

DISTNET
PROGRESS REPORT NO.1

Dipartimento di Informatica, Sistemistica e Telematica

Marzo 1985

Introduzione

Questo vuole essere il primo di una serie di "report" informativi sullo stato di avanzamento di quell'oggetto misterioso di cui molti parlano e di cui nessuno sa (quasi) nulla: **DistNet**.

Sono convinto che la massima diffusione delle informazioni sia essenziale sia per sollecitare un dibattito su un tema che assumerà probabilmente un'importanza "strategica" nell'ambito del Dipartimento, sia per sgombrare il campo da facili "misunderstanding" che possono nascere a proposito delle reti locali.

Mi sono proposto di informare periodicamente sullo stato di avanzamento dei lavori, in modo da un lato da costringere chi prende parte attiva alla cosa ad lasciare traccia del lavoro svolto, dall'altro da invogliare tutti i possibili interessati a diventare parti attive.

Joy Marino

Hanno contribuito al DistNet Progress Report #1:

Franco Davoli
Cristina Capurro
Lino Laganà
Joy Marino

DistNet Progress Report

1. Phase 0

Nel corso del 1983 è stato messo a punto un piccolo prototipo di rete locale che interconnetteva tramite linee seriali a bassa velocità un Vax 11/750, un PDP 11/23 e (saltuariamente) un PDP 11/34, tutti funzionanti sotto sistema operativo Unix (4.1BSD per il Vax, 2.8BSD per i PDP). Il protocollo di comunicazione utilizzato si basava su una versione particolarmente ottimizzata del software "Berknet" realizzato a Berkeley.

I servizi offerti da Berknet (ben noti ai pochissimi che ne fecero uso) consistevano essenzialmente in file transfer in modo "spooled", cioè non immediato, e mailing.

Nel luglio 1984 è stata finalmente installata la versione 4.2BSD di Unix sul VAX 11/750 del Dipartimento, l'elemento software indispensabile come ambiente in cui sviluppare la vera rete locale basata su hardware Ethernet.

Quando, in settembre, l'installazione si è consolidata per quanto riguarda il sistema a sé stante (cioè senza far uso di supporto di rete), alcuni fatti sono apparsi chiari:

- 1) il software detto "Berknet" non era più distribuito nella nuova release perchè considerato obsoleto;
- 2) i driver software disponibili supportavano numerosi hardware per rete locale (fra cui i controller Ethernet 3Com e Interlan, presi in considerazione a suo tempo quando si decise di acquistare hardware per Ethernet), ma non esistevano driver per hardware Digital (i controller DEUNA e DEQNA che erano stati nel frattempo acquistati);
- 3) la versione di Unix che era stata messa a punto faticosamente sui PDP era decisamente incompatibile con le primitive di comunicazione introdotte nella nuova versione di Unix per il Vax;
- 4) il kernel di 4.2BSD esteso con le primitive di comunicazione era un ordine di grandezza più complesso del kernel della precedente versione, 4.1BSD, e quindi era difficile comprenderne gli "internals" e i meccanismi di funzionamento;
- 5) non era chiaro che cosa si poteva fare con le primitive di cui sopra, ne che tipo di configurazione fosse richiesto.

Il quadro d'insieme era tanto sconcertante da mettere in dubbio agli occhi di uno dei principali interessati (Marino) la realizzabilità di una soluzione accettabile. Una pausa di riflessione era comunque necessaria, e quindi i quattro mesi successivi sono stati dedicati allo studio delle possibili alternative per uscire dall'impasse.

2. Phase One

2.1. Hardware

Alla fine di gennaio 1985 è stato installato l'hardware Ethernet secondo la seguente configurazione.

Una cavo coassiale Ethernet di 70 mt. circa corre nella canaletta lato laboratori del secondo e terzo piano del Nuovo Edificio e raggiunge, tramite il cavedio centrale, il piano "-1". Su tale cavo sono collegati sei transceiver Digital H4000 che collegano rispettivamente:

- 1) una interfaccia DEUNA per il PDP11/34 del laboratorio DIST NE3S;
- 2) una interfaccia DEUNA per il Vax "dist&dibe";
- 3) una interfaccia DEQNA per il PDP11/23 del laboratorio DIST NE3N;
- 4) una interfaccia DEUNA per il PDP11/24 del laboratorio DIST NE2;
- 5) una interfaccia DEQNA per il micro11 del laboratorio DIBE, NE2;
- 6) una interfaccia DEUNA per il Vax "dibe-vlsi", NE-1.

Un ulteriore transceiver Digital è previsto per l'interfaccia DEQNA del microVAX del laboratorio DIST NE3N.

2.2. Software

Per quanto riguarda il supporto di sistema operativo, occorre distinguere almeno tre attività tra loro collegate: le due versioni di Unix per Vax e PDP, i driver software.

2.2.1. Unix su Vax La versione di Unix 4.2BSD è attualmente configurata per il funzionamento in rete, anche se non esiste ancora il driver di supporto dell'hardware Digital; questo significa che le dimensioni del kernel e l'overhead di sistema attualmente validi saranno gli stessi (o quasi) anche in una futura situazione di interconnessione attiva. Inoltre è possibile condurre alcuni limitati esperimenti di uso delle primitive di rete ("sockets", etc) utilizzando un software loopback locale al sistema.

Per quanto riguarda il MicroVax con Ultrix, si può dire paradossalmente che questo è attualmente l'unico nodo effettivamente connesso in rete: infatti Ultrix V32 è perfettamente identico a Unix 4.2BSD, quindi supporta le stesse system calls ed è già dispone dei servers per i servizi di rete. Inoltre il kernel dispone del driver software necessario per gestire la DEQNA di interfaccia verso l'Ehertnet. Un secondo MicroVax connesso in rete potrebbe essere immediatamente operativo e colloquiare con il primo.

2.2.2. Unix su PDP E' stata acquisita una versione di Unix per il PDP (2.9BSD) che era detta essere compatibile con i supporti di rete validi per la 4.2BSD; in realtà questo si è rilevato non proprio vero. Infatti, se da un lato la versione 2.9BSD rappresenta un significativo miglioramento della precedente versione per quanto

DistNet Progress Report

riguarda il sistema a sè stante, il supporto di rete appare essere ancora in una forma sperimentale, addirittura mancante di alcuni moduli essenziali.

Si è provveduto quindi alla stesura dei moduli mancanti, utilizzando lo Unix del Vax come riferimento; il risultato finale è un kernel Unix che supporta le stesse primitive di rete del 4.2BSD (salvo alcune limitazioni minori). Le dimensioni del nucleo di sistema così ottenuto (circa 130 kbytes) sono però tali da occupare circa metà della memoria disponibile sulle macchine più piccole (11/23, 11/34); per un'utilizzazione efficiente sarebbe meglio disporre di un PDP11/73 con 512k-1Mbyte di memoria.

2.2.3. Drivers Si stanno esplorando alcune strade alternative per ottenere i driver software che supportino l'hardware Ethernet Digital. Tre sono le strade alternative utilizzate: acquisire dalla Digital in forma "source" i drivers per DEUNA e DEQNA che dovrebbero esistere sotto ULTRIX (la versione di Unix distribuita da Digital per Vax e MicroVax), e trasportarli sia su Vax con 4.2BSD (data la dichiarata compatibilità tra ULTRIX e 4.2BSD dovrebbe essere immediato), sia su PDP con 2.9BSD (qualche problema in più, di non difficile soluzione). In alternativa si intendono prendere contatti con Berkeley per sapere se tali drivers siano già stati realizzati.

L'ultima possibilità consiste nella stesura ex-novo del software necessario. La stesura di un driver minimale per DEQNA sembra essere abbastanza facile, mentre l'interfaccia DEUNA appare decisamente più complessa da gestire. In ogni caso, il know-how ottenibile lavorando a questo livello sarà un patrimonio importante per tutti i futuri sviluppi di Distnet.

3. I sette magnifici livelli

I "magnifici sette" sono i livelli definiti nel modello ISO-OSI per reti locali. Rappresentano la struttura formale più diffusa per descrivere i servizi offerti da una rete, e perciò verranno qui presi come riferimento per descrivere le funzionalità della futura DistNet. Occorre osservare però che non sempre la struttura di supporto di rete di Unix 4.2BSD (et similia) ha una corrispondenza uno a uno con i livelli ISO-OSI.

3.1. Physical Layer

Il primo livello è risolto semplicemente dal cavo Ethernet.

3.2. Data Link Layer

Come è noto (o dovrebbe essere noto) questo è il livello più alto supportato direttamente dalle interfacce Ethernet. Il colloquio tra DEUNA e/o DEQNA è limitato semplicemente alla trasmissione e ricezione di packets contenenti dati non interpretati. Osservare che Ethernet è un mezzo non affidabile (almeno dal punto di vista della certezza delle comunicazioni) e quindi occorre risol-

vere ai livelli superiori i meccanismi di acknowledge e ritrasmissione¹.

3.3. Network Layer

Il kernel di Unix realizza in modo estensivo le funzioni di routing tra reti differenti (Ethernet, ARPANET, link diretti, altre reti). Il meccanismo di routing è piuttosto complesso e può essere adattato dinamicamente. Nel nostro caso specifico avremo probabilmente il problema di ridurre al minimo le funzionalità di questo livello, data la semplicissima configurazione (a "bus") adottata.

3.4. Transport layer

Questo livello viene risolto dal protocollo TCP/IP, che nella struttura Unix 4.2BSD compare come una delle possibili famiglie di protocolli, ma che in realtà è l'unica utilizzata in modo significativo². Con TCP/IP è possibile realizzare circuiti virtuali affidabili oppure utilizzare comunicazioni a datagrams.

Pare anche che questi protocolli si avviino a diventare uno "standard de facto" nel campo delle reti locali: sono ormai numerosi i sistemi (Unix e non) che offrono TCP/IP come supporto di rete per i livelli superiori a quelli supportati da Ethernet. Esistono anche alcune realizzazioni in hardware (Interlan) ed alcune versioni per sistemi operativi Digital (RSX soprattutto).

3.5. Session Layer

Si può inquadrare in questo livello l'integrazione dei servizi offerti dai livelli inferiori con il sistema operativo. In Unix questo è risolto con il meccanismo dei "sockets".

Un socket è analogo ad un descrittore di file, e può essere quindi utilizzato indifferentemente al posto di questo; perciò è possibile eseguire read/write su un socket che, se questo è stato preventivamente associato ad un circuito virtuale TCP/IP, può eseguire trasferimento dati da nodo a nodo in modo trasparente.

Sui sockets possono inoltre essere applicate tutta una serie di system calls specifiche (accept, connect, bind, etc.) per creare l'associazione con i diversi protocolli disponibili.

¹ La ritrasmissione in caso di collisione garantisce solo che il packet sia inviato correttamente sul mezzo fisico, ma non può garantire la corretta ricezione da parte del destinatario.

² Tra le altre famiglie previste, ma di cui non è disponibile il software di supporto, esiste anche DECNET, l'insieme dei protocolli Digital che girano sotto VMS, RSX, etc.

Il kernel di Unix è stato infine esteso con nuove primitive di supporto di comunicazione (es select), per consentire la realizzazione di comunicazioni asincrone e di "servers" distribuiti.

3.6. Presentation Layer

Il sesto livello ed il successivo rappresentano i servizi che un sistema di rete integrato può offrire. Nel livello di presentazione si possono inquadrare alcuni servizi validi per la rete ARPANET, e quindi disponibili sotto Unix anche se sostanzialmente indipendenti dal s.o.: in particolare "telnet" è un servizio di terminale virtuale che consente il collegamento su uno qualsiasi dei calcolatori connessi in rete, mentre "ftp" consente il trasferimento di files, sempre in modo indipendente dal sistema operativo ospite.

3.7. Application Layer

All'ultimo livello si possono citare i servizi che sono stati integrati completamente in Unix. I principali sono:

- **rlogin**, servizio di login remoto (o terminale virtuale);
- **rccp** esegue il trasferimento di files o intere directories tra nodi qualsiasi della rete;
- **rsh** (remote shell), esecuzione remota di comandi con possibilità di redirezione dell'input/output in locale/remoto;
- **Mail**, posta elettronica in tutta la rete, con possibilità di forwarding sulla rete UUCP utilizzando un nodo di gateway;
- **rwho**, **ruptime**, per avere una "fotografia" istantanea della rete e di chi-sta-facendo-che-cosa su tutti i nodi collegati; utile anche per conoscere la distribuzione del carico.

Tutti i comandi eseguiti su macchine remote richiedono che l'utente possieda una "login" sulla macchina collegata e che esegua una fase di identificazione (login-name, password, etc.). E' però possibile definire un insieme di macchine accessibili agli stessi utenti con gli stessi nomi di login; in questo caso non occorre identificazione e l'accesso ai servizi di rete è pressoché trasparente.

4. Servizio e Ricerca

4.1. DistNet come strumento

E' indubbio che la rete locale costituisca un elemento strategico nello sviluppo futuro del Dipartimento; basti pensare per esempio alla proliferazione dei calcolatori di laboratorio e delle workstations (auspicata), oppure al problema di rendere accessibili le risorse più critiche e costose (sistemi di presentazione di immagini, di grafica, di acquisizione da telecamera, stampanti veloci o Letter Quality, etc.).

Evidentemente, quindi, una rete locale ha funzioni di servizio all'interno della nostra struttura. Si tratta però, a mio parere, di un servizio "sui generis" per almeno un motivo; ritengo infatti che non sia possibile trovare sul mercato un prodotto adatto alle nostre esigenze. Infatti, il mercato delle reti locali è caratterizzato o da prodotti "omogenei", cioè legati ad un unico fornitore di hardware sia per quanto riguarda il supporto di rete, sia per i calcolatori (e di conseguenza per il sistema operativo), oppure da sistemi che interconnettono macchine diverse, offrendo in genere servizi molto limitati. E' significativo il fatto che anche industrie che dispongono di risorse economiche maggiori delle nostre non sappiano verso quali prodotti orientarsi.

L'oggetto "rete locale" rappresenta un prodotto non ancora stabilizzato, le cui potenzialità (e di conseguenza aspettative dei potenziali utenti) sono molto maggiori delle realizzazioni consolidate. Non a caso le realizzazioni più interessanti che conosco sono state sviluppate in Università o Centri di ricerca americani. Il "pattern" tipico che si può individuare è quello tipico: ricerca universitaria → sviluppo finanziato e finalizzato → trasferimento in piccole industrie → ingegnerizzazione e diffusione sul mercato.

Gli esempi che mi sono più familiari: Newcastle Connection, UNET, protocollo TCP/IP, per citare solo quelli che hanno un qualche legame con DistNet. Se non vado errato, questa si può chiamare ricerca applicata.

In Italia non ci sono esempi "viable" di reti locali in ambito universitario: chi fa ricerca si accontenta di carta e matita, o tutt'al più di prototipi molto sperimentali. La messa a punto ed il mantenimento di una rete locale che dia garanzie di affidabilità nell'uso giorno per giorno richiedono un impegno di tipo professionale, che non paga in ambito accademico.

Ritengo che una soluzione possa e debba essere tentata: innanzi tutto accontentandosi di un certo grado di inefficienza ed approssimazione (ad esempio come la gestione del Vax), e contemporaneamente cercando di sviluppare le competenze sui sistemi di rete locale, trasferendo il "know-how" in ditte locali, richiedendo in cambio una qualche forma di supporto ed assistenza nel medio periodo.

4.2. DistNet come ricerca

Intendo utilizzare DistNet per fare ricerca, e sono convinto che DistNet possa essere utile per fare ricerca.

A mio avviso, un elemento indispensabile per fare ricerca in informatica avanzata è la disponibilità di una buona base di partenza, per non dover sempre rincorrere chi ha più mezzi e più esperienza alle spalle (leggi Università U.S.A.); la libera circolazione di software, in questo caso per rete locale, che caratterizza

l'ambiente Unix universitario costituisce tale base, anche se un certo sforzo iniziale di integrazione è comunque necessario.

La rete locale nasce come sistema aperto; questo vale per l'hardware: ad esempio interfacce compatibili di diversi produttori, possibilità di connessioni nuove sulla rete in funzione, circuiti virtuali multipli senza interferenze di traffico, etc. Vale anche per il software, almeno nella configurazione che intendiamo realizzare: infatti sarà possibile sperimentare nuove famiglie di protocolli e nuovi servizi di rete senza alterare il software esistente, e quindi senza disturbare al traffico di routine.

Non sono interessato alla rete locale se questa è intesa solo come servizio: ritengo che ci sia spazio per argomenti informatici interessanti; ad esempio, sono interessato a questi temi:

- file system distribuito, per arrivare ad una vera integrazione tra macchine con lo stesso sistema operativo;
- sistemi "message-passing" e protocolli di chiamata a procedura remota, con riferimento alle architetture multiprocessori e al loro software di base;
- conversione di rappresentazione efficiente, è uno dei temi "di punta" nelle reti di macchine eterogenee;
- stazioni "disk-less" e terminal servers, temi di ricerca applicata.

Altre aree di ricerca che penso possano interessare: controllo del carico sulla rete e politiche di gestione, bilanciamento del carico tra calcolatori, automazione d'ufficio, fault tolerance in ambiente distribuito, database distribuiti.

5. Breve introduzione ai comandi di rete

5.1. Stato della rete

I comandi disponibili più semplici sono quelli che forniscono lo stato della rete, e sono:

```
rwho [ -a ]
ruptime [ -a ][ -l ][ -t ][ -u ]
```

Il comando **rwho** fornisce informazione sugli utenti collegati su tutte le macchine connesse alla rete locale. Se un utente collegato non ha usato il terminale per un'ora o più, non viene considerato collegato al sistema, a meno che non venga specificata l'opzione "-a".

Il comando **ruptime** è l'analogo di "uptime(1)" e fornisce informazioni sul numero di utenti collegati, sull'uptime, sul carico di ogni macchina collegata alla rete. A seconda delle opzioni specificate l'elenco viene ordinato per carico "-l", uptime "-t", o per numero di utenti "-u". Il flag "-a" ha la stessa funzione che nel comando **rwho**.

5.2. Esecuzione remota

```
rlogin rhost [ -ec ][ -l username ]
rhost [ -ec ][ -l username ]
rsh host [ -l username ][ -n ] command
host [ -l username ][ -n ] command
```

Il comando **rlogin** consente di effettuare il login su una macchina remota (rhost) dalla macchina dove si è collegati (lhost). Grazie a questo comando si può lavorare come se si fosse collegati direttamente sull'rhost. Il comando "." disconnette dal remote host, dove "." è definito "escape character". Con l'opzione "-ec" è possibile fissare il carattere "c" quale escape.

Il comando **rsh** permette a un utente di eseguire un comando su una rhost senza effettuare il login sul sistema remoto. Se il comando non viene specificato, viene effettuato il remote login e il sistema richiede la password.

I comandi **rhost** ed **host** permettono l'esecuzione immediata dei comandi **rlogin** o **rsh** su un dato host. Ad esempio, con i comandi:

```
dist23 date
ugdistr date
```

si possono conoscere le "date" correnti sui nodi "dist23" e "ugdistr". Tuttavia per poter utilizzare questi comandi abbreviati occorre che nella directory /usr/hosts sia presente un file con il nome del host specificato. Tale file sarà un link simbolico con "/usr/ucb/rsh" e il comando **rsh** provvede a risolvere la chiamata. In particolare, se nella riga di chiamata compare un comando remoto, questo viene eseguito direttamente da **rsh** sullo host specificato, se non c'è viene istanziato un login remoto.

E' possibile specificare nel file "/etc/hosts.equiv" una lista di "rhosts". Se un utente possiede lo stesso nome di login sul sistema locale e su uno degli host specificati nel file suddetto, può eseguire il comando **rlogin** senza che gli venga richiesta la password. Ogni utente può inoltre possedere nella home directory un file ".rhosts" nel quale può specificare una lista di equivalenze del tipo

```
rhost username@rhost
```

che gli permette di effettuare remote login sull'rhost senza che gli venga richiesta la password. Per motivi di sicurezza questo file deve essere posseduto sia sul sistema locale che sul remoto.

Si noti che il comando **rsh** non richiede password: perciò se il proprio nome di login sul rhost non è lo stesso di quello sul local host, è indispensabile il file ".rhosts" per poter eseguire un comando con **rsh**. E' evidente che, per poter usare **rsh** e **rlogin** è necessario possedere una login sul sistema remoto.

5.3. Trasferimento di file

```
rcp file1 file2
rcp [ -r ] file ... directory
```

DistNet Progress Report

Il comando `rcp` copia file da una macchina all'altra. Ciascun argomento è della forma

```
rhost:path    (se è un file o directory remoto)
path         (se è un file o directory locale)
```

Se l'opzione `"-r"` è specificata e se uno dei file sorgenti è una directory, tutto il "subtree" di quella directory viene copiato nella directory destinataria (la destination deve essere una directory). Con questo comando è anche possibile trasferire files tra due sistemi remoti.

5.4. Altri comandi

```
ftp [ -v ][ -d ][ -i ][ -n ][ host ]
```

È un programma interattivo che consente di trasferire files tra sistemi remoti collegati alla rete. Costituisce un'interfaccia utente per il File Transfer Protocol standard dell'ARPANET.

```
telnet [ host [ port ] ]
```

È programma interattivo che consente di collegarsi ad altro sistema remoto usando il protocollo TELNET. Entrambi questi comandi sono utili quando si voglia colloquiare con sistemi non-UNIX. Per una più esauriente descrizione, si rimanda al manuale.

5.5. Sperimentazione

Finché non saranno pronte le interfaccia per l'hardware di rete, non sarà possibile utilizzare questi comandi per comunicare tra sistemi diversi. Tuttavia è già possibile provare ad usare questi comandi (per ora solo sul VAX) usando come rhost "localhost": questo consente la sperimentazione della sintassi dei comandi e delle prestazioni ottenibili in condizioni prossime a quelle di rete poco carica. Ovviamente "localhost" è il nome del VAX stesso, che quindi esegue sia le funzioni di calcolatore locale che di host remoto!

6. Prospettive future

Quali potrebbero essere i possibili sviluppi futuri di Ethernet all'interno del DIST, a medio e lungo termine (in quest'ultimo caso, non solo di Ethernet)? Gli aspetti da considerare sono numerosi, fortemente legati all'evoluzione tecnologica e alla diminuzione dei costi (e.g., componenti VLSI) e allo sviluppo delle reti a livello cittadino e, in particolare, nell'ambito universitario.

6.1. Terminalizzazione

La rete attuale si configura essenzialmente come rete di calcolatori. L'accesso in rete da un terminale avviene sempre e comunque attraverso la porta della macchina a cui questo è collegato. È necessario un protocollo di "terminale virtuale" per poter lavorare indifferentemente da ciascuna terminale su ciascuna macchina (per macchine eterogenee), o, in ambiente omogeneo Unix, un comando di "remote login".

Un accesso indipendente dei terminali (o personal computer) in rete si può realizzare tramite dispositivi denominati "communication server" o "terminal server", essenzialmente in tre modi:

- il server concentra un certo numero (4-8) di linee seriali RS-232 a cui accedono i terminali (quindi, un solo controller e transceiver per più terminali), e realizza la connessione con gli host, a loro volta connessi in rete tramite bus-controllers;
- stessa cosa, ma la connessione con gli host avviene tramite porte seriali collegate in rete con server analogo, e si realizzano circuiti virtuali indipendenti dal sistema operativo locale;
- ancora un terminal server per i terminali, ed uno lato host (e.g., Vax), costituito da una scheda che funziona da front-end, interfacciata col sistema operativo, eliminando le porte seriali.

In ogni caso, il vantaggio è costituito dall'eliminazione di wiring eccessivo, dalla possibilità di ricerca di una porta libera e dalla conseguente allocazione dinamica delle porte nel caso b), e dalla eliminazione di queste nei casi a) e c). Il caso a) è quello più alla nostra portata e più rispondente alla nostra filosofia (rete di calcolatori con s.o. Unix).

Il costo di tali dispositivi è per ora abbastanza elevato (e.g., Interlan 8 porte 4480\$, 4 porte 3500\$); tuttavia, la probabile diminuzione di costi da un lato e i vantaggi offerti rendono opportuno cominciare già da ora a pensare alla loro acquisizione nel medio termine.

6.2. Aperture verso l'esterno

Attualmente, i soli accessi esterni possibili ai nostri mezzi di calcolo sono costituiti dai due modem su linea commutata (300 baud lab NE2N e 1200 baud stanza Vax) collegati al Vax. È molto importante, come ormai si sottolinea da tempo, che il modem a 1200 sia dotato di "auto-call", in quanto questo consentirebbe l'ingresso come nodo attivo nella rete europea di sistemi Unix (EUNET) e la possibilità di uscire, tramite questa, sull'ARPANET. Tuttavia, la soluzione del problema dipende dalla SIP e non da noi.

Nel prossimo futuro (quando Italtel e Telettra avranno finito i concentratori) dovrebbero essere disponibile a Genova un accesso a ITAPAC (rete pubblica a commutazione di pacchetto). Sarà importante prevedere un link con questa, che potrebbe essere semplicemente via modem al concentratore con un normale protocollo asincrono, o anche direttamente in X.25 tramite un gateway.

Infine, è da considerare l'accesso in rete dei terminali (previsti) in Radio_net. La configurazione iniziale prevede alcuni terminali collegati via radio al Vax (con protocollo di accesso a canale comune di tipo CSMA per minimizzare

DistNet Progress Report

l'occupazione di banda) a 1200 baud, con ricerca automatica di una porta libera. In futuro, potrebbe essere interessante la realizzazione di un "terminal server" via radio per il collegamento diretto dei terminali remoti in rete.

6.3. Il BIL

Il BIL (Binario Informatico Ligure, per i pochi che non lo sanno) è per il momento un progetto "futuribile", ma non ci sarebbe da stupirsi se diventasse realtà in tempi brevi. Per quanto riguarda la rete si può delineare il seguente scenario.

Un nodo di DistNet, come minimo, potrebbe essere collegato al BIL. Se questo fosse costituito da una rete "broadband" tipo Sytek, non dovrebbe essere difficile interfacciarsi a livello di TCP/IP, e quindi rendere visibile il BIL da tutti i nodi di DistNet. In una tale situazione bisognerebbe preoccuparsi più del controllo degli accessi e della segretezza delle trasmissioni che della fattibilità tecnica di qualsiasi collegamento tra nodi qualsiasi.

7. Alcuni problemi

7.1. Macchine non-Unix

Almeno una macchina connessa in rete non avrà Unix come sistema operativo. I protocolli di comunicazione TCP/IP non sono Unix-dipendenti: sono stati sviluppati in ambiente Arpanet. La descrizione della struttura dei pacchetti e del protocollo di colloquio è contenuta in alcuni RFC ("Request For Comment") DARPA di relativamente facile reperibilità, o in alternativa, può essere dedotta dai moduli sorgente dell'implementazione Unix.

In ambito TCP/IP esistono i servizi di file transfer (ftp) e di login remoto (telnet), meno flessibili dei corrispondenti servizi realizzati per ambiente Unix (r_{cp} e r_{login-rsh}), ma facilmente trasportabili sotto altro sistema operativo. E' bene comunque mettere in evidenza le ovvie limitazioni di RT11, oltre al legittimo dubbio se convenga sviluppare software nuovo per sistemi obsoleti.

Pare anche che esista la possibilità di acquistare software di supporto per TCP/IP sotto RT11 (o RSX11), possibilità che richiede approfondimento sia per quanto riguarda i costi, sia per quanto riguarda le funzionalità.

E' bene mettere in evidenza che, allo stato attuale, l'hardware Ethernet installato realizza unicamente i livelli 1 e 2 del modello ISO-OSI, e quindi una macchina RT11 potrebbe al più, per mezzo di un semplice driver software, eseguire operazioni di R/W con un'altra macchina RT11 e nulla più.

7.2. Problema DIBE

Come è stato prima evidenziato, le strutture hardware di rete locale comprendono anche alcu-

ne macchine (due: un Vax 11/750 e un microPDP) dep DIBE, che ha scelto da tempo la strada di seguire a ruota i nostri orientamenti in fatto di strutture informatiche. E' importante in questa fase definire con chiarezza il tipo di rapporti che intendiamo avere, sia per quanto riguarda l'uso delle risorse comuni, sia soprattutto per la (eventuale) assistenza nell'installazione e per il trasferimento sulle loro macchine di software sviluppato o messo a punto dal DIST nell'ambito di DistNet.