00 3C 06 18 53 82 C0 04 12 82 C0	.,X...<..S.....
0020	04 04 04 14 00 17 33 A8 26 20 56 67 24 01 60 123.& Vg\$.`.
0030	40 00 72 2D 00 00 02 04 05 AB 00 00	@.r-.....

B.3.2 Bootp

Bootp è un protocollo per il bootstrap di stazioni diskless. La richiesta di downloading del sistema operativo viene inviata in broadcast (MAC-DSAP = FF-FF-FF-FF-FF-FF). Il pacchetto bootp viene imbustato in un pacchetto UDP, che viene imbustato in un pacchetto IP, che viene imbustato in un pacchetto Ethernet. Nell'header

IP è indicato che il protocollo sovrastante è UDP (Protocol = 17), mentre nell'header UDP è indicato che si tratta dell'applicativo Bootp (Source port = 68). La stazione che ha inviato il pacchetto monta una scheda 802.3 Digital (OUI = 08-00-2B).

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 6 arrived at 00:48:05.8680; frame size is 342 (0156 hex)
      bytes.
DLC: Destination = BROADCAST FFFFFFFF, Broadcast Ethernet
DLC: Source       = Station DEC 28C3A5
DLC: Ethertype    = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:   000. .... = routine
IP:   ...0 .... = normal delay
IP:   .... 0... = normal throughput
IP:   .... .0.. = normal reliability
IP: Total length = 328 bytes
IP: Identification = 17069
IP: Flags = 0X
IP: .0... .... = may fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 30 seconds/hops
IP: Protocol = 17 (UDP)
IP: Header checksum = 58F9 (correct)
IP: Source address = [0.0.0.0]
IP: Destination address = [255.255.255.255]
IP: No options
IP:
UDP: ----- UDP Header -----
UDP:
UDP: Source port = 68 (Bootp client)
UDP: Destination port = 67
UDP: Length = 308
UDP: No checksum
UDP:
BOOTP: ----- BOOTP Header -----
BOOTP:
BOOTP: Boot record type          = 1 (Request)
BOOTP: Hardware address type     = 1 10Mb Ethernet
BOOTP: Hardware address length = 6 bytes
BOOTP:
BOOTP: Hops = 0
BOOTP: Transaction id = 26423391
```

[illegible]

ADDR	HEX	ASCII
0000	FF FF FF FF FF 08 00+(....E.
0010	01 48 42 AD 00 00 1E 11	.HB.....X.....
0020	FF FF 00 44 00 43 01 34	...D.C.4.....&B
0030	33 91 AB 78 00 00 00 00	3..x.....
0040	00 00 00 00 00 00 08 00+(.....
0050	00 00 00 00 00 00 00 00
0060	00 00 00 00 00 00 00 00
0070	00 00 00 00 00 00 00 00
0080	00 00 00 00 00 00 00 00
0090	00 00 00 00 00 00 00 00
00A0	00 00 00 00 00 00 00 00
00B0	00 00 00 00 00 00 00 00
00C0	00 00 00 00 00 00 00 00
00D0	00 00 00 00 00 00 00 00
00E0	00 00 00 00 00 00 00 00
00F0	00 00 00 00 00 00 00 00
0100	00 00 00 00 00 00 00 00
0110	00 00 00 00 00 00 63 82c.Sc.....
0120	00 00 00 00 00 00 00 00
0130	00 00 00 00 00 00 00 00
0140	00 00 00 00 00 00 00 00
0150	00 00 00 00 00 00

B.3.3 ARP/RARP

Il protocollo ARP/RARP serve per mantenere una tabella di corrispondenza tra gli indirizzi di livello 3 IP e gli indirizzi di livello 2 MAC. In questo caso la stazione con MAC-DSAP 00-00-0C-00-4D-10 e IP address 130.192.2.17 (un router Cisco) vuole scoprire l'indirizzo MAC della stazione che ha indirizzo IP 130.192.2.46 e a tal fine nel pacchetto ARP/RARP specifica un target hardware address = 00-00-00-00-00-00. La richiesta è inviata con un pacchetto Ethernet in broadcast. Il protocol type di ARP/RARP è 806.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 164 arrived at 00:48:09.9282; frame size is 60 (003C hex)
      bytes.
DLC: Destination = BROADCAST FFFFFFFF, Broadcast Ethernet
DLC: Source       = Station Cisco 004D10, GARR-gw
DLC: Ethertype    = 0806 (ARP)
DLC:
ARP: ----- ARP/RARP frame -----
ARP:
ARP: Hardware type = 1 (10Mb Ethernet)
ARP: Protocol type = 0800 (IP)
ARP: Length of hardware address = 6 bytes
ARP: Length of protocol address = 4 bytes
ARP: Opcode 1 (ARP request)
ARP: Sender's hardware address = Cisco 004D10, GARR-gw
ARP: Sender's protocol address = [130.192.2.17]
ARP: Target hardware address = 000000000000
ARP: Target protocol address = [130.192.2.46]
ARP:

ADDR  HEX                                     ASCII
0000  FF FF FF FF FF FF 00 00 0C 00 4D 10 08 06 00 01  ....M....
0010  08 00 06 04 00 01 00 00 0C 00 4D 10 82 C0 02 11  ....M....
0020  00 00 00 00 00 00 82 C0 02 2E 4A 9A E6 68 03 FD  ....J..h..
0030  00 00 00 00 00 00 05 63 73 65 6C 74             .....cselt
```

B.3.4 RWHO

Rwho è un applicativo che appartiene al gruppo Runix, cui appartengono anche rcp, rlogin, ecc. Rwho invia in broadcast alcune informazioni relative ad un elaboratore, quali gli utenti collegati, la percentuale di utilizzo, il tempo intercorso dal bootstrap, ecc. Il pacchetto Rwho è imbustato in UDP, che è imbustato in IP, che è imbustato in Ethernet. La porta UDP di Rwho è 513.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 223 arrived at 00:48:11.3466; frame size is 102
      (0066 hex) bytes.
DLC: Destination = BROADCAST FFFFFFFF, Broadcast Ethernet
DLC: Source       = Station H-P 1375F0
DLC: Ethertype    = 0800 (IP)
DLC:
IP:  ----- IP Header -----
IP:
IP:  Version = 4, header length = 20 bytes
IP:  Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:  Total length = 88 bytes
IP:  Identification = 19497
IP:  Flags = 0X
IP:  .0.. .... = may fragment
IP:  ..0. .... = last fragment
IP:  Fragment offset = 0 bytes
IP:  Time to live = 30 seconds/hops
IP:  Protocol = 17 (UDP)
IP:  Header checksum = 458C (correct)
IP:  Source address = [130.192.2.97]
IP:  Destination address = [130.192.2.255]
IP:  No options
IP:
UDP: ----- UDP Header -----
UDP:
UDP:  Source port = 513 (Remote who)
UDP:  Destination port = 513
UDP:  Length = 68
UDP:  Checksum = C57C (correct)
UDP:
RUnix: ----- Remote who frame -----
RUnix:
RUnix: Version = 1
RUnix: Type = 1
RUnix: Send time = 30-May-94 22:41:30 GMT
RUnix: Receive time = 0
RUnix: Host = "meucci"
RUnix: Load average = 0.00 (5-minute), 0.00 (10-minute),
      0.00 (15-minute)
RUnix: Boot time = 19-Apr-94 18:02:03 GMT
RUnix: Nobody logged on
RUnix:
RUnix: [Normal end of "Remote who frame".]
RUnix:

```

ADDR	HEX	ASCII
0000	FF FF FF FF FF FF 08 00 09 13 75 F0 08 00 45 00u...E.
0010	00 58 4C 29 00 00 1E 11 45 8C 82 C0 02 61 82 C0	.XL)....E....a..
0020	02 FF 02 01 02 01 00 44 C5 7C 01 01 00 00 2D EAD.-.
0030	6B 9A 00 00 00 00 6D 65 75 63 63 69 00 00 00 00	k.....meucci....
0040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060	00 00 2D B4 1C 9B	..-...

B.3.5 NFS

NFS è il Network File System, un file system distribuito su rete e concepito esplicitamente per il mondo internet. Il pacchetto NFS è imbustato in RPC (Remote Procedural Call), che è imbustato in UDP e via di seguito. RPC è la porta 1021 di UDP e NFS è il Program = 100003 di RPC. Tali numeri di porta non sono previsti dallo standard RFC 1340, ma sono generati dinamicamente dall'applicativo "portmap".

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 154 arrived at 00:48:09.8085; frame size is 278
      (0116 hex) bytes.
DLC: Destination = Station Cisco 004D10, GARR-gw
DLC: Source      = Station DEC 37E61F
DLC: Ethertype   = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: Total length = 264 bytes
IP: Identification = 44847
IP: Flags = 0X
IP: .0.. .... = may fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 30 seconds/hops
IP: Protocol = 17 (UDP)
IP: Header checksum = E027 (correct)
IP: Source address = [130.192.5.4]
IP: Destination address = [130.192.2.10]
IP: No options
IP:

```



```

UDP: ----- UDP Header -----
UDP:
UDP: Source port = 1021 (Sun RPC)
UDP: Destination port = 2049
UDP: Length = 244
UDP: Checksum = 85A2 (correct)
UDP:
RPC: ----- SUN RPC header -----
RPC:
RPC: Transaction id = 2142883117
RPC: Type = 0 (Call)
RPC: RPC version = 2
RPC: Program = 100003 (NFS), version = 2
RPC: Procedure = 8 (Write to file)
RPC: Credentials: authorization flavor = 1 (Unix)
RPC: len = 40, stamp = 770312100
RPC: machine = polpl5.polito.it
RPC: uid = 2029, gid = 2000
RPC: 1 other group id(s):
RPC: gid 2000
RPC: Verifier: authorization flavor = 0 (Null)
RPC: [Verifier: 0 byte(s) of authorization data]
RPC:
RPC: [Normal end of "SUN RPC header".]
RPC:
NFS: ----- SUN NFS -----
NFS:
NFS: Proc = 8 (Write to file)
NFS: File handle = 0A1500001A1000008AB3070802000000
NFS: B8B2067B000000000000000000000000
NFS: Offset = 131072
NFS: [108 byte(s) of data]
NFS:
NFS: [Normal end of "SUN NFS".]
NFS:

```

ADDR	HEX	ASCII
0000	00 00 0C 00 4D 10 08 00 2B 37 E6 1F 08 00 45 00M....+7....E.
0010	01 08 AF 2F 00 00 1E 11 E0 27 82 C0 05 04 82 C0	.../.....'.....
0020	02 0A 03 FD 08 01 00 F4 85 A2 7F B9 CD 2D 00 00-..
0030	00 00 00 00 00 02 00 01 86 A3 00 00 00 02 00 00
0040	00 08 00 00 00 01 00 00 00 28 2D EA 07 A4 00 00(-.....
0050	00 10 70 6F 6C 70 6C 35 2E 70 6F 6C 69 74 6F 2E	..polpl5.polito.
0060	69 74 00 00 07 ED 00 00 07 D0 00 00 00 01 00 00	it.....
0070	07 D0 00 00 00 00 00 00 00 00 0A 15 00 00 1A 10
0080	00 00 8A B3 07 08 02 00 00 00 B8 B2 06 7B 00 00{..
0090	00 00 00 00 00 00 00 00 00 00 02 00 00 00 02
00A0	00 00 00 00 00 00 6C 00 00 00 6C 20 20 20 20 201...1
00B0	20 20 32 30 34 39 20 44 27 41 47 4F 53 54 49 4E	2049 D'AGOSTIN

```

00C0 4F 20 20 20 20 20 20 20 20 20 20 20 20 20 20 0
00D0 20 20 20 20 20 20 0A 20 20 63 70 75 20 3A 20 20 35 . cpu : 5
00E0 2E 36 36 20 73 65 63 2E 20 0A 20 20 20 20 20 20 .66 sec. .
00F0 20 20 32 30 35 30 20 4D 4F 54 54 41 20 20 20 20 2050 MOTTA
0100 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0110 20 20 20 20 20 20 0A .

```

B.3.6 TFTP

Il TFTP è il Trivial File Transfer Protocol utilizzato per effettuare un file transfer molto semplici su internet. TFTP è associato alla porta 69 di UDP che è imbustato in IP e così via. In questo caso il TFTP chiede di leggere il file \config.sys sul calcolatore con indirizzo IP 128.1.0.1 e indirizzo MAC 00-00-C0-E5-8C-11. Il calcolatore monta una scheda di rete locale Western Digital (OUI = 00-00-C0).

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 3 arrived at 13:41:17.9026; frame size is 75 (004B hex)
      bytes.
DLC: This frame is dated 17 day(s) after capture started.
DLC: Destination = Station WstDigE58C11
DLC: Source      = Station WstDig488C11
DLC: Ethertype   = 0800 (IP)
DLC:
IP:  ----- IP Header -----
IP:
IP:  Version = 4, header length = 20 bytes
IP:  Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:  Total length = 61 bytes
IP:  Identification = 1
IP:  Flags = 0X
IP:  .0.. .... = may fragment
IP:  ..0. .... = last fragment
IP:  Fragment offset = 0 bytes
IP:  Time to live = 255 seconds/hops
IP:  Protocol = 17 (UDP)
IP:  Header checksum = BBA9 (correct)
IP:  Source address = [128.1.0.2]
IP:  Destination address = [128.1.0.1]
IP:  No options
IP:

```

```

UDP: ----- UDP Header -----
UDP:
UDP: Source port = 1004 (TFTP)
UDP: Destination port = 69
UDP: Length = 41
UDP: Checksum = BBED (correct)
UDP:
TFTP: ----- Trivial file transfer -----
TFTP:
TFTP: Opcode = 1 (Read request)
TFTP: File name = "\config.sys"
TFTP: Mode = "netascii"
TFTP:
TFTP: *** 10 byte(s) of additional data present ***
TFTP:
TFTP: [Abnormal end of "Trivial file transfer".]
TFTP:

```

ADDR	HEX	ASCII
0000	00 00 C0 E5 8C 11 00 00 C0 48 8C 11 08 00 45 00H....E.
0010	00 3D 00 01 00 00 FF 11 BB A9 80 01 00 02 80 01	.=.....
0020	00 01 03 EC 00 45 00 29 BB ED 00 01 5C 63 6F 6EE.)....\con
0030	66 69 67 2E 73 79 73 00 6E 65 74 61 73 63 69 69	fig.sys.netascii
0040	00 00 00 00 00 00 00 00 00 00 00

B.3.7 DNS

Il DNS è il Domain Name Service, cioè quell'applicativo di rete che serve per tradurre i nomi negli indirizzi IP e viceversa. DNS si appoggia sulla porta 53 di UDP, UDP viene imbustato in IP, ecc. La query DNS vuole scoprire l'indirizzo dell'host ercole.polito.it.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 728 arrived at 00:48:24.2087; frame size is 76 (004C hex)
bytes.
DLC: Destination = Station DECnet00E97D, ERCOLE
DLC: Source      = Station DEC 181C22
DLC: Ethertype   = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay

```

```

IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      Total length = 62 bytes
IP:      Identification = 15103
IP:      Flags = 0X
IP:      .0.. .... = may fragment
IP:      ..0. .... = last fragment
IP:      Fragment offset = 0 bytes
IP:      Time to live = 30 seconds/hops
IP:      Protocol = 17 (UDP)
IP:      Header checksum = 5623 (correct)
IP:      Source address = [130.192.3.12]
IP:      Destination address = [130.192.3.1]
IP:      No options
IP:
UDP:      ----- UDP Header -----
UDP:
UDP:      Source port = 2076 (Domain)
UDP:      Destination port = 53
UDP:      Length = 42
UDP:      Checksum = B149 (correct)
UDP:
DNS:      ----- Internet Domain Name Service header -----
DNS:
DNS:      ID = 9346
DNS:      Flags = 01
DNS:      0... .... = Command
DNS:      .000 0... = Query
DNS:      .... ..0. = Not truncated
DNS:      .... ...1 = Recursion desired
DNS:      Flags = 0X
DNS:      ...0 .... = Unicast packet
DNS:      Question count = 1, Answer count = 0
DNS:      Authority count = 0, Additional record count = 0
DNS:
DNS:      Question section:
DNS:      Name = ercole.polito.it
DNS:      Type = Host address (A,1)
DNS:      Class = Internet (IN,1)
DNS:
DNS:      [Normal end of "Internet Domain Name Service header".]
DNS:

ADDR  HEX                                     ASCII
0000  AA 00 04 00 E9 7D 08 00 2B 18 1C 22 08 00 45 00  ....}..+..."..E.
0010  00 3E 3A FF 00 00 1E 11 56 23 82 C0 03 0C 82 C0  .>:.....V#.....
0020  03 01 08 1C 00 35 00 2A B1 49 24 82 01 00 00 01  ....5.*.I$. ....
0030  00 00 00 00 00 00 06 65 72 63 6F 6C 65 06 70 6F  ....ercole.po
0040  6C 69 74 6F 02 69 74 00 00 01 00 01          lito.it.....

```

B.3.8 X Windows

X Windows è indubbiamente uno degli applicativi più interessanti e moderni che possono essere trasportati su una rete internet. Il pacchetto X Windows in oggetto chiede di visualizzare la scritta "New mail on node POL88B from IN% "VANNOZZI@NIS.GARR.IT" "Daniele Vannozzi"" su una finestra X, collocata sull'host con indirizzo IP 130.192.5.41 e indirizzo MAC 08-00-2B-39-02-21. La richiesta è associata alla porta 6000 di TCP ed è generata dall'host con nome POL88B, indirizzo IP 130.192.2.16 e indirizzo MAC AA-00-04-00-92-7D.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 860 arrived at 00:48:27.3165; frame size is 274
      (0112 hex) bytes.
DLC: Destination = Station DEC 390221
DLC: Source      = Station DECnet00927D, POL88B
DLC: Ethertype   = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: Total length = 260 bytes
IP: Identification = 36610
IP: Flags = 0X
IP: .0.. .... = may fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 60 seconds/hops
IP: Protocol = 6 (TCP)
IP: Header checksum = E238 (correct)
IP: Source address = [130.192.2.16]
IP: Destination address = [130.192.5.41]
IP: No options
IP:
TCP: ----- TCP header -----
TCP:
TCP: Source port = 4005
TCP: Destination port = 6000 (X Windows)
TCP: Sequence number = 941492665
TCP: Acknowledgment number = 84081
TCP: Data offset = 20 bytes
TCP: Flags = 18
```

```
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 6144
TCP: Checksum = 9800 (correct)
TCP: No TCP options
TCP: [220 byte(s) of data]
TCP:
XWIN: ----- X Windows -----
XWIN:
XWIN: Request opcode = 70 (Poly Fill Rectangle)
XWIN: Drawable = 00300015, Graphics context = 0030001A
XWIN: X = 70, Y = 464, Width = 11, Height = 20
XWIN:
XWIN: Request opcode = 56 (Change GC)
XWIN: Graphics context = 00300016
XWIN: Value mask = 00010000
XWIN: Graphics exposures = 1 (True)
XWIN:
XWIN: Request opcode = 62 (Copy Area)
XWIN: Source drawable = 00300015
XWIN: Destination drawable = 00300015
XWIN: Graphics context = 00300016
XWIN: Source X = 4, Y = 24
XWIN: Destination X = 4, Y = 4
XWIN: Width = 880, Height = 460
XWIN:
XWIN: Request opcode = 61 (Clear Area)
XWIN: Exposures = 0 (False)
XWIN: Window = 00300015
XWIN: X = 4, Y = 464, Width = 880, Height = 20
XWIN:
XWIN: Request opcode = 104 (Bell)
XWIN: Percent = 0
XWIN:
XWIN: Request opcode = 76 (Image Text8)
XWIN: Drawable = 00300015, Graphics context = 00300027
XWIN: X = 4, Y = 479
XWIN: String = "New mail on node POL88B from IN%"VANNONZZI@NIS.GARR.IT"
        "Daniele Vannozzi"
XWIN:
XWIN: Request opcode = 62 (Copy Area)
XWIN: Source drawable = 00300015
XWIN: Destination drawable = 00300015
XWIN: Graphics context = 00300016
XWIN: Source X = 4, Y = 24
```

```
XWIN: Destination X = 4, Y = 4
XWIN: Width = 880, Height = 460
XWIN:
XWIN: Request opcode = 61 (Clear Area)
XWIN: Exposures = 0 (False)
XWIN: Window = 00300015
XWIN: X = 4, Y = 464, Width = 880, Height = 20
XWIN:
```

ADDR	HEX	ASCII
0000	08 00 2B 39 02 21 AA 00 04 00 92 7D 08 00 45 00	..+9.!.....}..E.
0010	01 04 8F 02 00 00 3C 06 E2 38 82 C0 02 10 82 C0<..8.....
0020	05 29 0F A5 17 70 38 1E 09 B9 00 01 48 71 50 18	.)...p8.....HqP.
0030	18 00 98 00 00 00 46 00 05 00 15 00 30 00 1A 00F.....0...
0040	30 00 46 00 D0 01 0B 00 14 00 38 00 04 00 16 00	0.F.....8.....
0050	30 00 00 00 01 00 01 00 00 00 3E 00 07 00 15 00	0.....>.....
0060	30 00 15 00 30 00 16 00 30 00 04 00 18 00 04 00	0...0...0.....
0070	04 00 70 03 CC 01 3D 00 04 00 15 00 30 00 04 00	..p...=.....0...
0080	D0 01 70 03 14 00 68 00 01 00 4C 4A 17 00 15 00	..p...h...LJ....
0090	30 00 27 00 30 00 04 00 DF 01 4E 65 77 20 6D 61	0.'0.....New ma
00A0	69 6C 20 6F 6E 20 6E 6F 64 65 20 50 4F 4C 38 38	il on node POL88
00B0	42 20 66 72 6F 6D 20 49 4E 25 22 56 41 4E 4E 4F	B from IN%"VANNO
00C0	5A 5A 49 40 4E 49 53 2E 47 41 52 52 2E 49 54 22	ZZI@NIS.GARR.IT"
00D0	20 20 22 44 61 6E 69 65 6C 65 20 56 61 6E 6E 6F	"Daniele Vanno
00E0	7A 7A 69 22 30 00 3E 00 07 00 15 00 30 00 15 00	zzi"0.>.....0...
00F0	30 00 16 00 30 00 04 00 18 00 04 00 04 00 70 03	0...0.....p.
0100	CC 01 3D 00 04 00 15 00 30 00 04 00 D0 01 70 03	..=.....0.....p.
0110	14 00	..

B.3.9 RIP

Il RIP è il Routing Information Protocol, un protocollo di routing utilizzato su reti IP di dominio pubblico. Il RIP è associato alla porta 520 di UDP. Il pacchetto UDP è imbustato su IP e così via. Il pacchetto è inviato in broadcast a livello MAC dalla stazione AA-00-04-00-CD-7D che ha nome POLME2 e indirizzo IP 130.192.2.147. Si noti che a livello 3 (IP) il broadcast è limitato alla subnet 2 della net 130.192, come appare evidente dall'IP destination address = [130.192.2.255] (255 è appunto il broadcast).

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 1823 arrived at 00:48:51.0046; frame size is 66
      (0042 hex) bytes.
DLC: Destination = BROADCAST FFFFFFFF, Broadcast Ethernet
DLC: Source      = Station DEcnet00CD7D, POLME2
```

```

DLC: Ethertype = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP: Total length = 52 bytes
IP: Identification = 23830
IP: Flags = 0X
IP:  .0.. .... = may fragment
IP:  ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 30 seconds/hops
IP: Protocol = 17 (UDP)
IP: Header checksum = 3491 (correct)
IP: Source address = [130.192.2.147]
IP: Destination address = [130.192.2.255]
IP: No options
IP:
UDP: ----- UDP Header -----
UDP:
UDP: Source port = 520 (Route)
UDP: Destination port = 520
UDP: Length = 32
UDP: Checksum = 69C7 (correct)
UDP:
RIP: ----- RIP Header -----
RIP:
RIP: Command = 2 (Response)
RIP: Version = 1
RIP: Unused = 0
RIP:
RIP: Routing data frame 1
RIP:   Address family identifier = 2 (IP)
RIP:   IP Address = [130.192.2.0]
RIP:   Metric = 1
RIP:

```

ADDR	HEX	ASCII
0000	FF FF FF FF FF FF AA 00 04 00 CD 7D 08 00 45 00}..E.
0010	00 34 5D 16 00 00 1E 11 34 91 82 C0 02 93 82 C0	.4].....4.....
0020	02 FF 02 08 02 08 00 20 69 C7 02 01 00 00 00 02i.....
0030	00 00 82 C0 02 00 00 00 00 00 00 00 00 00 00 00
0040	00 01	..

B.3.10 YP

L'applicativo YP (Yellow Pages) è un applicativo proposto da SUN ed aggiuntivo rispetto al DNS per gestire i nomi, gli indirizzi, gli UID e i GID in modo centralizzato, caratteristica questa indispensabile per la sicurezza di applicativi quali NFS. In questo caso il server di YP è quello relativo al dominio CISIP.polito.it e si tratta di una workstation Sony che interroga un server SUN. YP è imbustato in RPC, che è imbustato in UDP, che è imbustato in IP, che è imbustato in Ethernet.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 5279 arrived at 00:50:21.2675; frame size is 118
      (0076 hex) bytes.
DLC: Destination = Station Sun 11F49F
DLC: Source      = Station Sony 004832
DLC: Ethertype   = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: Total length = 104 bytes
IP: Identification = 64122
IP: Flags = 0X
IP: .0.. .... = may fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 30 seconds/hops
IP: Protocol = 17 (UDP)
IP: Header checksum = 94CB (correct)
IP: Source address = [130.192.5.22], poltcux1
IP: Destination address = [130.192.2.169]
IP: No options
IP:
UDP: ----- UDP Header -----
UDP:
UDP: Source port = 2376
UDP: Destination port = 660 (Sun RPC)
UDP: Length = 84
UDP: No checksum
UDP:
RPC: ----- SUN RPC header -----
RPC:
```

```

RPC: Transaction id = 770527999
RPC: Type = 0 (Call)
RPC: RPC version = 2
RPC: Program = 100004 (Yellow pages), version = 2
RPC: Procedure = 4 (Get first keyu-value pair in map)
RPC: Credentials: authorization flavor = 0 (Null)
RPC: [Credentials: 0 byte(s) of authorization data]
RPC: Verifier: authorization flavor = 0 (Null)
RPC: [Verifier: 0 byte(s) of authorization data]
RPC:
RPC: [Normal end of "SUN RPC header".]
RPC:
YP: ----- SUN Yellow Pages -----
YP:
YP: Proc = 4 (Get first keyu-value pair in map)
YP: Domain = CISIP.polito.it
YP: Map = group.byname
DLC: --- Frame too short

```

ADDR	HEX	ASCII
0000	08 00 20 11 F4 9F 08 00 46 00 48 32 08 00 45 00F.H2..E.
0010	00 68 FA 7A 00 00 1E 11 94 CB 82 C0 05 16 82 C0	.h.z.....
0020	02 A9 09 48 02 94 00 54 00 00 2D ED 52 FF 00 00	...H...T..-.R...
0030	00 00 00 00 00 02 00 01 86 A4 00 00 00 02 00 00
0040	00 04 00 00 00 00 00 00 00 00 00 00 00 00 00
0050	00 00 00 00 00 00 0F 43 49 53 49 50 2E 70 6F 6C 69CISIP.poli
0060	74 6F 2E 69 74 00 00 00 00 0C 67 72 6F 75 70 2E	to.it.....group.
0070	62 79 6E 61 6D 65	byname

B.3.11 SMB su TCP/IP

I pacchetti SMB possono essere imbustati in vari protocolli. In B.2.2 si è analizzato il caso di SMB imbustato in Netbios IBM; qui SMB viene imbustato in TCP (porta 46586), che a sua volta viene imbustato in IP. Naturalmente, perché due protocolli SMB possano parlarsi è indispensabile che tutti gli imbustamenti siano compatibili (un SMB su Netbios IBM non può dialogare con un SMB su TCP/IP).

In questo esempio si vede per la prima volta il trasporto di IP su 802.3. Si sfrutta il LLC con LLC-SSAP e LLC-DSAP uguali a 06, codifica definita dall'IEEE per il Department of Defense Internet Protocol. Altre possibilità sono il trasporto di IP come pacchetto SNAP (descritto nel capitolo 5 e previsto dal RFC 1042) oppure l'utilizzo dell'imbustamento Ethernet come ad esempio in A.3.10.

```

DLC: ----- DLC Header -----
DLC:

```

```

DLC: Frame 3 arrived at 14:30:33.0552; frame size is 92 (005C hex)
      bytes.
DLC: Destination = Station 3Com 138162
DLC: Source      = Station 3Com 138372
DLC: 802.3 length = 78
DLC:
LLC: ----- LLC Header -----
LLC:
LLC: DSAP = 06, SSAP = 06, Command, Unnumbered frame: UI
LLC:
IP:  ----- IP Header -----
IP:
IP:  Version = 4, header length = 20 bytes
IP:  Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:  Total length = 75 bytes
IP:  Identification = 0
IP:  Flags = 4X
IP:  .1.. .... = don't fragment
IP:  ..0. .... = last fragment
IP:  Fragment offset = 0 bytes
IP:  Time to live = 60 seconds/hops
IP:  Protocol = 6 (TCP)
IP:  Header checksum = 8E2B (correct)
IP:  Source address = [15.6.73.50]
IP:  Destination address = [15.6.73.68]
IP:  No options
IP:
TCP: ----- TCP header -----
TCP:
TCP:  Source port = 5696
TCP:  Destination port = 46586
TCP:  Sequence number = 346879454
TCP:  Acknowledgment number = 1109177
TCP:  Data offset = 20 bytes
TCP:  Flags = 18
TCP:  ..0. .... = (No urgent pointer)
TCP:  ...1 .... = Acknowledgment
TCP:  .... 1... = Push
TCP:  .... .0.. = (No reset)
TCP:  .... ..0. = (No SYN)
TCP:  .... ...0 = (No FIN)
TCP:  Window = 1424
TCP:  Checksum = 0 (No checksum sent)
TCP:  No TCP options
TCP:  [35 byte(s) of data]

```

```

TCP:
SMB: ----- SMB Spool Byte Range Response -----
SMB:
SMB: Function = C1 (Spool Byte Range)
SMB: Tree id      (TID) = 0021
SMB: Process id   (PID) = 354A
SMB: Return code = 0,0 (OK)
SMB:

ADDR  HEX                                     ASCII
0000  02 60 8C 13 81 62 02 60  8C 13 83 72 00 4E 06 06  .`.b.`.r.N..
0010  03 45 00 00 4B 00 00 40  00 3C 06 8E 2B 0F 06 49  .E..K..@.<.+..I
0020  32 0F 06 49 44 16 40 B5  FA 14 AC F5 DE 00 10 EC  2..ID.@.....
0030  B9 50 18 05 90 00 00 00  00 FF 53 4D 42 C1 00 00  .P.....SMB...
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0050  00 21 00 4A 35 00 00 00  00 00 00 00 00 00 00 00  .!.J5.....

```

B.3.12 ISODE

ISODE è un ambiente per sviluppare applicazioni OSI su trasporto TCP/IP e renderne estremamente semplice la portabilità su architetture OSI. Si noti come un pacchetto di trasporto OSI è imbustato in un pacchetto di Development Environment, che è imbustato in un pacchetto TCP, che è imbustato in un pacchetto IP, che è imbustato in un pacchetto Ethernet.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 5 arrived at 09:36:42.0164; frame size is 60 (003C hex)
    bytes.
DLC: Destination = Station Sun 01DA53
DLC: Source      = Station Exceln518736
DLC: Ethertype   = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: Total length = 46 bytes
IP: Identification = 4997
IP: Flags = 0X
IP: .0.. .... = may fragment

```

```

IP:  ..0. .... = last fragment
IP:  Fragment offset = 0 bytes
IP:  Time to live = 60 seconds/hops
IP:  Protocol = 6 (TCP)
IP:  Header checksum = 59C6 (correct)
IP:  Source address = [192.9.200.170]
IP:  Destination address = [192.9.200.193]
IP:  No options
IP:
TCP:  ----- TCP header -----
TCP:
TCP:  Source port = 1082
TCP:  Destination port = 102 (ISO)
TCP:  Sequence number = 1065666
TCP:  Acknowledgment number = 38786
TCP:  Data offset = 20 bytes
TCP:  Flags = 18
TCP:  ..0. .... = (No urgent pointer)
TCP:  ...1 .... = Acknowledgment
TCP:  .... 1... = Push
TCP:  .... .0.. = (No reset)
TCP:  .... ..0. = (No SYN)
TCP:  .... ...0 = (No FIN)
TCP:  Window = 4096
TCP:  Checksum = 9A5C (correct)
TCP:  No TCP options
TCP:  [6 byte(s) of data]
TCP:
ISO_DE:  ----- ISO Development Environment -----
ISO_DE:
ISO_DE:  Multi-frame TPDU: frames 5, 6, 7
ISO_DE:  Version = 3
ISO_DE:  Packet length = 22
ISO_DE:
ISO_TP:  ----- ISO Transport Layer -----
ISO_TP:
ISO_TP:  Header length = 17
ISO_TP:  TPDU type = E (Connection request)
ISO_TP:  Destination reference = 0000
ISO_TP:      Source reference = 0096
ISO_TP:  Additional options = 01
ISO_TP:      .... 0... = No use of network expedited in class 1
ISO_TP:      .... .0.. = Use explicit AK variant in class 1
ISO_TP:      .... ..0. = Use 16-bit checksum in class 4
ISO_TP:      .... ...1 = Use expedited data transfer
ISO_TP:      Source TSAP: "<8096>"
ISO_TP:  Destination TSAP: "<0206>"
ISO_TP:

```

ADDR	HEX	ASCII
0000	08 00 20 01 DA 53 08 00 14 51 87 36 08 00 45 00	..S...Q.6..E.
0010	00 2E 13 85 00 00 3C 06 59 C6 C0 09 C8 AA C0 09<.Y.....
0020	C8 C1 04 3A 00 66 00 10 42 C2 00 00 97 82 50 18	...:..f..B....P.
0030	10 00 9A 5C 00 00 03 00 00 16 11 E0	...\\.....

B.4 DECNET FASE IV

Gli esempi dei sottoparagrafi seguenti sono relativi all'architettura di rete proprietaria DECnet fase IV. L'evoluzione di DECnet nota con il nome di DECnet fase V è invece totalmente compatibile con OSI e per gli esempi si veda il paragrafo A.6.

B.4.1 DAP

Il DAP è il Data Access Protocol, un protocollo usato per accedere a dati su host remoti. Il DAP è imbustato nel Network Service Protocol (NSP, livello 4), che è imbustato nel DECnet Routing Protocol (DRP, livello 3), che è imbustato in Ethernet (protocol type = 6003). Gli indirizzi di livello 3 dei nodi DECnet mittente e destinatario sono rispettivamente 7.45 e 7.52, cioè appartengono entrambi all'area DECnet numero 7.

A questi due indirizzi di livello 3 corrispondono due MAC address AA-00-04-00-2D-1C e AA-00-04-00-34-1C, che sono calcolati algebricamente a partire dagli indirizzi di livello 3 e cioè facendo seguire ad AA-00-04-00 i 16 bit dell'indirizzo di livello 3, con i due byte scambiati. Per un esempio di calcolo si veda A.4.3.

All'atto del bootstrap il protocollo DECnet provvede a sostituire gli indirizzi MAC delle schede con quelli calcolati algebricamente a partire dagli indirizzi di livello 3.

```

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 13 arrived at 17:01:07.9740; frame size is 71 (0047 hex)
      bytes.
DLC:  Destination = Station DECnet00341C
DLC:  Source      = Station DECnet002D1C
DLC:  Ethertype   = 6003 (DECNET)
DLC:
DRP:  ----- DECNET Routing Protocol -----
DRP:
DRP:  Data Length = 55,  Optional Padding Length = 1
DRP:  Data Packet Format = 26
DRP:           0... .... = no padding
DRP:           .0.. .... = version

```

```

DRP:      ...1. .... = Intra-Ethernet packet
DRP:      ...0 .... = not return packet
DRP:      .... 0... = do not return to sender
DRP:      .... .110 = Long Data Packet Format
DRP: Data Packet Type = 6
DRP: Destination Area      = 00
DRP: Destination Subarea   = 00
DRP: Destination ID        = 7.52
DRP: Source Area           = 00
DRP: Source Subarea        = 00
DRP: Source ID             = 7.45
DRP: Next Level 2 Router   = 00
DRP: Visit Count           = 0
DRP: Service Class         = 00
DRP: Protocol Type         = 00
DRP:
NSP: ----- Network Services Protocol -----
NSP:
NSP: Message Identifier = 60
NSP:      0... .... = Non-extensible field
NSP:      .110 .... = Begin-End Data Message
NSP:      .... 00.. = Data Message
NSP:      .... ..00 = always zero
NSP: Type           = 0 (Data Message)
NSP: Sub-type       = 6 (Begin-End Data Message)
NSP: Logical Link Destination = 1413
NSP: Logical Link Source      = 0C39
NSP: Data Acknowledgment Number
NSP:   Acknowledge Qualifier      = ACK
NSP:   Message Number Acknowledged = 1
NSP: Data Segment Number = 2 (normal ACK expected)
NSP: [24 data bytes]
NSP:
DAP: ----- Data Access Protocol -----
DAP:
DAP: Code = 3 (Access)
DAP: Access Function = Directory List
DAP: Access Options Type:
DAP:   Bit 0: I/O errors are non-fatal
DAP: File Name Specification = "SYS$MANAGER:*.EXE;*"
DAP:

ADDR  HEX                                     ASCII
0000  AA 00 04 00 34 1C AA 00 04 00 2D 1C 60 03 37 00  ....4.....-.`.7.
0010  81 26 00 00 AA 00 04 00 34 1C 00 00 AA 00 04 00  .&.....4.....
0020  2D 1C 00 00 00 00 60 13 14 39 0C 01 80 02 00 03  -.....`.9.....
0030  00 06 01 13 53 59 53 24 4D 41 4E 41 47 45 52 3A  ....SYS$MANAGER:
0040  2A 2E 45 58 45 3B 2A                                     *.EXE;*

```

B.4.2 End System Hello

Il seguente pacchetto è un End System Hello del livello 3 DECnet (DRP) generato da un end system con indirizzo 60.975 per essere riconosciuto da un router ed incluso nelle tabelle di instradamento. Il pacchetto è trasmesso all'indirizzo MAC di multicast del DECnet e cioè AB-00-00-03-00-00.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 96 arrived at 00:48:08.1274; frame size is 60 (003C hex)
      bytes.
DLC: Destination = Multicast AB0000030000
DLC: Source       = Station DECnet00CFF3
DLC: Ethertype    = 6003 (DECNET)
DLC:
DRP: ----- DECNET Routing Protocol -----
DRP:
DRP: Data length = 33
DRP: Control Packet Format = 0D
DRP:      0... .... = no padding
DRP:      .000 .... = reserved
DRP:      .... 110. = Ethernet Endnode Hello Message
DRP:      .... ...1 = Control Packet Format
DRP: Control Packet Type = 06
DRP: Version Number   = 02
DRP: ECO Number       = 00
DRP: User ECO Number  = 00
DRP: ID of Transmitting Node = 60.975
DRP: Information      = 03
DRP:      0... .... = reserved
DRP:      .0.. .... = not blocking request
DRP:      ..0. .... = multicast traffic accepted
DRP:      ...0 .... = verification ok
DRP:      .... 0... = do not reject
DRP:      .... .0.. = no verification required
DRP:      .... ..11 = endnode
DRP: Receive Block Size = 1498
DRP: Area (reserved)    = 0
DRP: Verification Seed  = 0000000000000000
DRP: Neighbor System ID = 60.75
DRP: Hello timer (seconds) = 30
DRP: MPD (reserved)     = 0
DRP: [1 bytes of Data to test the circuit]
DRP:

```

ADDR	HEX	ASCII
0000	AB 00 00 03 00 00 AA 00 04 00 CF F3 60 03 21 00`!.
0010	0D 02 00 00 AA 00 04 00 CF F3 03 DA 05 00 00 00


```
0020 00 00 00 00 00 00 AA 00 04 00 4B F0 1E 00 00 01 .....K.....
0030 AA 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

B.4.3 Router Hello

Questo pacchetto differisce dal precedente in quanto trasmesso da un router per farsi conoscere dagli end node. Il router mittente è il 31.458 (nome POLFIS) come si può capire anche dall'indirizzo di mittente MAC. Infatti $31 \times 1024 + 458$ è uguale a 32202, cioè in esadecimale 7D-CA. Invertendo i due byte si ottiene CA-7D e premettendo l'OUI DECnet AA-00-04 si ottiene un MAC-SSAP aa-00-04-00-CA-7D.

Il pacchetto contiene anche gli indirizzi di livello 3 degli altri router collegati alla stessa LAN e cioè: 31.492, 31.501, 31.402, 31.401 e 31.412.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 139 arrived at 00:48:09.3545; frame size is 78 (004E hex)
      bytes.
DLC: Destination = Multicast AB0000030000
DLC: Source       = Station DECnet00CA7D, POLFIS
DLC: Ethertype    = 6003 (DECNET)
DLC:
DRP: ----- DECNET Routing Protocol -----
DRP:
DRP: Data length = 62
DRP: Control Packet Format = 0B
DRP:      0... .... = no padding
DRP:      .000 .... = reserved
DRP:      .... 101. = Ethernet Router Hello Message
DRP:      .... ...1 = Control Packet Format
DRP: Control Packet Type = 05
DRP: Version Number    = 02
DRP: ECO Number        = 00
DRP: User ECO Number   = 00
DRP: ID of Transmitting Node = 31.458
DRP:      Information = 02
DRP:      0... .... = reserved
DRP:      .0.. .... = not blocking request
DRP:      ..0. .... = multicast traffic accepted
DRP:      ...0 .... = verification ok
DRP:      .... 0... = do not reject
DRP:      .... .0.. = no verification required
DRP:      .... ..10 = level 1 router
DRP: Receive Block Size = 1498
DRP: Router's priority  = 64
DRP: Area (reserved)    = 0

```

```

DRP: Hello timer (seconds) = 15
DRP: MPD (reserved) = 0
DRP: E-List length = 43
DRP: Ethernet Name, reserved = 00000000000000
DRP: Router/State length = 35
DRP:
DRP: Router ID = 31.492
DRP: Priority and State = C0
DRP:      1... .... = State known 2-way
DRP:      .100 0000 = Router's priority
DRP:
DRP: Router ID = 31.501
DRP: Priority and State = C0
DRP:      1... .... = State known 2-way
DRP:      .100 0000 = Router's priority
DRP:
DRP: Router ID = 31.402
DRP: Priority and State = C0
DRP:      1... .... = State known 2-way
DRP:      .100 0000 = Router's priority
DRP:
DRP: Router ID = 31.401
DRP: Priority and State = C0
DRP:      1... .... = State known 2-way
DRP:      .100 0000 = Router's priority
DRP:
DRP: Router ID = 31.412
DRP: Priority and State = C0
DRP:      1... .... = State known 2-way
DRP:      .100 0000 = Router's priority
DRP:

```

ADDR	HEX	ASCII
0000	AB 00 00 03 00 00 AA 00 04 00 CA 7D 60 03 3E 00}`.>.
0010	0B 02 00 00 AA 00 04 00 CA 7D 02 DA 05 40 00 0F}...@..
0020	00 00 2B 00 00 00 00 00 00 00 23 AA 00 04 00 EC	...+.....#.....
0030	7D C0 AA 00 04 00 F5 7D C0 AA 00 04 00 92 7D C0	}.....}.....}.
0040	AA 00 04 00 91 7D C0 AA 00 04 00 9C 7D C0}.....}.

B.4.4 Routing di livello 1

Il seguente pacchetto è un distance vector di livello 1 trasmesso dal router 60.8 (nome DIDVX2), e riguardante quindi l'area 60, ai router ad esso adiacenti. Per ogni nodo dell'area 60 contiene una tripletta {Indirizzo, Hops, Costo}. I nodi noti al router DIDVX2 sono 32 e quindi il distance vector contiene 32 triplette.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 88 arrived at 00:48:07.9638; frame size is 294 (0126 hex)
      bytes.
DLC: Destination = Multicast AB0000030000
DLC: Source       = Station DECnet0008F0, DIDVX2
DLC: Ethertype    = 6003 (DECNET)
DLC:
DRP: ----- DECNET Routing Protocol -----
DRP:
DRP: Data length = 278
DRP: Control Packet Format = 07
DRP:      0... .... = no padding
DRP:      .000 .... = reserved
DRP:      .... 011. = Level 1 Routing Message
DRP:      .... ...1 = Control Packet Format
DRP: Control Packet Type = 03
DRP: Source Node       = 60.8
DRP: Reserved field = 0
DRP: RTGINFO count = 32
DRP: Start ID = 480
DRP: ID = 480  Hops = 31  Cost = 1023
DRP: ID = 481  Hops = 31  Cost = 1023
DRP: ID = 482  Hops = 31  Cost = 1023
DRP: ID = 483  Hops = 31  Cost = 1023
DRP: ID = 484  Hops = 31  Cost = 1023
DRP: ID = 485  Hops = 31  Cost = 1023
DRP: ID = 486  Hops = 01  Cost = 4
DRP: ID = 487  Hops = 01  Cost = 4
DRP: ID = 488  Hops = 31  Cost = 1023
DRP: ID = 489  Hops = 31  Cost = 1023
DRP: ID = 490  Hops = 01  Cost = 4
DRP: ID = 491  Hops = 11  Cost = 44
DRP: ID = 492  Hops = 31  Cost = 1023
DRP: ID = 493  Hops = 31  Cost = 1023
DRP: ID = 494  Hops = 31  Cost = 1023
DRP: ID = 495  Hops = 31  Cost = 1023
DRP: ID = 496  Hops = 31  Cost = 1023
DRP: ID = 497  Hops = 31  Cost = 1023
DRP: ID = 498  Hops = 01  Cost = 4
DRP: ID = 499  Hops = 31  Cost = 1023
DRP: ID = 500  Hops = 31  Cost = 1023
DRP: ID = 501  Hops = 31  Cost = 1023
DRP: ID = 502  Hops = 31  Cost = 1023
DRP: ID = 503  Hops = 31  Cost = 1023
DRP: ID = 504  Hops = 31  Cost = 1023
DRP: ID = 505  Hops = 31  Cost = 1023
DRP: ID = 506  Hops = 31  Cost = 1023
DRP: ID = 507  Hops = 31  Cost = 1023
```

```

DRP: ID = 508 Hops = 31 Cost = 1023
DRP: ID = 509 Hops = 01 Cost = 4
DRP: ID = 510 Hops = 01 Cost = 4
DRP: ID = 511 Hops = 01 Cost = 4
DRP:
DRP: Checksum = 0020
DRP:

```

ADDR	HEX	ASCII
0000	AB 00 00 03 00 00 AA 00 04 00 08 F0 60 03 16 01`...
0010	07 08 F0 00 20 00 E0 01 FF 7F FF 7F FF 7F FF 7F
0020	FF 7F FF 7F 04 04 04 04 FF 7F FF 7F 04 04 2C 2C,,
0030	FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F 04 04 FF 7F
0040	FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
0050	FF 7F 04 04 04 04 04 04 20 00 00 02 FF 7F FF 7F
0060	FF 7F FF 7F 04 04 FF 7F 20 20 FF 7F FF 7F FF 7F
0070	04 04 04 04 04 04 04 04 04 04 04 04 FF 7F FF 7F
0080	FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F 04 04
0090	04 04 FF 7F FF 7F FF 7F FF 7F 04 04 20 00 A0 02
00A0	FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F 10 10
00B0	FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
00C0	FF 7F 04 04 04 04 FF 7F FF 7F FF 7F FF 7F FF 7F
00D0	FF 7F FF 7F FF 7F FF 7F FF 7F 04 04 FF 7F 04 04
00E0	20 00 C0 02 FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
00F0	FF 7F FF 7F FF 7F FF 7F 04 04 FF 7F FF 7F 04 04
0100	20 20 FF 7F FF 7F FF 7F 48 48 FF 7F FF 7F FF 7FHH.....
0110	FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F
0120	FF 7F FF 7F B0 2A*

B.4.5 Routing di livello 2

Il seguente pacchetto è trasmesso dal router di area 31 e indirizzo 31.501 e contiene un distance vector di livello 2. Si tratta di 63 triplette {Area, Hops, Costo} che indicano la visibilità del router 31.501 sulle 63 aree DECnet (massimo teorico).

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 113 arrived at 00:48:08.5342; frame size is 152
      (0098 hex) bytes.
DLC: Destination = Multicast AB0000030000
DLC: Source       = Station DECnet00F57D
DLC: Ethertype    = 6003 (DECNET)
DLC:
DRP: ----- DECNET Routing Protocol -----
DRP:
DRP: Data length = 136

```

```

DRP: Control Packet Format = 09
DRP:      0... .... = no padding
DRP:      .000 .... = reserved
DRP:      .... 100. = Level 2 Routing Message
DRP:      .... ...1 = Control Packet Format
DRP: Control Packet Type = 04
DRP: Source Node      = 31.501
DRP: Reserved field = 0
DRP: RTGINFO count = 63
DRP: Start Area = 1
DRP: Area = 1      Hops = 31      Cost = 1023
DRP: Area = 2      Hops = 31      Cost = 1023
DRP: Area = 3      Hops = 31      Cost = 1023
DRP: Area = 4      Hops = 31      Cost = 1023
DRP: Area = 5      Hops = 31      Cost = 1023
DRP: Area = 6      Hops = 31      Cost = 1023
DRP: Area = 7      Hops = 31      Cost = 1023
DRP: Area = 8      Hops = 31      Cost = 1023
DRP: Area = 9      Hops = 31      Cost = 1023
DRP: Area = 10     Hops = 31      Cost = 1023
DRP: Area = 11     Hops = 31      Cost = 1023
DRP: Area = 12     Hops = 31      Cost = 1023
DRP: Area = 13     Hops = 31      Cost = 1023
DRP: Area = 14     Hops = 31      Cost = 1023
DRP: Area = 15     Hops = 31      Cost = 1023
DRP: Area = 16     Hops = 31      Cost = 1023
DRP: Area = 17     Hops = 31      Cost = 1023
DRP: Area = 18     Hops = 31      Cost = 1023
DRP: Area = 19     Hops = 31      Cost = 1023
DRP: Area = 20     Hops = 31      Cost = 1023
DRP: Area = 21     Hops = 31      Cost = 1023
DRP: Area = 22     Hops = 31      Cost = 1023
DRP: Area = 23     Hops = 31      Cost = 1023
DRP: Area = 24     Hops = 31      Cost = 1023
DRP: Area = 25     Hops = 31      Cost = 1023
DRP: Area = 26     Hops = 31      Cost = 1023
DRP: Area = 27     Hops = 31      Cost = 1023
DRP: Area = 28     Hops = 31      Cost = 1023
DRP: Area = 29     Hops = 31      Cost = 1023
DRP: Area = 30     Hops = 31      Cost = 1023
DRP: Area = 31     Hops = 00      Cost = 0
DRP: Area = 32     Hops = 31      Cost = 1023
DRP: Area = 33     Hops = 31      Cost = 1023
DRP: Area = 34     Hops = 31      Cost = 1023
DRP: Area = 35     Hops = 31      Cost = 1023
DRP: Area = 36     Hops = 31      Cost = 1023
DRP: Area = 37     Hops = 31      Cost = 1023
DRP: Area = 38     Hops = 31      Cost = 1023
DRP: Area = 39     Hops = 31      Cost = 1023

```

```

DRP: Area = 40      Hops = 31      Cost = 1023
DRP: Area = 41      Hops = 31      Cost = 1023
DRP: Area = 42      Hops = 31      Cost = 1023
DRP: Area = 43      Hops = 31      Cost = 1023
DRP: Area = 44      Hops = 31      Cost = 1023
DRP: Area = 45      Hops = 31      Cost = 1023
DRP: Area = 46      Hops = 31      Cost = 1023
DRP: Area = 47      Hops = 31      Cost = 1023
DRP: Area = 48      Hops = 31      Cost = 1023
DRP: Area = 49      Hops = 31      Cost = 1023
DRP: Area = 50      Hops = 31      Cost = 1023
DRP: Area = 51      Hops = 31      Cost = 1023
DRP: Area = 52      Hops = 31      Cost = 1023
DRP: Area = 53      Hops = 31      Cost = 1023
DRP: Area = 54      Hops = 31      Cost = 1023
DRP: Area = 55      Hops = 31      Cost = 1023
DRP: Area = 56      Hops = 31      Cost = 1023
DRP: Area = 57      Hops = 31      Cost = 1023
DRP: Area = 58      Hops = 31      Cost = 1023
DRP: Area = 59      Hops = 31      Cost = 1023
DRP: Area = 60      Hops = 31      Cost = 1023
DRP: Area = 61      Hops = 01      Cost = 20
DRP: Area = 62      Hops = 31      Cost = 1023
DRP: Area = 63      Hops = 31      Cost = 1023
DRP:
DRP: Checksum = 8436
DRP:

```

ADDR	HEX	ASCII
0000	AB 00 00 03 00 00 AA 00 04 00 F5 7D 60 03 88 00}~...
0010	09 F5 7D 00 3F 00 01 00 FF 7F FF 7F FF 7F FF 7F	..}.?.....
0020	FF 7F FF 7F FF 7F FF 7F FF 7F 7F FF 7F FF 7F
0030	FF 7F FF 7F FF 7F FF 7F FF 7F 7F FF 7F FF 7F
0040	FF 7F FF 7F FF 7F FF 7F FF 7F 7F FF 7F FF 7F
0050	FF 7F FF 7F 00 00 FF 7F FF 7F FF 7F FF 7F FF 7F
0060	FF 7F FF 7F FF 7F FF 7F FF 7F 7F FF 7F FF 7F
0070	FF 7F FF 7F FF 7F FF 7F FF 7F 7F FF 7F FF 7F
0080	FF 7F FF 7F FF 7F FF 7F FF 7F 7F FF 7F FF 7F
0090	14 04 FF 7F FF 7F 36 846.

B.5 LAT

Il LAT (Local Area Transport) è il classico esempio di protocollo che non può essere trattato tramite router, ma solo tramite bridge, in quanto sprovvisto del livello 3. Infatti la busta LAT è contenuta direttamente nella busta Ethernet. In questo caso

il pacchetto LAT contiene un solo carattere "\", probabilmente proveniente da un terminale VT100 collegato ad un terminal server.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 29 arrived at 00:48:06.5073; frame size is 60 (003C hex)
    bytes.
DLC: Destination = Station DEC 3761DF, CENSA1
DLC: Source       = Station DEC 2A7463
DLC: Ethertype    = 6004 (DEC LAT)
DLC:
LAT: ----- Local Area Transport -----
LAT:
LAT: Message type / flags = 02
LAT:      0000 00.. = Run
LAT:      .... ..1. = To host
LAT:      .... ...0 = No response requested
LAT: Number of slots = 1
LAT:   Destination link ID = 5F01
LAT:   Source link ID = B802
LAT:   Sequence number = 71
LAT: Acknowledgement number = CA
LAT:
LAT: ----- Local Area Transport Data To Host (Slot 0) -----
LAT:
LAT: Destination sublink ID = 26
LAT:   Source sublink ID = 21
LAT: Data length = 1
LAT: Slot type / Credits = 00
LAT:      0000 .... = Data
LAT:      .... 0000 = 0 Credits
LAT: Data = "\"
LAT:

ADDR  HEX                                     ASCII
0000  08 00 2B 37 61 DF 08 00 2B 2A 74 63 60 04 02 01  ..+7a...+*tc`...
0010  01 5F 02 B8 71 CA 26 21 01 00 5C 4F 00 00 00 00  ._...q.&!...O....
0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....

```

B.6 ISO CLNS

I seguenti pacchetti appartengono al livello 3 ISO CLNS (connectionless). ISO CLNS è un insieme di protocolli che comprende ISO 8473 (Protocol ID = 81H), ISO 9542 ES-IS (Protocol ID = 82H) e ISO 10598 IS-IS (Protocol ID = 83H).

B.6.1 ISO Session Layer

Il seguente pacchetto, di tipo "give tokens", appartiene al Session OSI ed è imbustato in un pacchetto di Transport OSI (ISO_TP), imbustato in un pacchetto ISO 8473 (livello 3 OSI), imbustato in 802.2 (LLC), imbustato in 802.3. Si noti che ISO CLNS richiede sempre la presenza di LLC (SAP = FE) e quindi il pacchetto non può essere di tipo Ethernet, ma deve essere di tipo 802.3. A livello 3, gli indirizzi usati sono del tipo Local Character format, cioè stringhe di caratteri.

```

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 8 arrived at  10:15:50.1488; frame size is 101 (0065 hex)
      bytes.
DLC:  Destination = Station U-B   38F200
DLC:  Source      = Station DG    010400
DLC:  802.3 length = 87
DLC:
LLC:  ----- LLC Header -----
LLC:
LLC:  DSAP = FE, SSAP = FE, Command, Unnumbered frame: UI
LLC:
CLNP: ----- ISO Network Layer -----
CLNP:
CLNP: Protocol ID = 81 (ISO Connectionless Network Protocol)
CLNP: Header length = 57
CLNP: Version / Protocol ID extension = 01
CLNP: Remaining PDU lifetime is 1.5 seconds
CLNP: Flags/type byte = 9C
CLNP:      1... .... = Segmentation permitted
CLNP:      .0.. .... = Last segment
CLNP:      ..0. .... = Error not reported if frame discarded
CLNP:      ...1 1100 = Data PDU
CLNP: Segment length = 84
CLNP: Checksum = 0000
CLNP: Destination address: "Ppci" (Local Character format)
CLNP:      Source address: "P10MAN2" (Local Character format)
CLNP: Data unit identifier = 0008
CLNP: Segment offset = 0
CLNP: Total length = 84
CLNP: Padding: 21 bytes
CLNP:
ISO_TP: ----- ISO Transport Layer -----
ISO_TP:
ISO_TP: Header length = 4
ISO_TP: TPDU type = F (Data)
ISO_TP: Destination reference = 0001
ISO_TP: EOT: Last TPDU of a sequence

```



```
ISO_TP: Send sequence number = 2
ISO_TP:
SESS: ----- ISO Session Layer -----
SESS:
SESS: SPDU type = 1 (Give Tokens)
SESS: Length of SPDU parameter field = 1
SESS:
SESS: Unknown parameter code = 0
SESS: Parameter length = 0
SESS: SPDU type = 0 (Exception Report)
SESS: SPDU type = 0 (Exception Report)
SESS: Length of SPDU parameter field = 4
SESS:
SESS: Unknown parameter code = 0
SESS: Parameter length = 22
```

ADDR	HEX	ASCII
0000	00 DD 00 38 F2 00 08 00 1B 01 04 00 00 57 FE FE	...8.....W..
0010	03 81 39 01 03 9C 00 54 00 00 0A 50 70 63 69 00	..9....T...Ppci.
0020	DD 00 38 F2 00 07 50 31 30 4D 41 4E 32 00 08 00	..8...P10MAN2...
0030	00 00 54 CC 15 00 00 00 00 00 00 00 00 00 00	..T.....
0040	00 00 00 00 00 00 00 00 00 00 04 F0 00 01 82 01
0050	01 00 00 00 00 00 04 00 16 00 69 34 27 00 00 00i4'...
0060	00 8D 27 38 00	..'8.

B.6.2 ISO SMB

Il seguente pacchetto è un SMB imbustato in ISO_TP, imbustato in ISO 8473, imbustato in IEEE 802.2, imbustato in IEEE 802.3. La sintassi usata per gli indirizzi di livello 3 è Local Binary format. Si noti che l'indirizzo di livello 2 MAC è contenuto nell'indirizzo di livello 3.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 2 arrived at 16:26:09.4999; frame size is 126 (007E hex)
    bytes.
DLC: Destination = Station AT&T 010271
DLC: Source       = Station AT&T 0100A9
DLC: 802.3 length = 112
DLC:
LLC: ----- LLC Header -----
LLC:
LLC: DSAP = FE, SSAP = FE, Command, Unnumbered frame: UI
LLC:
CLNP: ----- ISO Network Layer -----
CLNP:
```

```

CLNP: Protocol ID = 81 (ISO Connectionless Network Protocol)
CLNP: Header length = 49
CLNP: Version / Protocol ID extension = 01
CLNP: Remaining PDU lifetime is 3.0 seconds
CLNP: Flags/type byte = 9C
CLNP:      1... .... = Segmentation permitted
CLNP:      .0.. .... = Last segment
CLNP:      ..0. .... = Error not reported if frame discarded
CLNP:      ...1 1100 = Data PDU
CLNP: Segment length = 109
CLNP: Checksum = 0000
CLNP: Destination address: 490000000000000108006A010271FE01
      (Local Binary format)
CLNP: Source address: 490000000000000108006A0100A9FE01
      (Local Binary format)
CLNP: Data unit identifier = 0427
CLNP: Segment offset = 0
CLNP: Total length = 109
CLNP:
ISO_TP: ----- ISO Transport Layer -----
ISO_TP:
ISO_TP: Header length = 4
ISO_TP: TPDU type = 6 (Ack )
ISO_TP: Destination reference = 0002
ISO_TP: Next expected sequence number = 35
ISO_TP: Credit value = 7
ISO_TP:
ISO_TP: ----- ISO Transport Layer -----
ISO_TP:
ISO_TP: Header length = 4
ISO_TP: TPDU type = F (Data)
ISO_TP: Destination reference = 0002
ISO_TP: EOT: Last TPDU of a sequence
ISO_TP: Send sequence number = 82
ISO_TP:
SMB: ----- SMB Delete File Command -----
SMB:
SMB: Function = 06 (Delete File)
SMB: Tree id      (TID) = 6000
SMB: Process id   (PID) = 2D24
SMB: File pathname = "\COPYDEST.4"
SMB: Attribute flags = 0006
SMB:      .... .... ..0. .... = File(s) not changed since last archive
SMB:      .... .... ...0 .... = No directory file(s)
SMB:      .... .... .... 0... = No volume label info
SMB:      .... .... .... .1.. = System file(s)
SMB:      .... .... .... ..1. = Hidden file(s)
SMB:      .... .... .... ...0 = No read only file(s)
SMB:

```

ADDR	HEX	ASCII
0000	08 00 6A 01 02 71 08 00 6A 01 00 A9 00 70 FE FE	..j..q..j....p..
0010	03 81 31 01 06 9C 00 6D 00 00 10 49 00 00 00 00	..1....m...I....
0020	00 00 01 08 00 6A 01 02 71 FE 01 10 49 00 00 00j..q...I...
0030	00 00 00 01 08 00 6A 01 00 A9 FE 01 04 27 00 00j.....'...
0040	00 6D 04 67 00 02 23 04 F0 00 02 D2 FF 53 4D 42	.m.g..#.....SMB
0050	06 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060	00 00 00 00 00 60 24 2D 00 00 00 00 01 06 00 0D`\$-.....
0070	00 04 5C 43 4F 50 59 44 45 53 54 2E 34 00	..\\COPYDEST.4.

B.6.3 ISH

In questo caso ci troviamo in presenza di un pacchetto ISO 9542 ES-IS di tipo ISH (Intermediate System Hello). Si tratta cioè di un router OSI (IS: Intermediate System) che invia in multicast la sua identità agli End System (ES) affinché la inseriscano nella loro router cache locale.

```

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 167 arrived at  00:48:10.0241; frame size is 60 (003C hex)
      bytes.
DLC:  Destination = Multicast 09002B000004
DLC:  Source       = Station DECnet00F57D
DLC:  802.3 length = 29
DLC:
LLC:  ----- LLC Header -----
LLC:
LLC:  DSAP = FE, SSAP = FE, Command, Unnumbered frame: UI
LLC:
CLNP: ----- ISO Network Layer -----
CLNP:
CLNP: Protocol ID = 82 (ISO Routing Exchange Protocol)
CLNP: Header length = 26
CLNP: Version / Protocol ID extension = 01
CLNP: PDU type: Intermediate System Hello (ISH)
CLNP: Holding time is 30 seconds
CLNP: Checksum = 0000
CLNP: Network entity title: 470020001FAA000400F57D00
      (ISO 6523-ICO Binary format)
CLNP: Suggested ES configuration timer = 600
CLNP:
DLC:  Frame padding= 17 bytes

```

ADDR	HEX	ASCII
0000	09 00 2B 00 00 04 AA 00 04 00 F5 7D 00 1D FE FE	..+.....}....
0010	03 82 1A 01 00 04 00 1E 00 00 0C 47 00 20 00 1FG. ..

```

0020  AA 00 04 00 F5 7D 00 C6 02 02 58 00 00 00 00 00 .....}....X.....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

B.7 SNAP-PDU

Le SNAP-PDU sono delle particolari PDU di livello 2 LLC che servono ad imbustare nel campo dati LLC un protocollo non standard. LLC-DSAP e LLC-SSAP assumono il valore esadecimale AA e dopo il campo control sono inseriti 3 byte di OUI e 2 byte di protocol type.

B.7.1 AppleTalk

Nel pacchetto che segue vediamo che lo SNAP header contiene un vendor ID o OUI=080007, corrispondente alla ditta Apple, e un protocol type o SNAP type = 809B, corrispondente al protocollo AppleTalk.

```

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 30 arrived at  00:03:36.9210; frame size is 60 (003C hex)
      bytes.
DLC:  Destination = Multicast 090007FFFFFF
DLC:  Source       = Station 0080D300146D
DLC:  802.3 length = 43
DLC:
LLC:  ----- LLC Header -----
LLC:
LLC:  DSAP = AA, SSAP = AA, Command, Unnumbered frame: UI
LLC:
SNAP:  ----- SNAP Header -----
SNAP:
SNAP:  Vendor ID = 080007 (Apple)
SNAP:  Type = 809B (AppleTalk)
SNAP:  [35 byte(s) of data]
DLC:  Frame padding= 3 bytes

```

ADDR	HEX	ASCII
0000	09 00 07 FF FF FF 00 80 D3 00 14 6D 00 2B AA AAm.+..
0010	03 08 00 07 80 9B 00 23 00 00 00 00 03 A5 FF CB#.....
0020	01 01 01 03 A5 08 CB 03 84 80 03 E8 82 03 84 80
0030	03 E8 82 03 E9 00 03 EB 00 00 00 00

B.7.2 DEC MOP

Il MOP è il Maintenance Operation Protocol della Digital. Il suo header SNAP contiene un OUI = 08002B corrispondente a Digital Eq. Corp. e un protocol type = 6002 corrispondente al protocollo DEC MOP RC.

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 6 arrived at 00:09:05.8256; frame size is 73 (0049 hex)
      bytes.
DLC: Destination = Multicast AB0000020000
DLC: Source       = Station DEC 3761DF, CENSA1
DLC: 802.3 length = 59
DLC:
LLC: ----- LLC Header -----
LLC:
LLC: DSAP = AA, SSAP = AA, Command, Unnumbered frame: UI
LLC:
SNAP: ----- SNAP Header -----
SNAP:
SNAP: Vendor ID = 08002B
SNAP: Type = 6002 (DEC MOP RC)
SNAP: [51 byte(s) of data]
```

ADDR	HEX	ASCII
0000	AB 00 00 02 00 00 08 00 2B 37 61 DF 00 3B AA AA+7a..i..
0010	03 08 00 2B 60 02 07 00 00 00 01 00 03 04 00 00	...+`.....
0020	02 00 02 41 00 64 00 01 AA 90 01 01 01 07 00 06	...A.d.....
0030	08 00 2B 37 61 DF 0A 00 10 C0 6D 0D D3 BF 7B CD	..+7a.....m...{.
0040	01 FF FF FF FF FF FF 3C 10<.

B.8 NOVELL NETWARE

Il protocollo Novell Netware usa a livello 3 il protocollo IPX derivato da Xerox XNS. La caratteristica atipica di IPX è quella di poter usare diversi tipi di imbustamento a livello 2. Novell può infatti sfruttare Ethernet, ed allora indica un protocol type uguale a 8137, ma può sfruttare anche 802.3 senza usare la busta 802.2.

Questa seconda possibilità è evidenziata nei due esempi che seguono.

B.8.1 Open file

Una busta 802.3 dovrebbe sempre contenere una busta 802.2. Questo nel caso di

IPX non si verifica, in quanto dopo il campo length della busta MAC 802.3 troviamo subito il pacchetto IPX. Questa eccezione viene discriminata in ricezione in quanto la busta IPX inizia sempre con 16 bit detti di checksum, ma posti sempre uguali a 1. Se la busta IPX venisse interpretata come busta LLC si avrebbe un LLC-SSAP uguale a FF, cioè broadcast che è chiaramente impossibile.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 5 arrived at 09:23:17.6181; frame size is 72 (0048 hex)
      bytes.
DLC: Destination = Station 3Com 7AEC53
DLC: Source      = Station 3Com 7AEC6E
DLC: 802.3 length = 58
DLC:
XNS: ----- XNS Header -----
XNS:
XNS: Checksum = FFFF
XNS: Length = 57
XNS: Transport control = 00
XNS:      0000 .... = Reserved
XNS:      .... 0000 = Hop count
XNS: Packet type = 17 (Novell NetWare)
XNS:
XNS: Dest net = 00000001, host = 000000000001 (POLI-LAIB),
      socket = 1105 (NetWare Server)
XNS: Source net = 00001000, host = 02608C7AEC6E, socket = 16387
      (4003)
XNS:
XNS: ----- Novell Advanced NetWare -----
XNS:
XNS: Request type = 2222 (Request)
XNS: Seq no=14 Connection no=1 Task no=7
XNS:
NCP: ----- Open File Request -----
NCP:
NCP: Request code = 76
NCP:
NCP: Dir handle = 03
NCP: Search attribute flags = 06
NCP:      .... .1.. = System files allowed
NCP:      .... ..1. = Hidden files allowed
NCP: Desired access rights = 13
NCP:      000. .... = Not defined
NCP:      ...1 .... = Exclusive (single-user mode)
NCP:      .... 0... = Allow others to open for writing
NCP:      .... .0.. = Allow others to open for reading
NCP:      .... ..1. = Open for writing
NCP:      .... ...1 = Open for reading

```

```
NCP: File name = "/MGAI/DATA01.LOG"
NCP:
NCP: [Normal end of NetWare "Open File Request" packet.]
NCP:
```

ADDR	HEX	ASCII
0000	02 60 8C 7A EC 53 02 60 8C 7A EC 6E 00 3A FF FF	.`.z.S.`.z.n....
0010	00 39 00 11 00 00 00 01 00 00 00 00 01 04 51	.9.....Q
0020	00 00 10 00 02 60 8C 7A EC 6E 40 03 22 22 0E 01`.z.n@."".
0030	07 00 4C 03 06 13 10 2F 4D 47 41 49 2F 44 41 54	..L..../MGAI/DAT
0040	41 30 31 AE 4C 4F 47 49	A01.LOGI

B.8.2 Login request

Il pacchetto seguente contiene una login request di un client Novell su un server Novell, imbustata in un protocollo NCP, imbustata in IPX, imbustata direttamente in 802.3. Si notino gli indirizzi IPX che sono gerarchici su due livelli (net e host).

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 9 arrived at 13:58:56.6606; frame size is 66 (0042 hex)
    bytes.
DLC: Destination = Station 3Com 217692
DLC: Source       = Station 3Com 119421
DLC: 802.3 length = 52
DLC:
XNS: ----- XNS Header -----
XNS:
XNS: Checksum = FFFF
XNS: Length = 51
XNS: Transport control = 00
XNS:      0000 .... = Reserved
XNS:      .... 0000 = Hop count
XNS: Packet type = 17 (Novell NetWare)
XNS:
XNS: Dest  net = 00217692, host = 02608C217692, socket = 1105
    (NetWare Server)
XNS: Source net = 00217692, host = 02608C119421, socket = 16385
    (4001)
XNS:
XNS: ----- Novell Advanced NetWare -----
XNS:
XNS: Request type = 2222 (Request)
XNS: Seq no=78   Connection no=4   Task no=1
XNS:
NCP: ----- Login Request -----
```

```

NCP:
NCP:   Request/sub-function code = 23,0
NCP:
NCP:   Name = "DAN"
NCP:   Password = "GLIDE*"
NCP:
NCP:   [Normal end of NetWare "Login Request" packet.]
NCP:

ADDR  HEX                                     ASCII
0000  02 60 8C 21 76 92 02 60 8C 11 94 21 00 34 FF FF  .`.!v..`...!.4..
0010  00 33 00 11 00 21 76 92 02 60 8C 21 76 92 04 51  .3...!v..`.!v..Q
0020  00 21 76 92 02 60 8C 11 94 21 40 01 22 22 4E 04  .!v..`...!@."N.
0030  01 00 17 00 0C 00 03 44 41 4E 06 47 4C 49 44 45  .....DAN.GLIDE
0040  2A 00                                           *.

```

B.9 VINES

Banyan Vines è un sistema operativo di rete concorrente di Novell Netware e di Microsoft LAN Manager.

B.9.1 Mail service

Il pacchetto seguente contiene in una busta Ethernet il livello 3 Vines IP, che contiene una busta VIPC (Vines Interprocess Communications Protocol), che contiene una busta Matchmaker, che contiene una busta di mail service.

```

DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 2 arrived at 15:08:02.7842; frame size is 110 (006E hex)
      bytes.
DLC:  Destination = Station 0074003D0000
DLC:  Source       = Station 3Com 177746
DLC:  Ethertype    = 0BAD (Banyan VINES)
DLC:
VIP:  ----- VINES IP Header -----
VIP:
VIP:  Checksum = FFFF (Null checksum)
VIP:  Packet length = 96
VIP:
VIP:  Transport control = 1F
VIP:           00.. .... = Unused
VIP:           ..0. .... = Do not return metric notification packet

```



```

VIP:      ...1 .... = Return exception notification packet
VIP:      .... 1111 = Hop count remaining (15)
VIP:
VIP: Protocol type = 1 (Interprocess Communications Protocol - VIPC)
VIP:
VIP: Destination network.subnetwork = 01000003.0001
VIP: Source network.subnetwork      = 001E86AF.80DE
VIP:
VIPC: ----- VINES IPC Header -----
VIPC:
VIPC: Source port          = 025D
VIPC: Destination port    = 0004
VIPC:
VIPC: Packet type = 1 (Data)
VIPC:
VIPC: Control = 60
VIPC: 0... .... = Do not send immediate acknowledgment
VIPC: .1.. .... = End of message
VIPC: ..1. .... = Beginning of message
VIPC: ...0 .... = Do not abort current message
VIPC: .... 0000 = Unused
VIPC:
VIPC: Source connection ID      = 0111
VIPC: Destination connection ID = 002D
VIPC:
VIPC: Sequence number          = 7
VIPC: Acknowledgment number    = 6
VIPC:
VIPC: Length = 62
VIPC:
MATCH: ----- VINES MATCHMAKER Header -----
MATCH:
MATCH: Packet type = 0 (Call)
MATCH:
MATCH: Transaction ID          = 0
MATCH: Program number          = 0
MATCH: Version number          = 1
MATCH: Procedure value         = 4
MATCH: Procedure arguments = 81 00 06 8A 00 1F . . .
MATCH:
MAIL: ----- VINES MAIL SERVICE Header -----
MAIL:
MAIL: Matchmaker packet type = 0 (Call)
MAIL: Procedure value = 4 (Any Changes?)
MAIL:
MAIL:

ADDR  HEX                                     ASCII
0000  00 74 00 3D 00 00 02 60 8C 17 77 46 0B AD FF FF .t.=...`..wF....

```

```

0010  00 60 1F 01 01 00 00 03  00 01 00 1E 86 AF 80 DE  .`.
0020  02 5D 00 04 01 60 01 11  00 2D 00 07 00 06 00 3E  .]...`...-.....>
0030  00 00 00 00 00 00 00 00  00 01 00 04 81 00 06 8A  .....
0040  00 1F 50 61 75 6C 20 4A  61 63 6B 73 6F 6E 40 33  ..Paul Jackson@3
0050  38 36 2B 50 43 58 40 43  6F 6E 76 65 72 67 65 6E  86+PCX@Convergen
0060  74 00 00 07 47 65 6E 65  72 61 6C 00 D1 BB      t...General...

```

B.10 BRIDGE PDU

Le Bridge PDU sono le PDU generate dal protocollo spanning tree dei bridge (capitolo 10). Nel caso in esame si tratta di una PDU 802.3 contenente una PDU 802.2 con LLC-SAP uguale a 42 esadecimale. La PDU 802.3 è inviata in multicast al MAC address 01-80-C2-00-00-00.

In questo caso la Bridge PDU è stata catturata su una LAN direttamente connessa al root bridge, come appare evidente dal fatto che gli indirizzi del root bridge (root bridge ID) e del bridge che ha trasmesso la PDU (sending bridge ID) sono identici.

In particolare la LAN su cui è stata catturata la PDU era connessa alla porta 3 del bridge.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 1 arrived at  00:01:44.1485; frame size is 60 (003C hex)
      bytes.
DLC: Destination = Multicast 0180C2000000
DLC: Source       = Station DEC  28C724
DLC: 802.3 length = 38
DLC:
LLC: ----- LLC Header -----
LLC:
LLC: DSAP = 42, SSAP = 42, Command, Unnumbered frame: UI
LLC:
BPDU: ----- Bridge Protocol Data Unit Header -----
BPDU:
BPDU: Protocol Identifier = 0000
BPDU: Protocol Version   = 00
BPDU:
BPDU: BPDU Type == 00 (Configuration)
BPDU:
BPDU: BPDU Flags = 00
BPDU: 0... .. = Not Topology Change Acknowledgment
BPDU: .000 0000 = Unused
BPDU:
BPDU: Root Identifier   = 0080.DEC  28C723
BPDU: Priority          = 0080

```

```

BPDU:  MAC Address      = DEC   28C723
BPDU:
BPDU:  Root Path Cost   = 0
BPDU:
BPDU:  Sending Bridge Id = 0080.DEC   28C723.0003
BPDU:  Priority          = 0080
BPDU:  MAC Address      = DEC   28C723
BPDU:  Port              = 0003
BPDU:
BPDU:  Message Age       = 0.0        seconds
BPDU:  Information Lifetime = 1920.0
BPDU:  Root Hello Time   = 128.0
BPDU:  Forward Delay     = 1920.0
BPDU:
DLC:  Frame padding= 8 bytes
    
```

ADDR	HEX	ASCII
0000	01 80 C2 00 00 00 08 00 2B 28 C7 24 00 26 42 42+ (. \$. & B B
0010	03 00 00 00 00 00 00 80 08 00 2B 28 C7 23 00 00+ (. # . .
0020	00 00 00 80 08 00 2B 28 C7 23 00 03 00 00 0F 00+ (. #
0030	01 00 0F 00 00 00 00 00 00 00 00 00

BIBLIOGRAFIA

- [1] Network General, "Sniffer Manuals", Network General, Menlo Park CA (USA).
- [2] Reynolds, J. Postel, "RFC 1340: Assigned Numbers", 07/10/1992.
- [3] IBM, "Token-Ring Network: Architecture Reference", Pub. No. SC30-3374-01, second edition, August 1987.
- [4] Cisco Systems, "Internetworking Technology Overview", Codice documento DOC-ITO13 78-1070-01, 1993.
- [5] Cisco Systems, "Router Products: Configuration and Reference", Codice documento DOC-R9.1 78-0959-01, September 1992, Vol. I, II, III.
- [6] Martin, J. Leben, "DECnet Phase V: An OSI Implementation", Digital Press, Bedford MA (USA), 1992.
- [7] R. Perlman, "Interconnections: Bridges and Routers", Addison-Wesley, Reading MA (USA), 1992.
- [8] Postel, J. Reynolds, "RFC 1042: A Standard for the Transmission of IP Datagrams over IEEE 802 Networks", February 1988.