INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONE: 2240-1
CABLE: CENTRATOM - TELEX 460392 - I

SECOND SCHOOL ON ADVANCED TECHNIQUES
IN COMPUTATIONAL PHYSICS
(18 January - 12 February 1988)

SMR.282/9

DECNET

M.NATIELLO

Quantum Chemistry Group, University of Uppsala, Sweden

DECNET

- * Communication & computer networks.
- * Nodes & Links.
- * ISO standards of communication.
- * Layers. Interfaces (↕) — Protocols (↔).
- * Transparency: User protocol is independent of the lower layers.

Layer	**Person 1**	**Person 2**
User	Transmit a feeling (Happiness)	Receive the message
Presentation	Put it in words (language)	Understand words
	(I am happy)	
Session	Make the logical link	(same)
	(It's me who is saying/writing it)	
Transport	Voice/written paper	(Hear/read)
Network	Check destination	Check origin
& Communication	& Transmission	& Reception
Physical link	Sound waves,telephone, a letter	(same)

DECNET layer system

- ⋆ User
- ⋆ Network management & Network application
- ⋆ Session control
- ⋆ End communication
- ⋆ Routing
- ⋆ Data link
- ⋆ Physical link

User layer

- * Issue commands to be executed across the network (copy, login,...).
- * Run programs requiring use of the network (read files in another machine).

Network layer

- * Configurates, monitors and controls the network.
- * Database with circuits, routes, privileges, line characteristics, ...
- * Network Control Program.

Session layer

- * Maps node name to address.
- * Validates incoming connect request.

Routing & End communication layers

- * Logical link management
- * Message segmentation
- * Message routing
- * Congestion control
- * Guaranteed delivery

Data link

- * Sends data through the Physical Link according to line characteristics.

Physical Link

- ○ Point-to-point connection.
  - **Static** (or dynamic: upon request).
  - Connection between two nodes using terminal ports.
  - Routes(pathway).
- ○ Multipoint connection: Several nodes "hang" to a wire.

Ethernet

- Devices: Ethernet wire, H4000, Transciever cable, Ethernet controller card.
- Top transmission speed: 10Mbits/sec. (Cf. terminal lines: 19200 bits/sec.)
- All nodes are logically adjacent.
- Carrier Sense, Multiple Access with Collission Detect.
- Everybody can talk to each other.
- Messages addressed to other people are ignored.
- If two participants start talking together, they wait a random time and start again (hoping that everybody is listening).
- Example: Mat Nat Fak.

* Invoke a network request:

```
NODE::
```

  ⋆ The network user is identified as DECNET in the target machine (few privileges).

```
NODE"USER PASSWORD"::
```

  ⋆ The network user is identified as the local USER in the target machine (the password is sent through the network).

* `dir VERONA"MARIO venezia"::dua0:[mario]`

* Proxy database: Associate a remote user with a local user.

  Network request is checked with the database.

  Remote user inherits the rights of the associated local user.

* `dir VERONA::dua0:[mario]`

Network operation at the DCL level

- Device handling: **Assign, Define**
- File manipulation: **Copy, Create, Delete,**

  Set, Type, Edit,...
- Remote resource sharing: **Print/remote**

  Submit/remote
- Terminal to terminal interactive communication:

  Phone node::user
- Mail: **mail node::user**
- Remote program execution: **run**
- Remote login: **set host NODE**
- Use of network objects:

  Use NCP to define a program as a network object at the target node.

Intertask communication with command procedures

* Write a command file `RING.COM` in node A.

- `RING.COM`

  Open a logical link (`Task=answer`).

  Write on the link.

  Read from the link.

  Output to screen.

  Close the link.

* Write a command file `ANSWER.COM` in node B

(Similar to `RING.COM` but read before write).

`sys$net` is the name of the logical link.

* `@RING.COM`

```
$!Ring.com --- proxy.  Remote node is max.
$open/read/write link max::"task=answer"
$write link "hola"
$read link text
$write sys$output text
$close link
$exit
```

```
$!Answer.com
$open/read/write link sys$net
$read link text       !First read.
$write link "que tal" !Now tell.
$write sys$output text
$close link
$exit
```

At remote node:

```
------------------------------------------
Connect request received at Date & Time
from remote process KVAX3::"0=MARIO"
for object "ANSWER"
------------------------------------------
hola
```

At local node:

```
que tal
```

Task to task communication in FORTRAN

* Node A:

```
$!proc1.com
$assign max::"""task=proc2""" remote
$run proc1
$exit
```

```
C     proc1.for
      open(1,file='remote',status='new')
      do j=1,5
      write(1,*)J
      type *, 'We send number ',j
      read(1,*)i
      type *, 'We get number ',i
      enddo
      close(1)
      end
```

* Node B:

```
$!proc2.com
$run proc2
$exit
```

```
C     proc2.for
      open(1,file='sys$net',status='old')
      do j=1,5
      read(1,*)i
      type *, 'We get number ',i
      ij=i*j
      write(1,*) ij
      type *, 'We send number ',ij
      enddo
      close(1)
      end
```

- Output for node A:

```
We send number 1
We get number 1
We send number 2
We get number 4
We send number 3
We get number 9
We send number 4
We get number 16
We send number 5
We get number 25
```

- Output for node B:

```
-----------------------------------------
Connect request received at Date & Time
from remote process KVAX3::"0=MARIO"
for object "PROC2"
-----------------------------------------
We get number 1
We send number 1
We get number 2
We send number 4
We get number 3
We send number 9
We get number 4
We send number 16
We get number 5
We send number 25
```