Networking is IPC

Step 1: Getting two systems to Communicate
Assembling Some Pieces

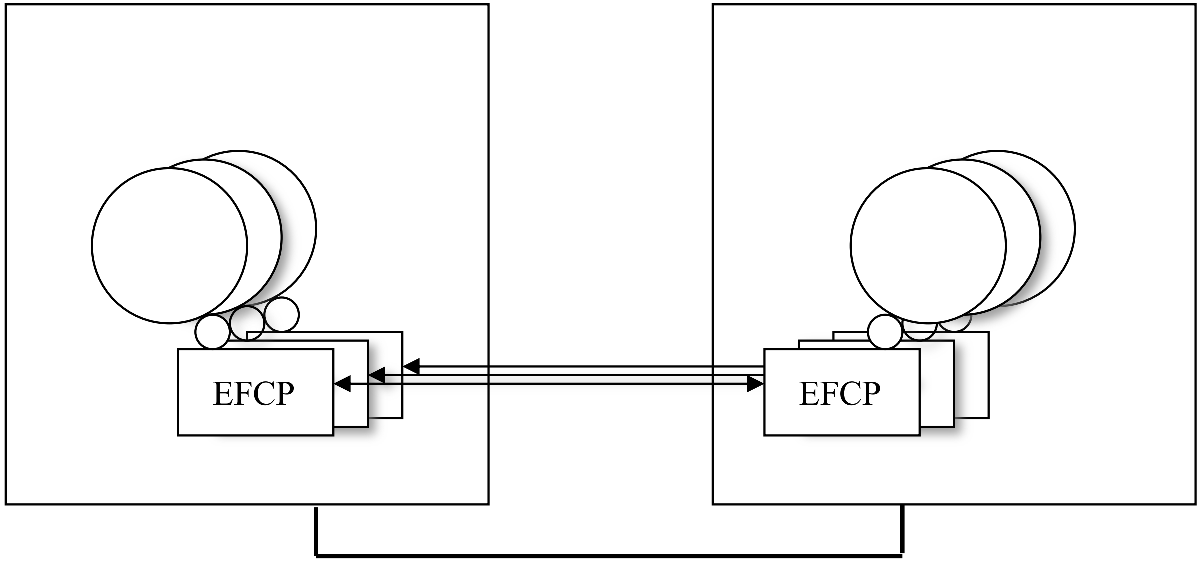
John Day

"Architecture is doing the algebra,
before doing the arithmetic."

3: Simultaneous Communication Between Two Systems

i.e. multiple applications at the same time

- To support this we have multiple instances of the EFCP.



Will have to add the ability in EFCP to distinguish one flow from another.

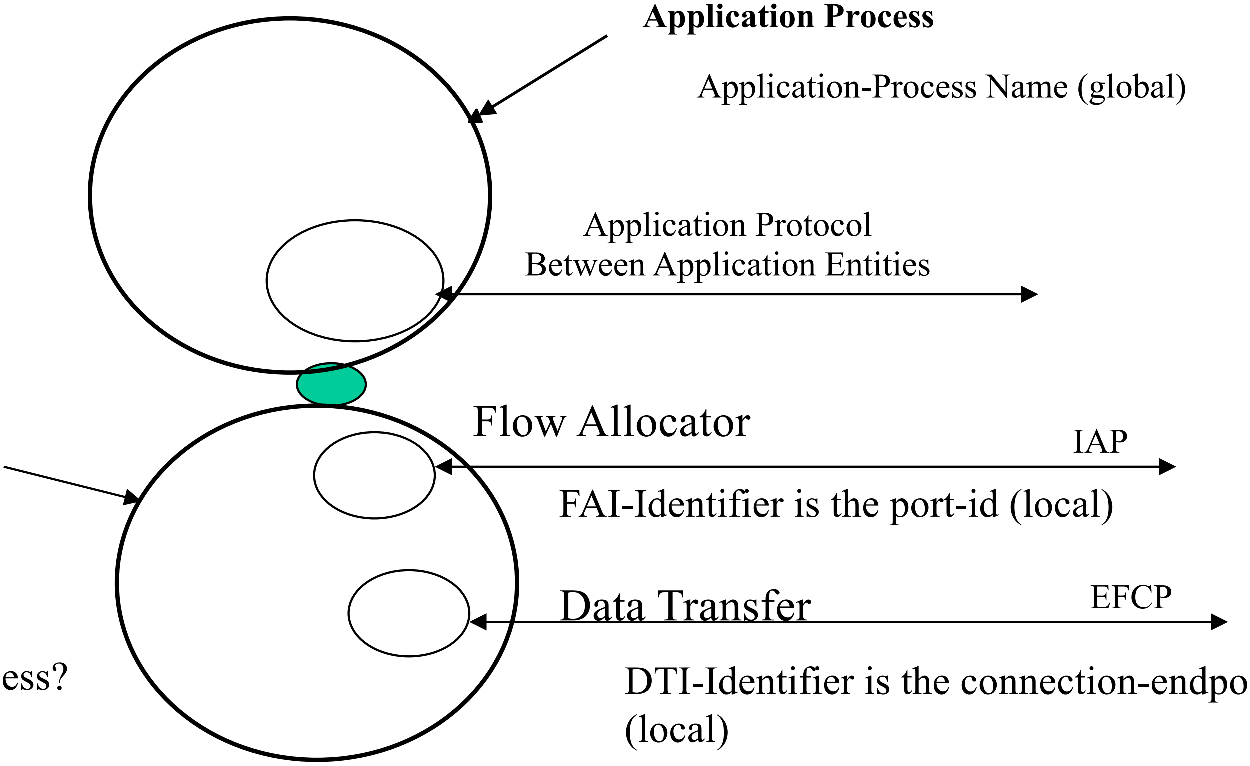
Typically use the port-ids of the source and destination.

Connection-id

Dest-port	Src-port	Op	Seq #	Cksm	Data

Also include the port-ids in the information sent in IAP to be used in EFCP synchronization (establishment).

What Does an IPC Process Look Like?



Application Process

Application-Process Name (global)

Application Protocol
Between Application Entities

Flow Allocator

IAP

FAI-Identifier is the port-id (local)

Data Transfer

EFCP

DTI-Identifier is the connection-endpoint-id (local)

IPC Process

AP-Name (global)

Main task of an IPC Process?

Managing IPC

Flow Allocator

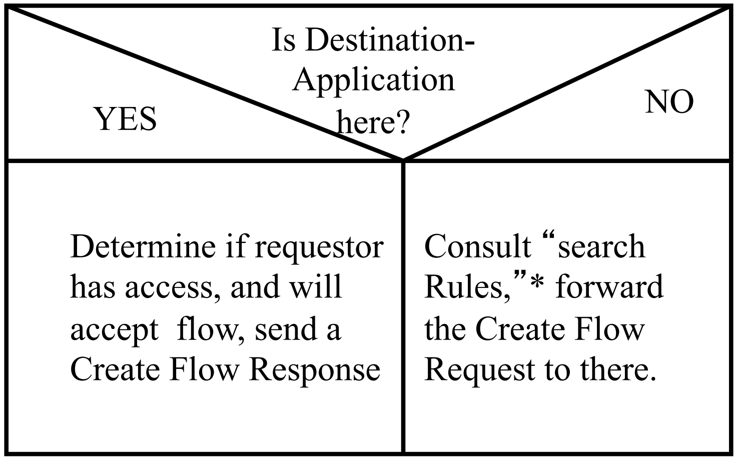
(formerly IAP)

- Create_Request (flow object, destination-application-name, source-application-name, source-CEP-id, QoS parameters, policy list, access control)
  - Send to the other guy and instantiate Data Transfer
- Create_Response (flow object, destination-application-name, source-application-name, source-CEP-id, destination-CEP-id QoS parameters, policy list, access control, status)
  - Returned by the other guy.

  - This is a degenerate case. There is a subtlety here.
  - The “search rules” do not indicate where the destination is, but where is the next place to look for the destination.
    - In this case, the first place we look, says it’s me!

Flow Allocator
(What Does It Do?)

- The Flow Allocator has two inputs:
  - Allocate requests from the user of the DIF
  - Create flow requests from from another IPC Process.
- What does it do?
  - Translate the Allocate request into a create flow request.
  - After that it is the same:



Is Destination-Application here?

YES

NO

Determine if requestor has access, and will accept flow, send a Create Flow Response

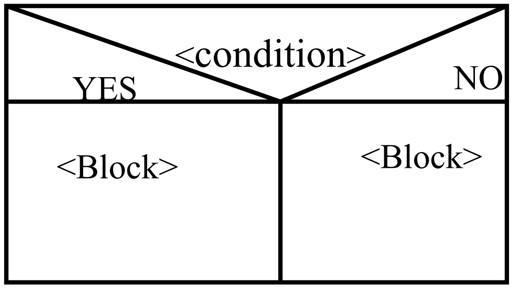
Consult “search Rules,”* forward the Create Flow Request to there.

One of Day’s Asides

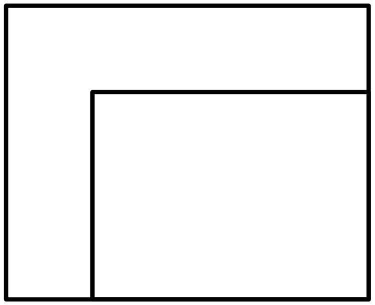
Nassi-Schneiderman Diagrams (1972)

- Ever do Flow Charts?
- I despise Flow Charts
  - It is Programming in Assembler!
- This is better.
- See the Wikipedia page

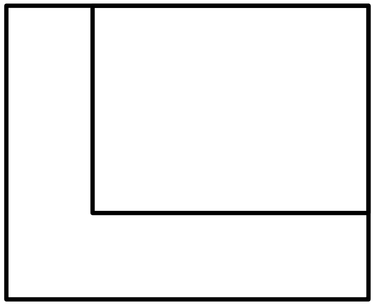
A block of prose,
pseudo-code, code



If . . . . Then . . . Else



Until <condition> Do . . .

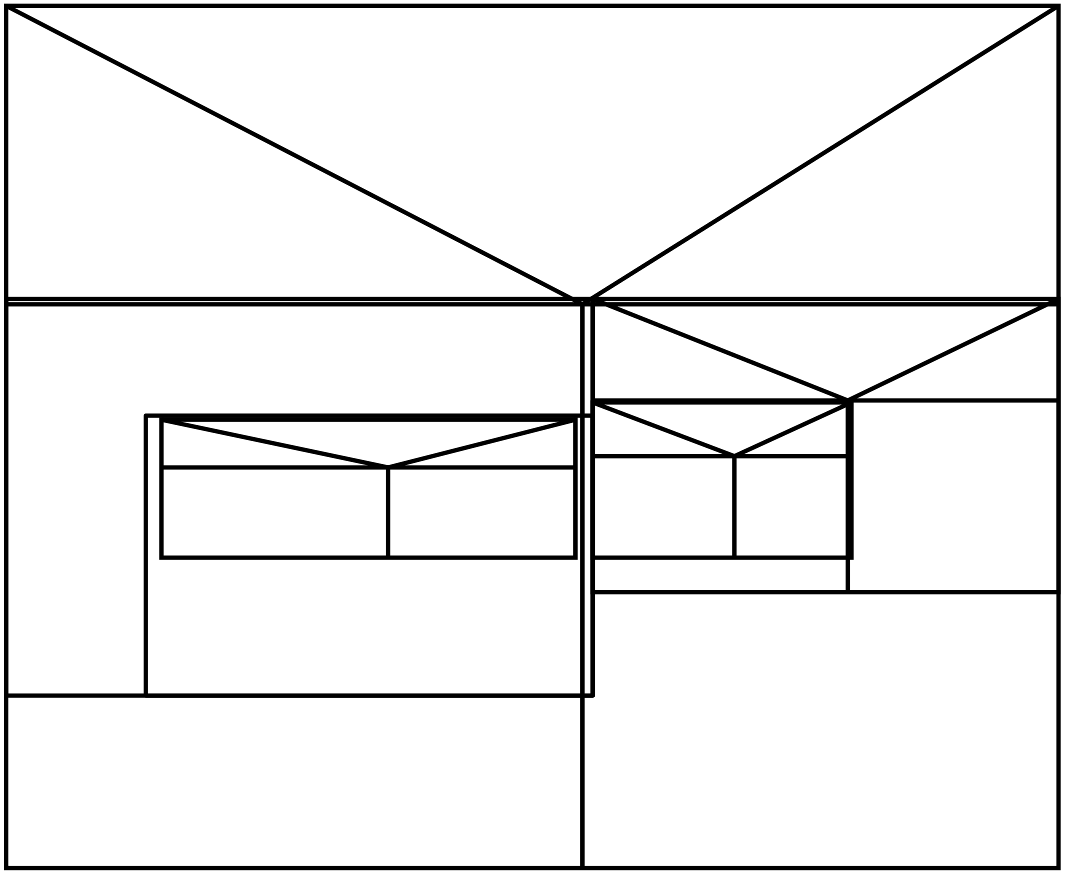


Do . . . While <condition>

Day’s Aside Continued

Nassi-Schneiderman Diagrams

- They nest.

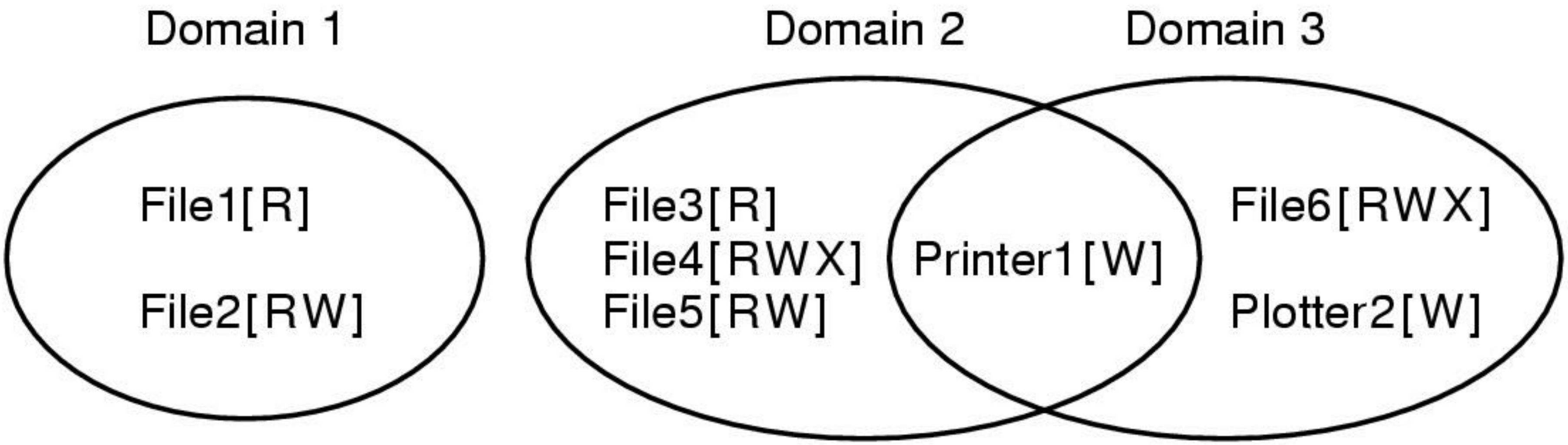


Back to Work

- According to the IPC Model,
- Once the Flow Allocator finds the Requested Application, it must determine whether or not the Requestor has access to it.

A Refresher on
Protection Mechanisms

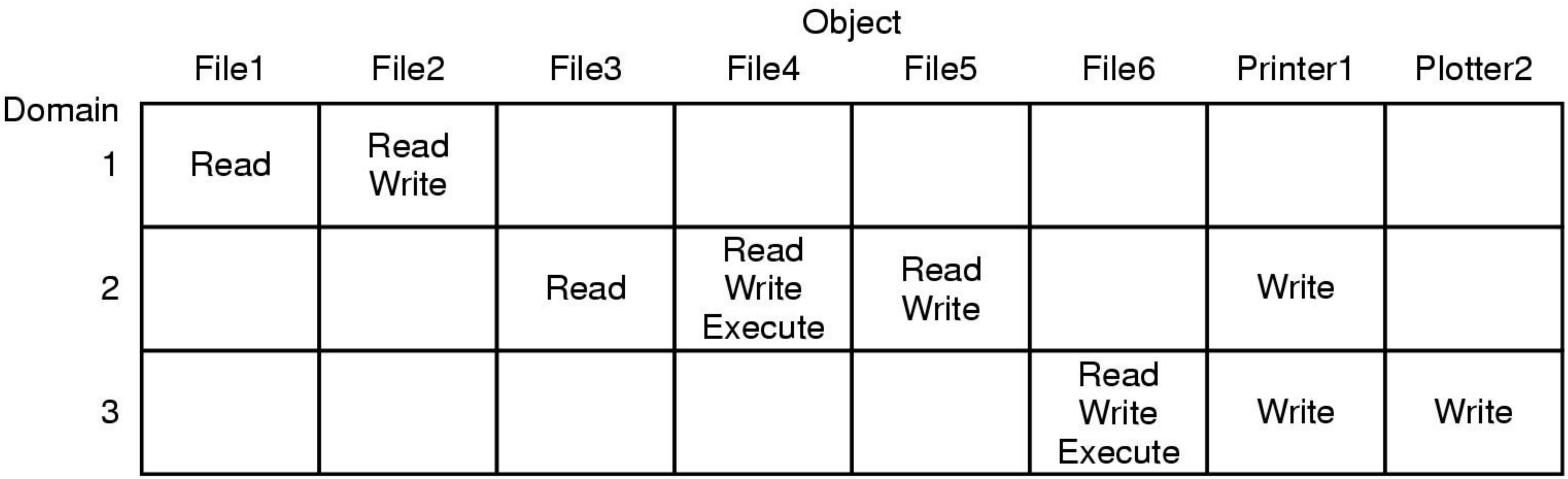
Protection Domains (1)



© A. Tanenbaum, Modern Operating Systems

Examples of three protection domains
(objects, rights)

Protection Domains (2)



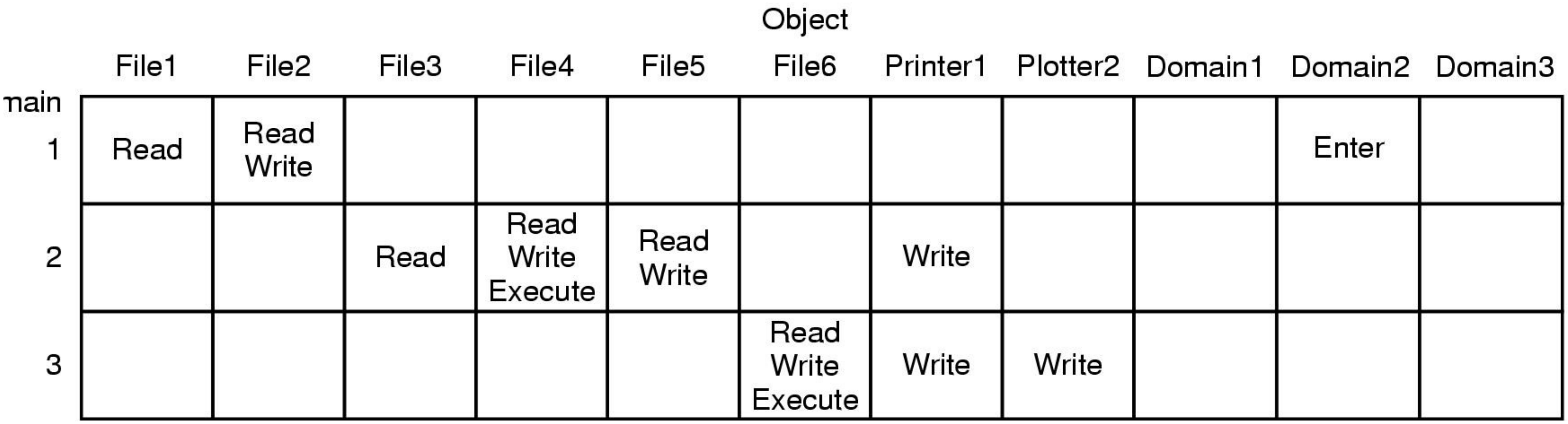
Object

Domain	File1	File2	File3	File4	File5	File6	Printer1	Plotter2
1	Read	Read Write						
2			Read	Read Write Execute	Read Write		Write	
3						Read Write Execute	Write	Write

© A. Tanenbaum, Modern Operating Systems

A protection matrix

Protection Domains (3)

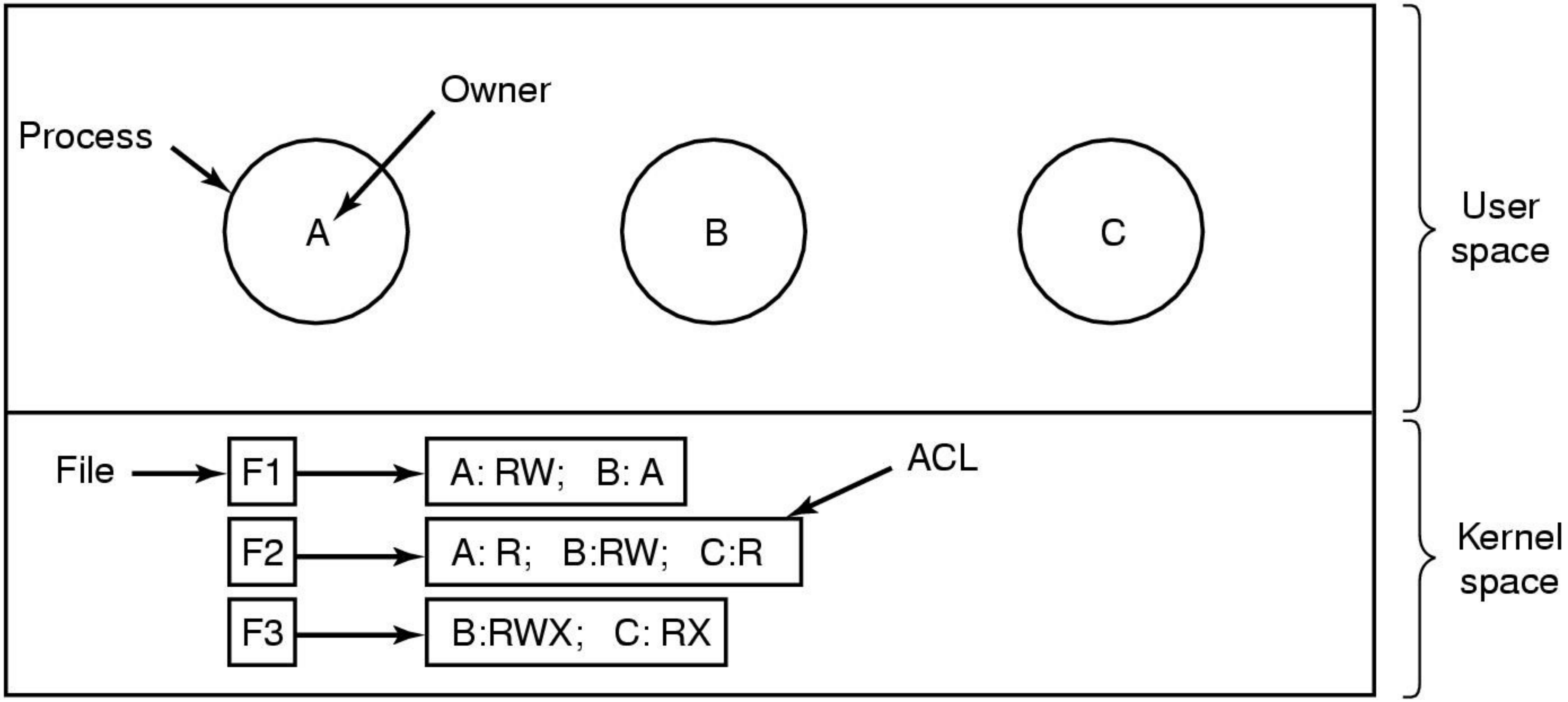


						Object					
nain	File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
1	Read	Read Write								Enter	
2			Read	Read Write Execute	Read Write		Write				
3						Read Write Execute	Write	Write			

© A. Tanenbaum, Modern Operating Systems

A protection matrix with domains as objects

Access Control Lists (1)

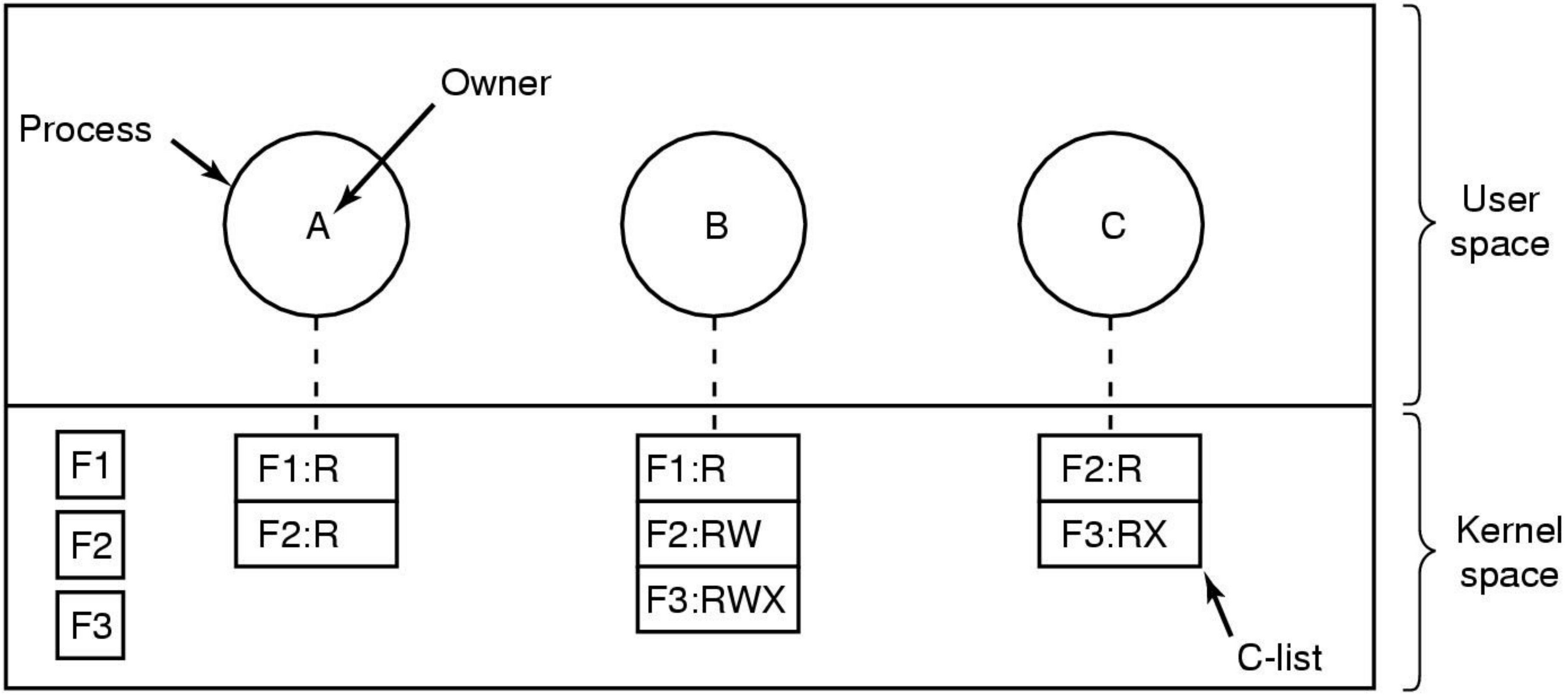


© A. Tanenbaum, Modern Operating Systems

Use of access control lists of manage file access

Protection Matrix by columns

Capabilities (1)



© A. Tanenbaum, Modern Operating Systems

Each process has a capability list

Protection Matrix by row.

Capabilities (2)

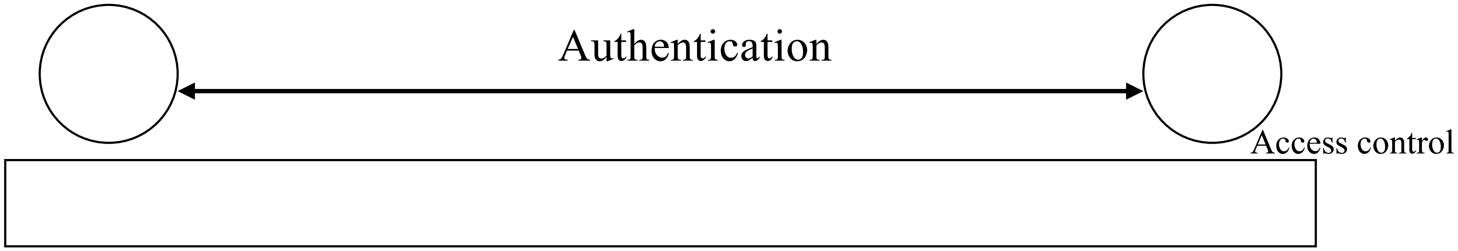
- Cryptographically-protected capability

Server	Object	Rights	f(Objects, Rights, Check)

- Generic Rights
  1. Copy capability
  2. Copy object
  3. Remove capability
  4. Destroy object
- The destination generates the capability and distributes it to the source.
  - We will have to generalize this for domains rather than individual applications in order for it to scale.

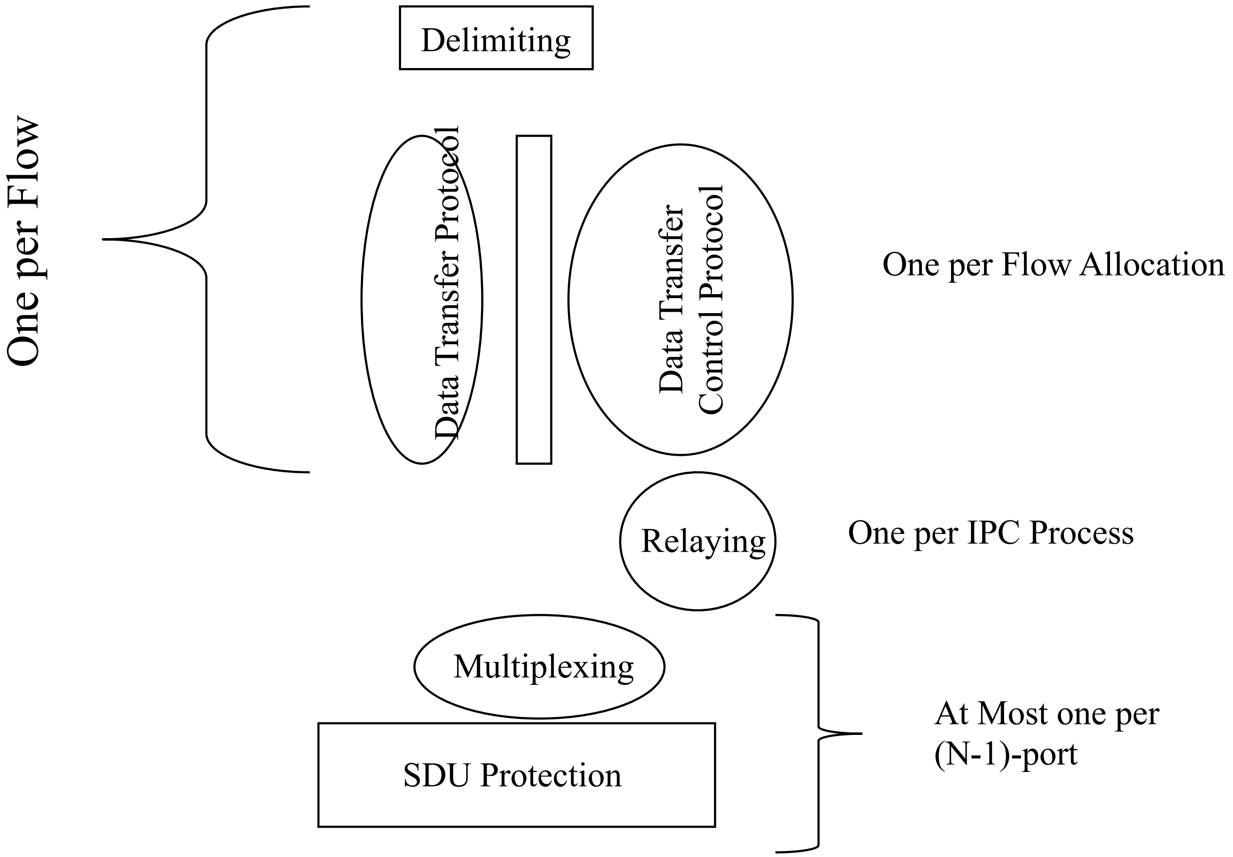
© A. Tanenbaum, Modern Operating Systems

Just Access Control?



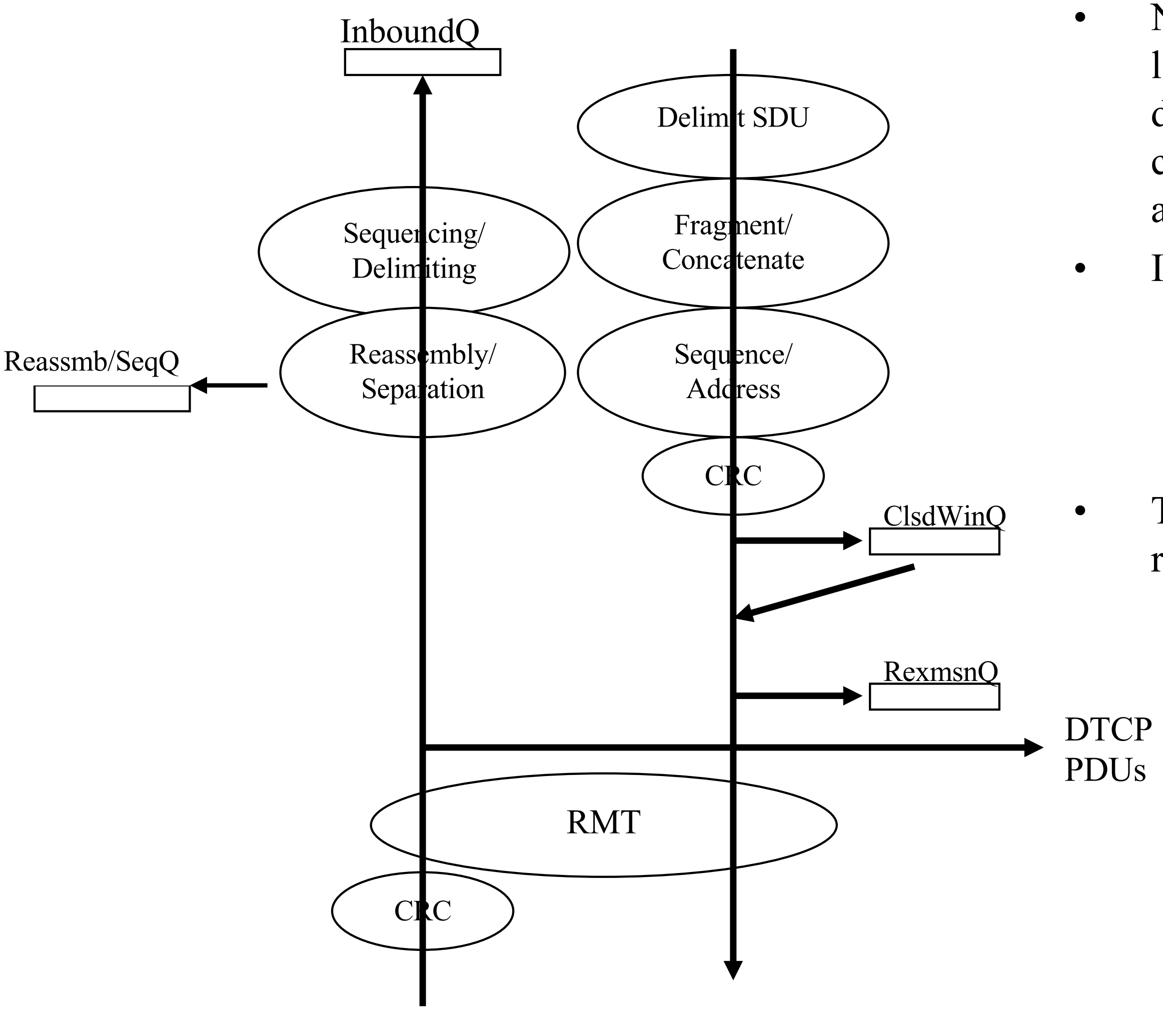
- Yes, all IPC can do is determine whether the requesting application has access to an application.
- IPC can not be sure that the application requested is really that application.
- Only the source application can determine that.
- IPC Access Control.
- The Applications must do Authentication.
  - Members of the Layer may authenticate each other.

Data Transfer



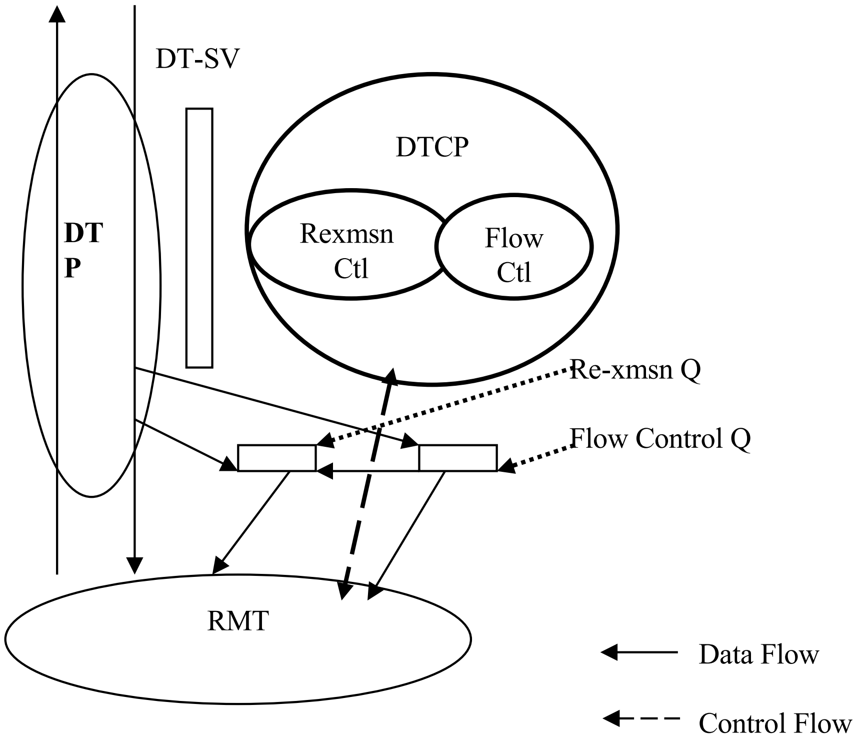
- More flows coming in the top than going out the bottom.
- Tightly coupled mechanisms in the Data Transfer Protocol.
- Loosely coupled mechanisms in the Data Transfer Control Protocol.
- SDU Protection must be common to the layer.

Data Transfer Protocol



- Notice that the flow is a straight shot, very little processing and if there is anything to do, it is moved to the side. The most complex thing DTP does is reassembly and ordering.
- If there is a DTCP for this flow:
  - If the flow control window closes, PDUs are shunted to the flow controlQ.
  - If the flow does retransmission, a copy of the PDU is put on the rexmsnQ.
- These PDUs are now DTCP's responsibility to send when appropriate.

Finally Data Transfer Control Protocol



- For flows with retransmission (acks) and/or flow control, a DTCP is required.
- DTCP controls flow volume, the RMT controls flow rate of (N-1)-flows.
- We need to bound 3 timers, MPL, Ack delay, and retries.
- For the rest of the course we will be looking at how current protocols follow this model.

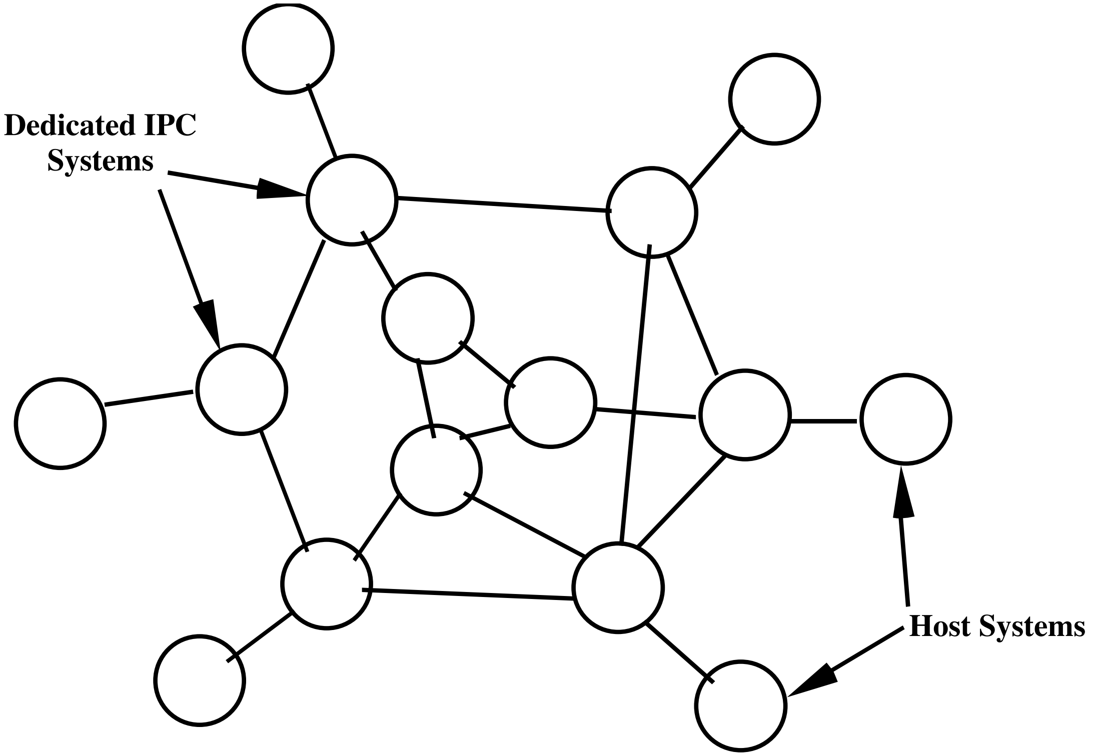
That’s The Basics of Data Communication Now for Networking

(But it turns out that
Data Comm is a degenerate case
of Networking)

Wireless Introduces New Considerations: II

- However, wireless comes with a cost, almost all of which derive from it being a shared medium:
  - It will have limited range.
  - Wireless is a notoriously hostile environment for corrupting PDUs.
  - There is contention for sending PDUs.
    - There are methods that get utilization beyond the basic 36%, but contention is still an issue and greatly reduces the capacity of the network.
- Wireless isn't going to scale well to large networks.
- Consequently, wireless tends to be used at the periphery of a wired network that relays.
  - But we would like to keep the property that makes all members of a layer appear directly connected (one hop away).
- So, what about N Systems Connected with Relays?

5: Communicating with N Systems (On the Cheap)



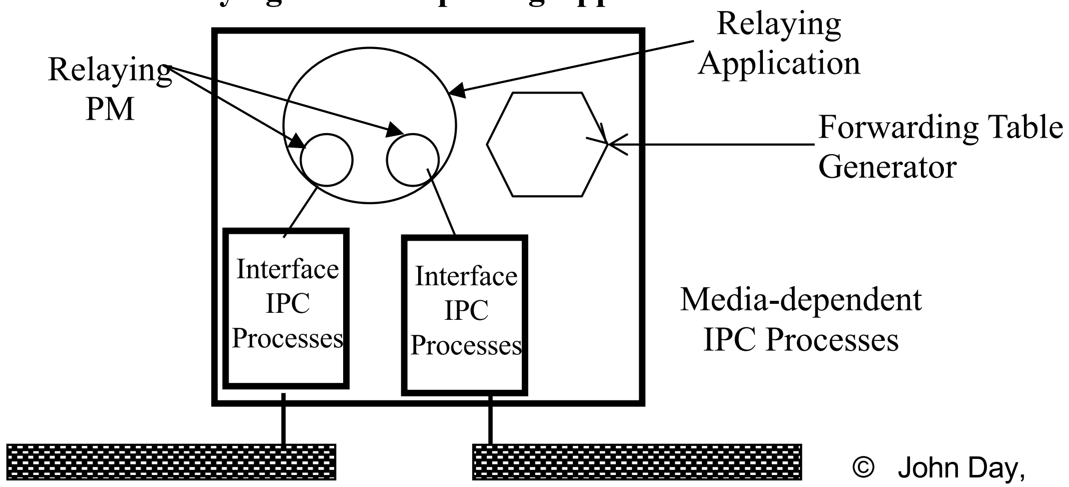
By dedicating systems to IPC, reduce the number of lines required and even out usage by recognizing that not everyone talks to everyone else the same amount.

Communications on the Cheap

- We will need systems dedicated to relaying and multiplexing.
- That requires some new elements:
  - Globally accepted names for source and destination IPC Processes.
  - And also for the relays. Relays require names for routing. Have to know where you are to determine where to go next, i.e. forwarding.
  - Need some way to figure out the forwarding table, traditionally this has been routing, which will need to exchange information on connectivity.
- Will need to add more information to the PDUs to carry the names for relaying and multiplexing.
  - Interface IPC Facilities will need one too if they are multiple access.

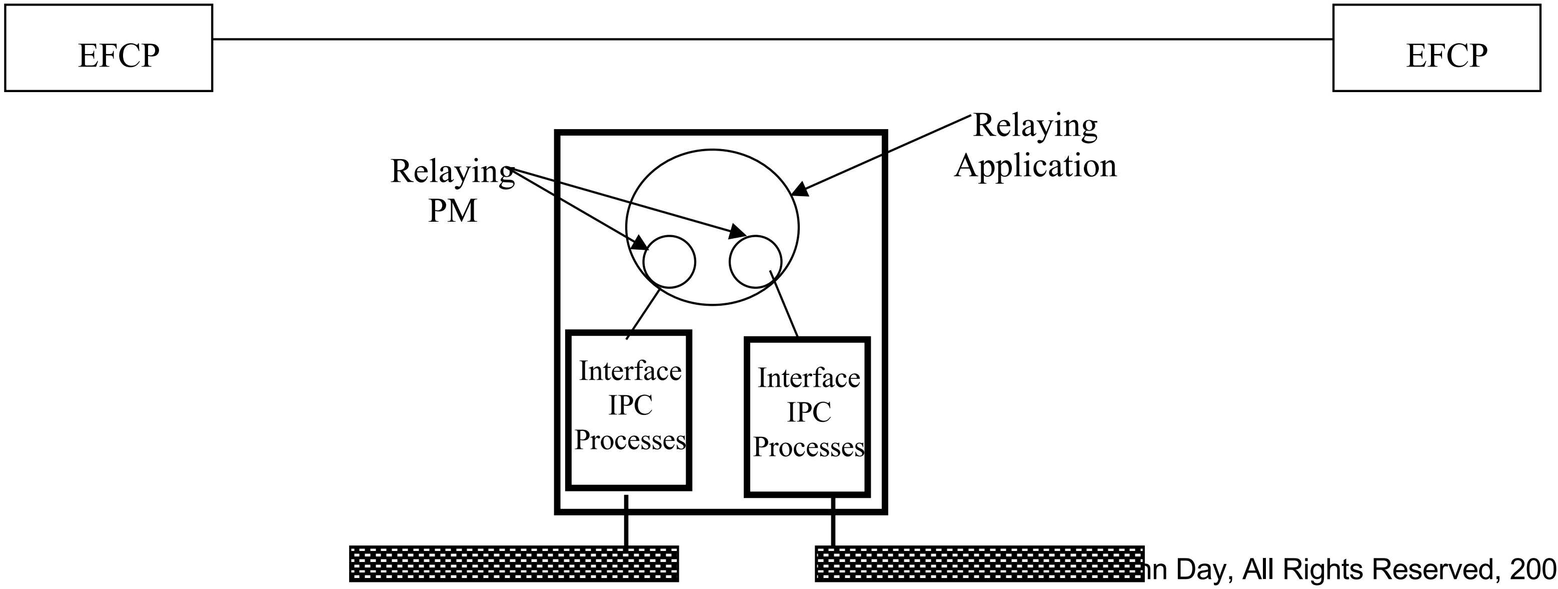
Dest Addr	Src Addr	Dest-port	Src-port	Op	Seq #	CRC	Data

Common Relaying and Multiplexing Application Header



Communications on the Cheap

- But relaying systems create problems too.
  - Can’t avoid transient congestion from time-to-time.
  - Annoying bit errors can occur in their memories.
- Will have to have an EFCP operating over the relays to ensure required QoS reliability parameters.
  - Our virtual IPC Facility isn’t very virtual.



Now Things Get Complicated
(The Next Few Lectures)

- Next Week
  - Forwarding (often referred to as routing)
  - Relaying (Network Layer)
- The Following Week
  - Addressing (the most important topic in networking; the least understood, and easier than we have made it)
- (Mid-Term)
- After the Mid-Term
  - Error and Flow Control over Relaying (Transport Layer)
  - Congestion Control in Networks.