# FIBRE CHANNEL SEMINAR

## FIBRE CHANNEL ASSOCIATION

# ACKNOWLEDGEMENT

This Seminar is based upon material originally created  by Dal Allan and Roger Cummings for Technology Forums Interface conferences in the USA and Europe from 1988 to 1993.

The material has been revised and updated by the FCA Technical Committee with significant contributions from Kumar Malavalli and Horst Truestedt.

# FIBRE CHANNEL SEMINAR
# COURSE OUTLINE

- **First Session**
  - Introduction
  - Architecture
  - FC-0 Physical
  - FC-1 Encode/Decode
- **Second Session**
  - FC-2 Framing Protocol
- **Third Session**
  - FC-3 Common Services
  - FC-4 Protocol Mappings
- **Fourth Session**
  - More on Topologies
  - The Present and the Future
  - Summary

# INTRODUCTION

- ANSI
- Where did Fibre Channel come from?
- What were Fibre Channel Requirements
- What you get when you meet requirements
- Fibre Channel vs IBM's ESCON Channel
- Documents

# ANSI

- CBEMA - Computer & Business Machines Manufacturing Association (Washington, DC)
- ANSI - American National Standards Institute (New York, NY); sets policy, represents USA in ISO meetings
- X3 - Computer Standards
- T9 - Technical Committee for Interfaces (replaced by T10, T11, T12)
- T10 (T9.2) - Technical Committee - Storage Device Interfaces Responsible for SCSI, ATA, SFF, and SSA
- T11 (T9.3) - Technical Committee - Storage Device Interfaces Responsible for IPI, HIPPI, and FC
- WG - Working Group of X3T11

# FIBRE CHANNEL TIMELINE

- Fibre Channel began as part of IPI Enhanced Physical
  - Scope widened to include HIPPI, SCSI and other interface protocols
- Chartered with wider scope by ANSI X3T9 in 1988
- First year spent looking for implementations to adopt. Reviewed:
  - ICL MacrorLan (2500 intallations world-wide in 7 years
  - HP-FL; DG Fibre-ICB; ANCOR FastNet; Canstar Hubnet
  - Picked IBM High Speed Interconnect in Sept89.
- First major decision was selecting coding scheme. Considered:
  - 4B/5B (FDDI); DEC 8B/10B with FEC; scrambling; HP 16B/20B
  - Picked IBM 8B/10B in Oct89
- Decided to add a low cost Copper fibre variant

# FIBRE CHANNEL TIMELINE (Cont'd)

- Work started on FC-PH in earnest in late 1989:
  - First merged FC-PH draft presented at March 1991 Working Group
  - FC-PH forwarded to X3T9 for peer review in August 1991:
- SC connector adopted in October 1991
- FC-PH forward to X3T9 for standardization again in February 1992:
- FC-PH forwarded for first Public Review in June 1992
- FC-PH forwarded for second Public Review in June 1993
- FC-PH on path to becoming an ANSI Standard in 1994

# FIBRE CHANNEL TIMELINE (Cont'd)

- Continuing FC-related projects:
  - Mappings of existing protocols:
    - SCSI (FCP and GPP)
    - IPI-3
    - SBCCS
    - LE
    - ATM
  - Topology-specific Fabric projects:
    - Fabric Generic Requirements
    - Point-to-Point - in base FC-PH standard
    - Cross Point Switch Fabric (with Distributed Fabric Annex)
    - Arbitrated Loop
  - Enhanced Physical (FC-EP)

# FIBRE CHANNEL REQUIREMENTS

## *Solve problems with existing channel/interfaces.*

- Small footprint connectors
  - Serial for ruggedness (fewer pins)
    - Scalable to 10 kilometer operating distance for one hop
    - Scalable to 100 megabytes/s in payload (each directrion)
  - Support multiple cost/performance levels - small systems to super computers
- Large connectivity (more than existing multidrop channels)
- Low-level protocol for efficiency over distance (not an extender)
- Efficient multiplexing of multiple "streams" into a single port
  - Break the relationship between physical interface and protocol
  - Carry multiple existing interface command sets (SCSI, IPI-3, HIPPI-FP etc.) through the same port. Preserve current driver software.

## *Develop a single Channel/Network - Fibre Channel*

# WHAT YOU GET IF YOU MEET REQUIREMENTS

**YOU CAN RUN EXISTING COMMAND SETS**
**ON TOP OF FC-PH**
**WITH MULTIPLE PHYSICAL VARIANTS**
**EACH WITH HIGH PERFORMANCE**
**USING FIBER FOR DISTANCE**
**AND COPPER FOR LOW COST!**

- Scalable Distance
- Scalable Performance
- Repeatable Performance
- Hardware-based protocol  for efficiency
- One hardware se  supports all upper-level protocols
- Data Transparency
- Low-level Error Recovery

# WHAT IS FC-PH

- A Channel/Network hybrid
  - Enough of a network to gain connectivity, distance, serial interfaces
  - Enough of a channel to retain simplicity, reliablility, hardware functionality
  - Multiple topologies
    - Siwtche
    - Point-to-point
    - Arbitrated Loop
- Major departure from "traditional" networks
  - Traditional networks have "passive" interconnections between nodes
  - Fibre Channel allows for active, intelligent topologies
  - Fabric topology must be self-managed

# FIBER CHANNEL'S RELATIONSHIP TO IBM'S ESCON CHANNEL

- No one thought there was a relationship
- Until September 1989 the ANSI committee had no idea that there was a relationship:
  - A number of key people in the Working Group were active in the ESCON development (we know now)
  - They brought their ESCON experience, did not try to force IBM's approach or implementation
  - Fiber Channel is much the better for their involvement
- Fibre Channel uses the same data coding scheme
  - IBM holds the patent on the 8B/10B coding scheme
- ESCON has Two sets of  transceivers:
  - Standard 200 megabaud LED (FDDI-like) transmitter for multimode to 3 Km
  - Extended Distance Facility (XDF) laser transmitter for single mode to 20 Km (speed to 18 MegaBytes/s)
- Connector is D-shell type with laser safety shutter
- ESCON Director operates similar to Class 1 "dedicated connection" mode

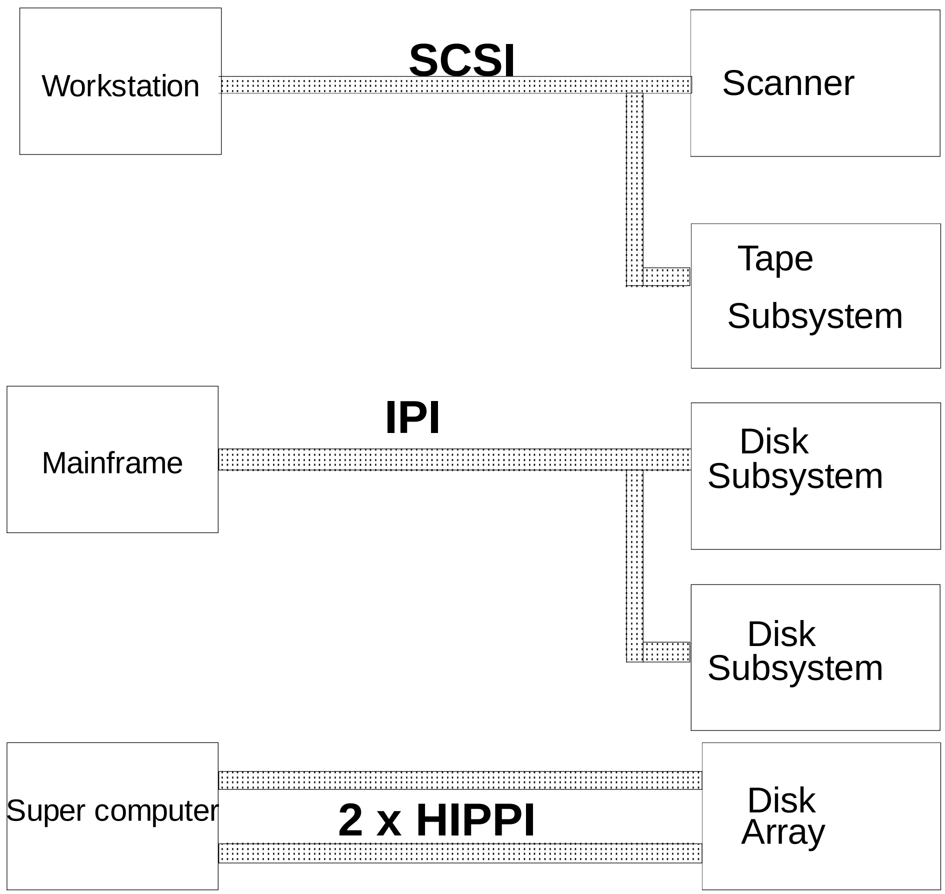
# FIBRE CHANNEL
# PROJECTS AND DOCUMENTS

| | |
|---|---|
| ▪ FC-AL | Arbitrated Loop Topology |
| ▪ FC-ATM | ATM over FC |
| ▪ FC-EP | Fibre Channel Enhanced Physical |
| ▪ FC-FG | Fabric Generic Requirements |
| ▪ FC-FP | Mapping to HIPPI Framing Protocol |
| ▪ FC-IG | Implementation Guide |
| ▪ FC-LE | Link Encapsulation |
| ▪ FC-PH | Fibre Channel Physical and Signaling Interface |
| ▪ FC-SB | Mapping to Single-Byte Command Code Sets |
| ▪ FC-GS | Generic Services (proposed project) |
| ▪ FC-XS | Cross Point Switch Fabric Topology |
| ▪ HIPPI-FC | FC over HIPPI |
| ▪ IPI-3 Disk | Revision to IPI-3 Disk |
| ▪ IPI-3 Tape | Revision to IPI-3 Tape |
| ▪ SCSI-GPP | SCSI-3 Generic Packetized Protocol for FC |
| ▪ SCSI-FCP | SCSI-3 Phase Emulation Protocol |

# ARCHITECTURE

- Existing Channels
- Architecture
- Topologies
- Architectural Levels
  - FC-0 Physical
  - FC-1 Encode/Decode
  - FC-2 Framing Protocol
  - FC-3 Common Services
  - FC-4 Protocol Mappings
  - ULP Upper Layer Protocol

# EXISTING CHANNEL ARCHITECTURE



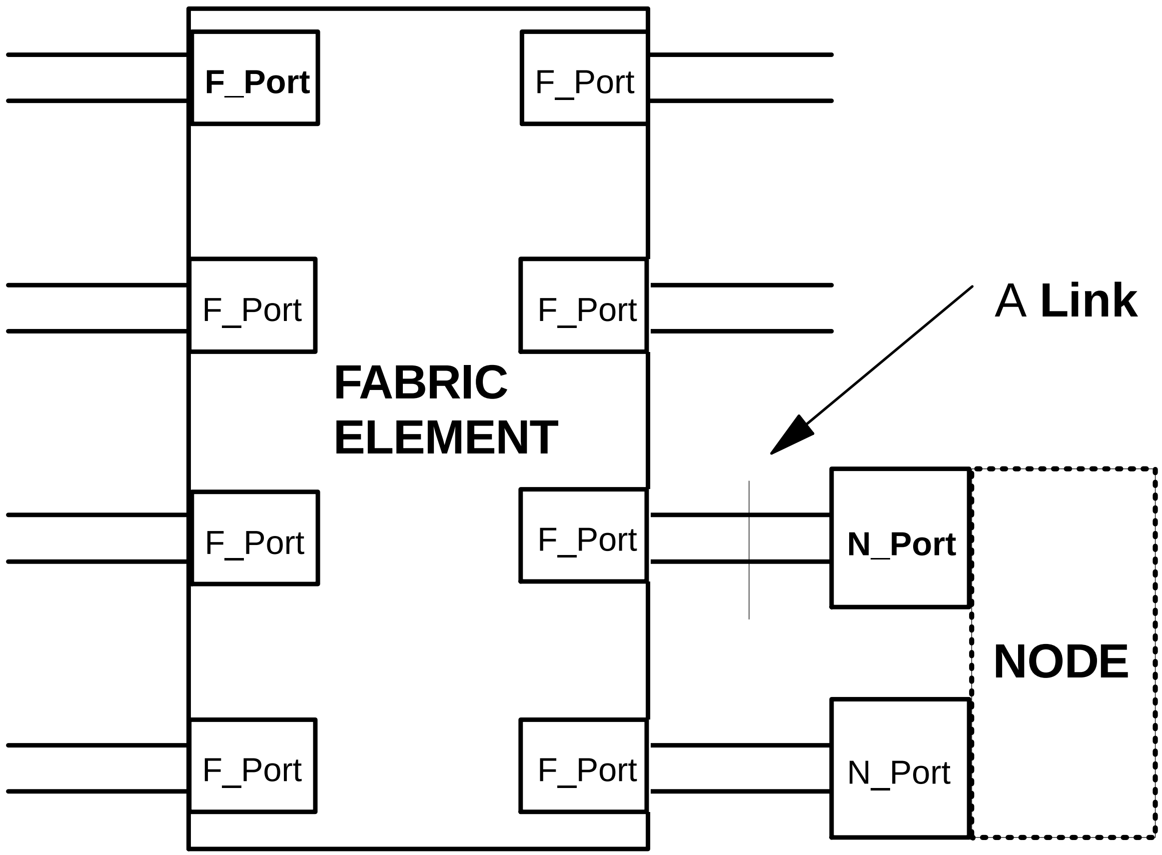
# PROBLEMS WITH EXISTING CHANNEL ARCHITECTURES

- Footprint Too Large for small systems and peripherals
- Distances Too Short for true distributed systems
- Connectivity Too Limited - using multidrop
- Speed Too Low - high speed required
- All channels need the same extension
- A fixed topology that requires manual intervention for addressing

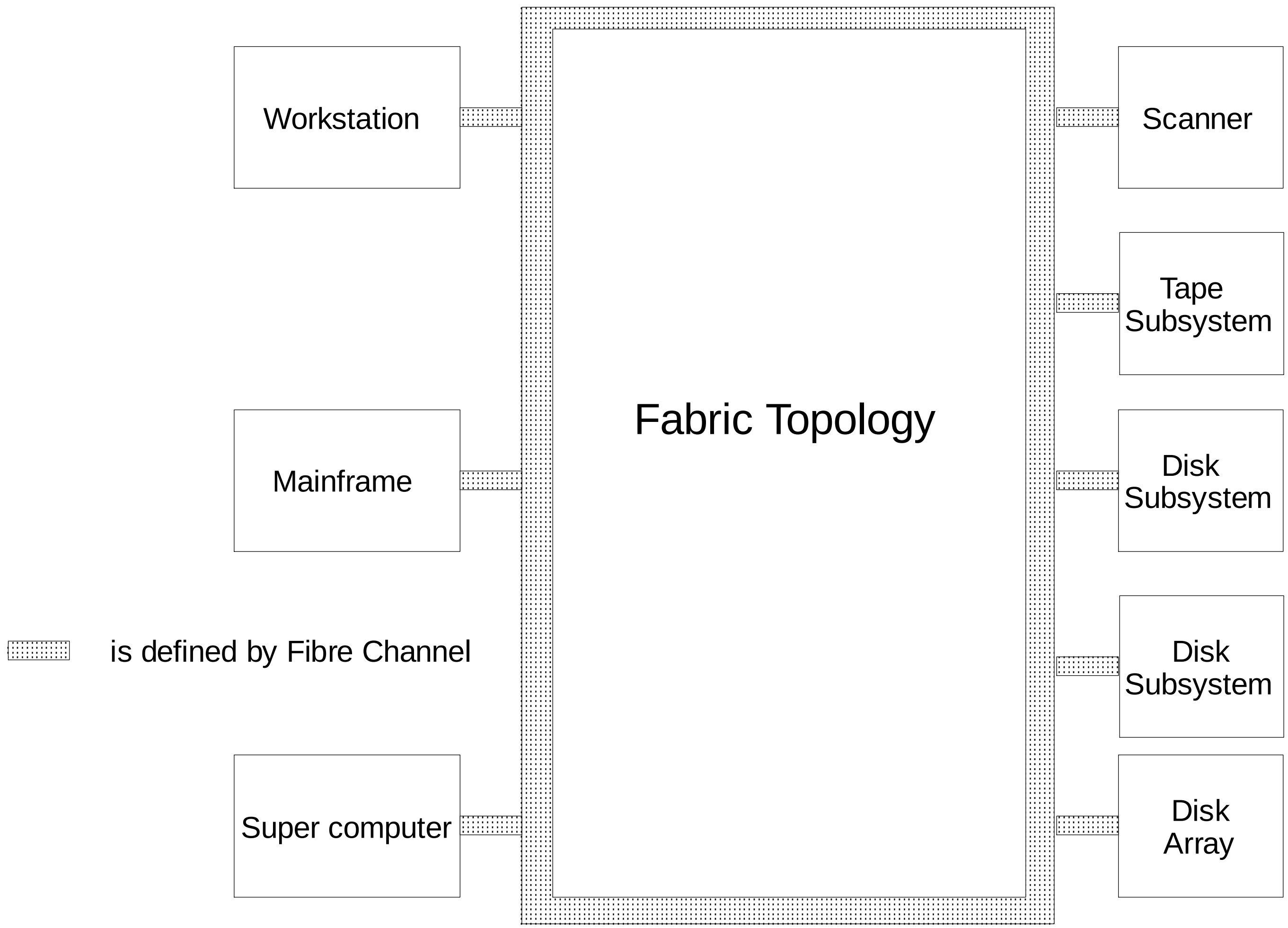
# PROPOSED SOLUTION

- Share a common I/O port
- Share the media
- Reduce hardware costs - recurring and non-recurring
- Reduce system size because fewer ports (multiplex and share bandwidth)
- System Integrators already support two or more interfaces
- Software in place for all supported interfaces
- Make Fibre Channel a common, efficient transport system
- Make Fibre Channel independent of the command set used

# BASIC TERMS


F_Port
F_Port
F_Port
F_Port
A Link
FABRIC
ELEMENT
F_Port
F_Port
N_Port
NODE
F_Port
F_Port
N_Port

# FIBRE CHANNEL ARCHITECTURE
# SHOWING CROSS-POINT SWITCH TOPOLOGY



# MORE TERMS

| | | | | |
|---|---|---|---|---|
| SCSI | has | Initiators | and | Targets |
| IPI | has | Masters | and | Slaves |
| HIPPI | has | Sources | and | Destinations |
| FC | has | **Originators** | and | **Responders** |

- Searched for new names with no prior (emotional) connotations
- Other words with special meaning IN Fibre Channel:

| | | |
|---|---|---|
| **Connection** | **Exchange** | **Sequence** |
| **Dedicated** | **Datagram** | **Exchange** |
| **Login** | **Multiplex** | **Intermix** |
| **Fabric** | **Port** | **Frame** |

# CHANNEL TRANSFER PROTOCOLS

| | | | | |
|---|---|---|---|---|
| SCSI | is | Half-Duplex | Control is | Half-Duplex |
| IPI | is | Full-Duplex | | Half-Duplex |
| HIPPI | is | Full-Duplex | | Simplex |
| FC | is | Full-Duplex | | Full-Duplex |

- In FC, information can flow in both directions simultaneously
  - **Full-Duplex**
  - Transmit on one Fiber, separately Receive on the other Fiber
- In the future,  Wave Division Multiplexing may allow transfer in both directions on one Fiber

# DATA TRANSFER PHILOSOPHY

- Very different from existing peripheral and parallel channel interfaces
- Full bidirectional operation:
  - Separate fibers for each direction
  - Separate "transmit state machine" and "receive state machine"
  - Receive state machine operates on theory of "maximum surprise"
- Symmetrical N_Port designs for computers and peripherals:
  - Universal connectivity makes SCSI and IPI third-party copy finally useful!

# DATA TRANSFER PHILOSOPHY

- Fully bidirectional operation - separate fibers for each direction, can transmit and receive simultaneously
- Structured organization:
  - Supports multiple physical variants at 132, 266, 531 and 1062.5 megabaud (defined to carry payload of 12.5, 25, 50 and 100 megabytes/s respectively)
  - Encode/Decode scheme uses IBM patented 8B/10B code (one-time license fee of $5K already offered)
  - Framing Protocol supports:
    - Variable-length frames
    - Hardware disassembly/reassembly of sequences (chunks)
    - Control of Fabric operation by delimiters
    - Small built-in command set to provide configuration management, support error recovery etc.
    - ACK and BSY frames formatted for hardware processing

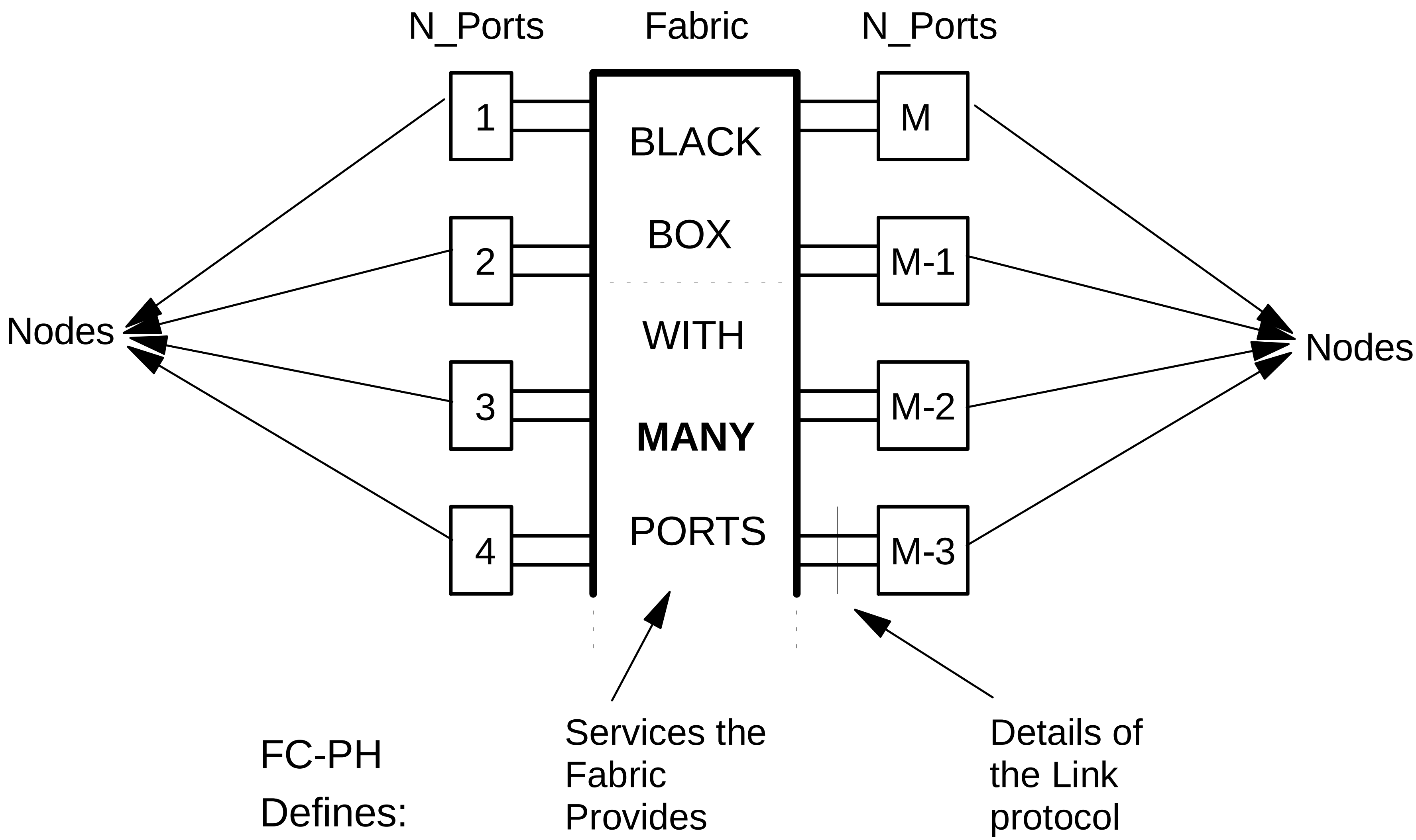
# CLASSES OF SERVICE

- Class 1
  - Dedicated Connection between two N_Ports
  - Guaranteed delivery
  - Frames received in transmitted order
- Class 2
  - Frame Switched
  - Buffer-to-Buffer flow control
  - Guaranteed delivery
  - order not guaranteed in the general case
- Class 3
  - Datagrams or "Ship and Pray"
  - Neither Delivery or receipt order guaranteed

# FIBRE CHANNEL TOPOLOGIES

- Point-to-Point
  - Exactly two N_Ports connected together
  - No Fabric Elements present
  - No "Fabric" Services available
- Cross-point switch topology
  - Generic Cross-point switch environment
  - Distributed Tree variant
- Arbitrated Loop topology
  - Low cost attachment of 3 -126 ports
  - Special Environments

# CROSS-POINT SWITCH TOPOLOGY EXAMPLE




FC-PH

Defines:

- Key point - Fabric details are mostly transparent to Nodes

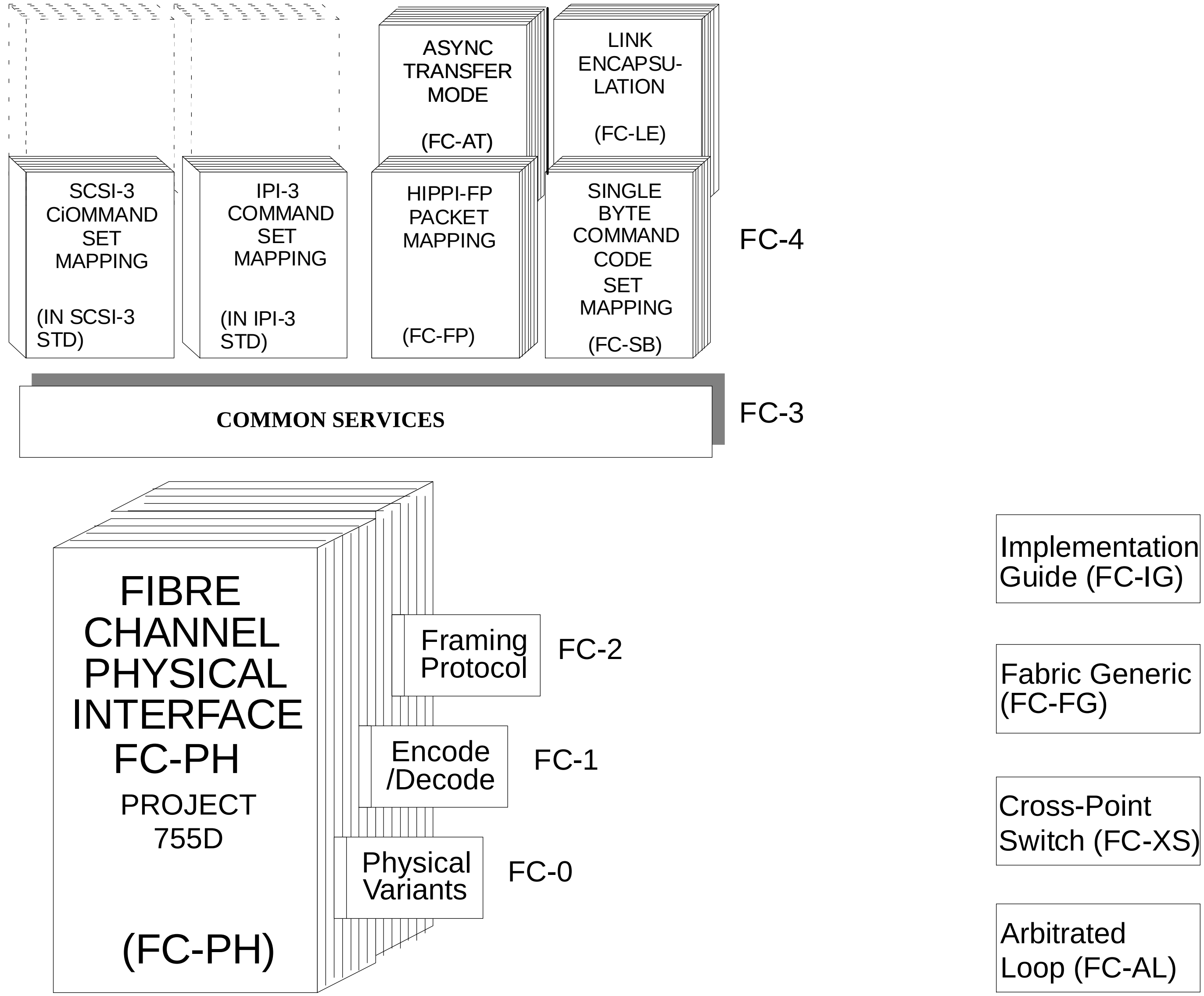
# TOPOLOGY REQUIREMENTS

- Single-level address domain
- "What goes in must come out" (down to frame level)
  - depends on class of service used
- Number of ports only restricted by 24 bit ID fields
  - Some topologies may further restrict this number
- Support heterogeneous Fabric Elements from multiple vendors
- Support Login Protocol to define items in Framing Protocol
  - frame sizes
  - legal number of outstanding frames (credit)
  - etc.
- Attempt to be as transparent to N_Ports as possible

# ARCHITECTURAL LEVELS

- FC-0 Physical
- FC-1 Encode/Decode
- FC-2 Framing Protocol
- FC-3 Common Services
- FC-4 Protocol Mappings
- ULP Upper Layer Protocols

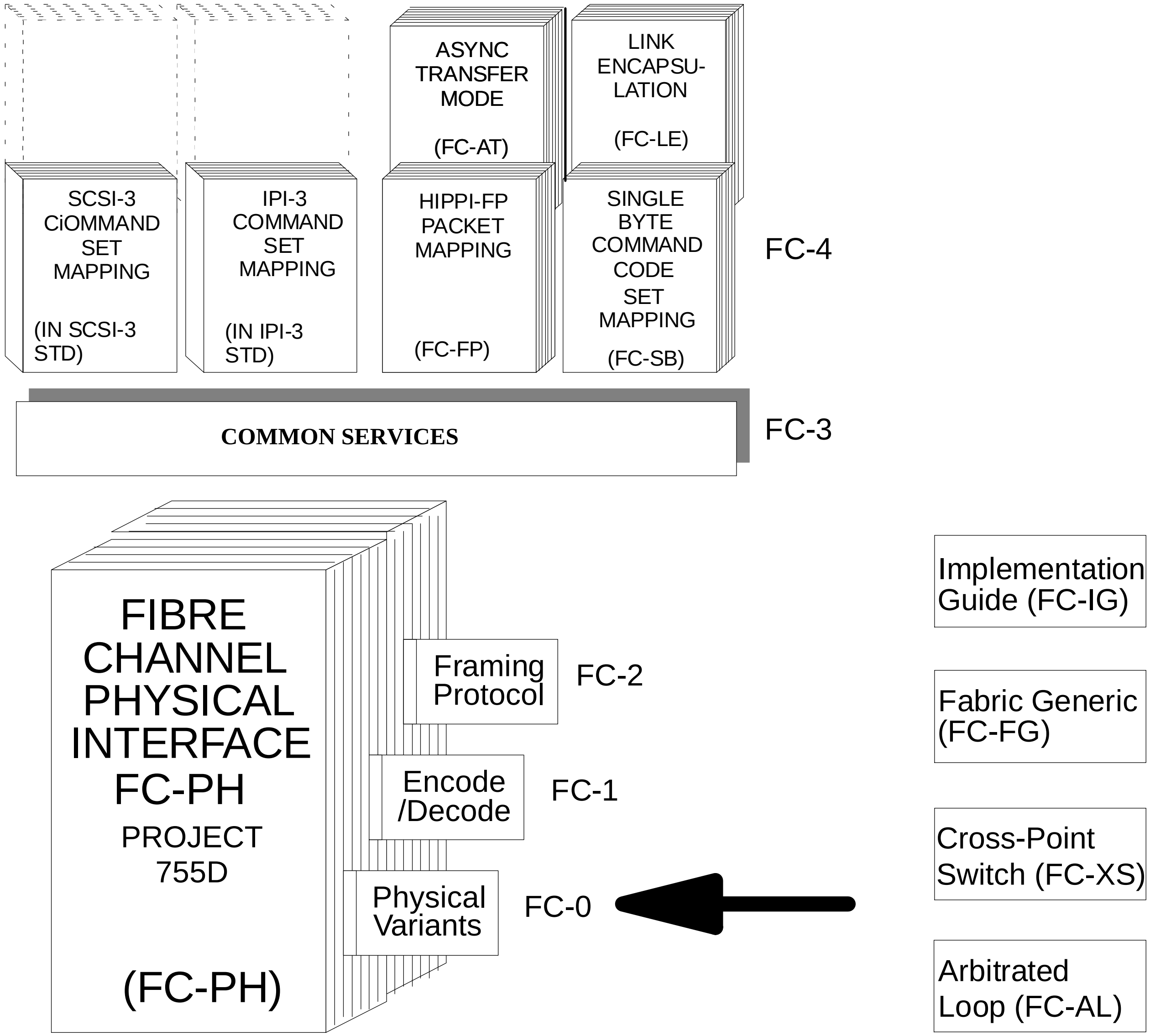
# FIBRE CHANNEL LEVELS AND THE PARTS OF THE STANDARD



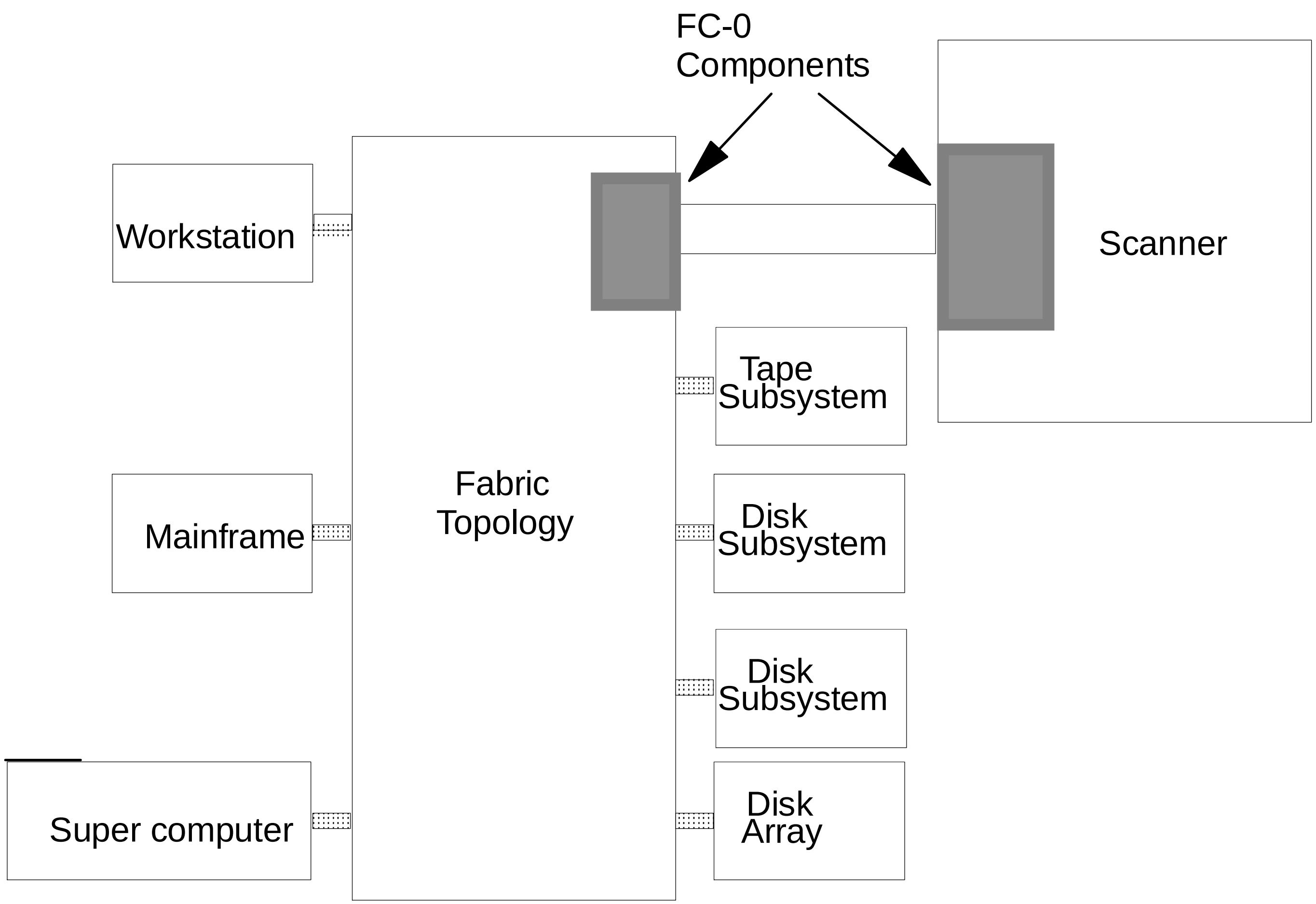
# FC-0 CONNECTORS AND MEDIA

- Introduction
- Connectors
- Variants
  - Fibre
  - Copper
- Physical Interface
- Internal Parallel Interface Recommendation

# FIBRE CHANNEL LEVELS
# PHYSICAL VARIANTS

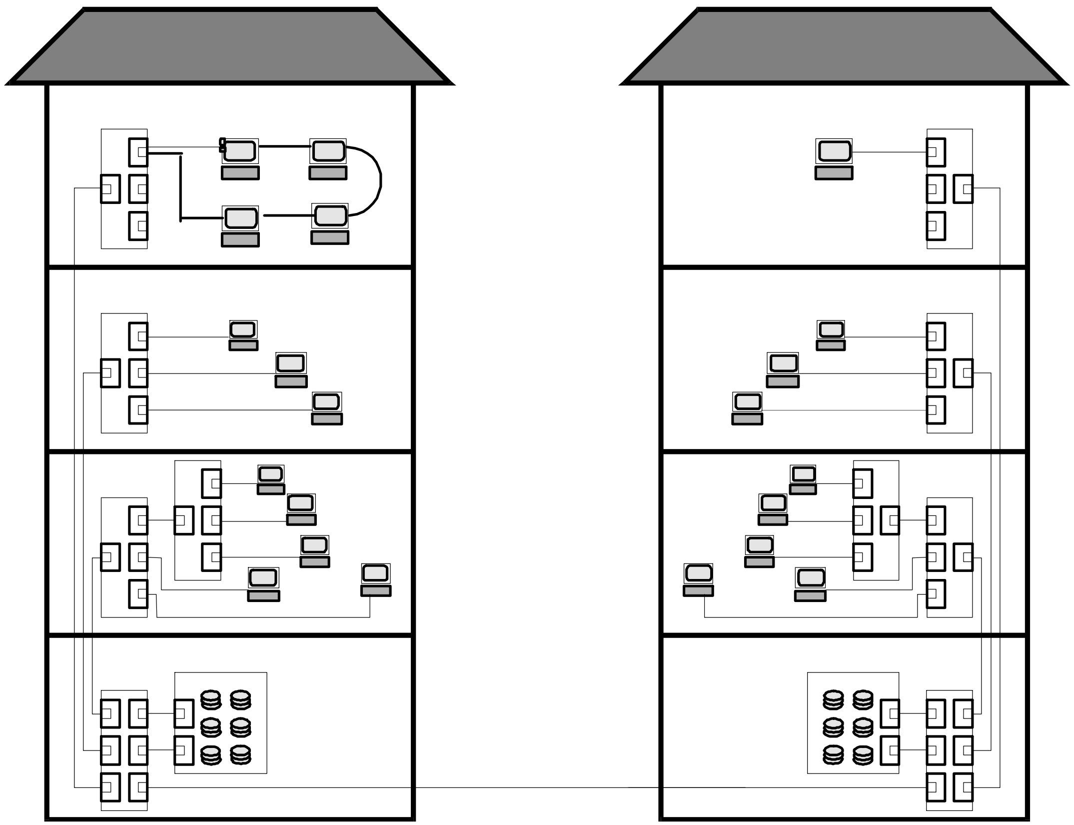


# LOCATION OF FC-0



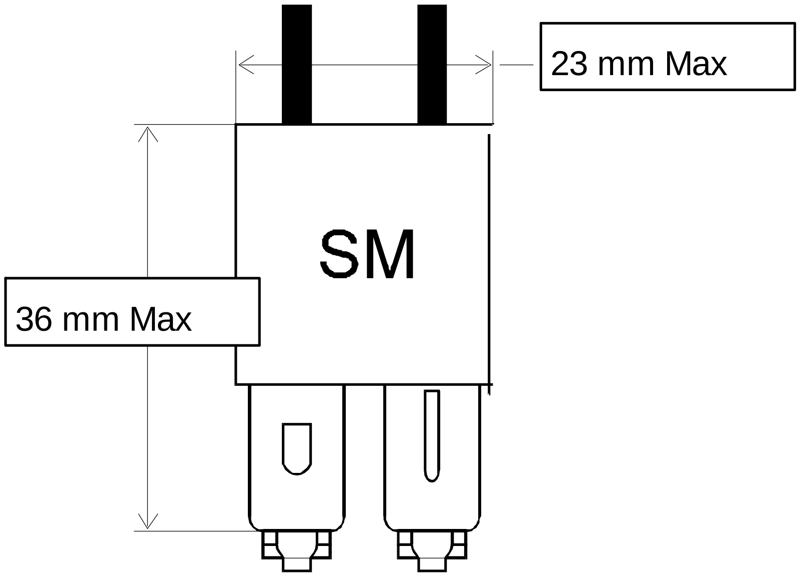

is Fibre
Channel
Interface

# PHYSICAL FLEXIBILITY

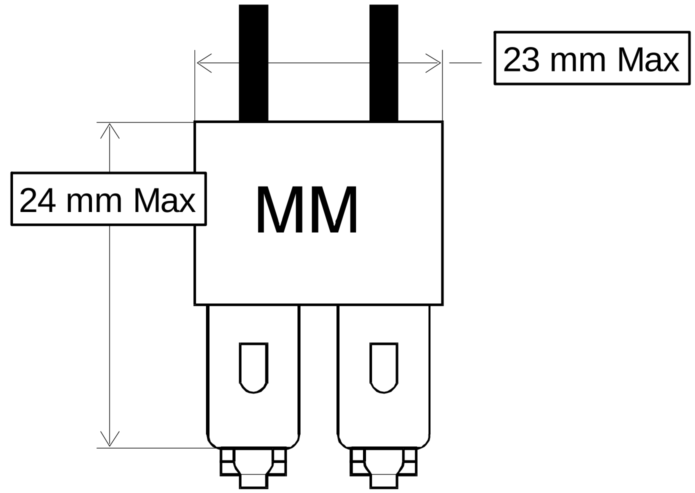


# CONNECTORS

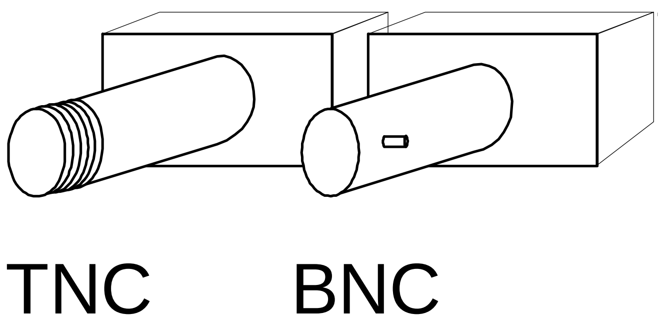
## Single mode SC Connector



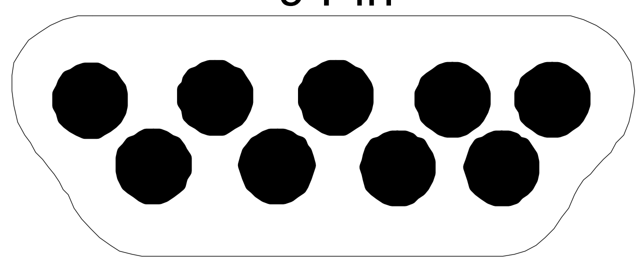

## Multi-Mode SC Connector




## Coax Connector




## STP Connector
## 9 Pin



# PHYSICAL VARIANT NOMENCLATURE

**Each physical variant is described by a symbol made up of one code from each of four groups: 1G-2G-3G-4G such as 100-SM-LL-L**

- **First Group (1G) defines speed**
  - 100  1062.5 megabaud
  - 50  531 megabaud
  - 25  266 megabaud
  - 12.5 132 megabaud.
- **Second Group (2G) defines media**
  - SM  Single Mode
  - M5  Multimode 50 µm
  - M6  Multimode 62.5 µm
  - TV  Video Cable
  - MI  Miniature Cable
  - TP  Twisted pair
- **Third group (3G) transmitter type:**
  - LL  Longwave Laser
  - SL  Shortwave Laser
  - LD  Longwave LED
  - EL  Electrical
- **Fourth group (4G) defines distance:**
  - L  Long Distance
  - I  Interim Distance
  - S  Short Distance

# 100 MB/S PHYSICAL VARIANTS

**1062.5 MBaud serial signaling rate**

| Variant | Operating Range (kilometer) | Transmitter | Media Type |
|---|---|---|---|
| 100-SM-LL-L | 0.002 - 10 | 1300 nm laser | 9 um single mode |
| 100-SM-LL-I | 0.002 - 2 | 1300 nm laser | 9 um single mode |
| 100-M5-SL-S | 0.002 - 0.5 | 780 nm laser | 50 um multimode |
| 100-TV-EL-S | 0 - 0.025 | ECL | 75 ohm video coax |
| 100-MI-EL-S | 0 - 0.010 | ECL | 75 ohm miniature coax |

# 50 MB/S PHYSICAL VARIANTS

## 531.25 MBaud serial signaling rate

| Variant | Operating Range (kilometer) | Transmitter | Media Type |
|---|---|---|---|
| 50-SM-LL-L | 0.002 - 10 | 1300 nm laser | 9 um single mode |
| 50-M5-SL-I | 0.002 - 1 | 780 nm laser | 50 um multimode |
| 50-TV-EL-S | 0 - 0.050 | ECL | 75 ohm video coax |
| 50-MI-EL-S | 0 - 0.020 | ECL | 75 ohm miniature coax |

# 25 MB/S PHYSICAL VARIANTS

## 265.625 MBaud serial signaling rate

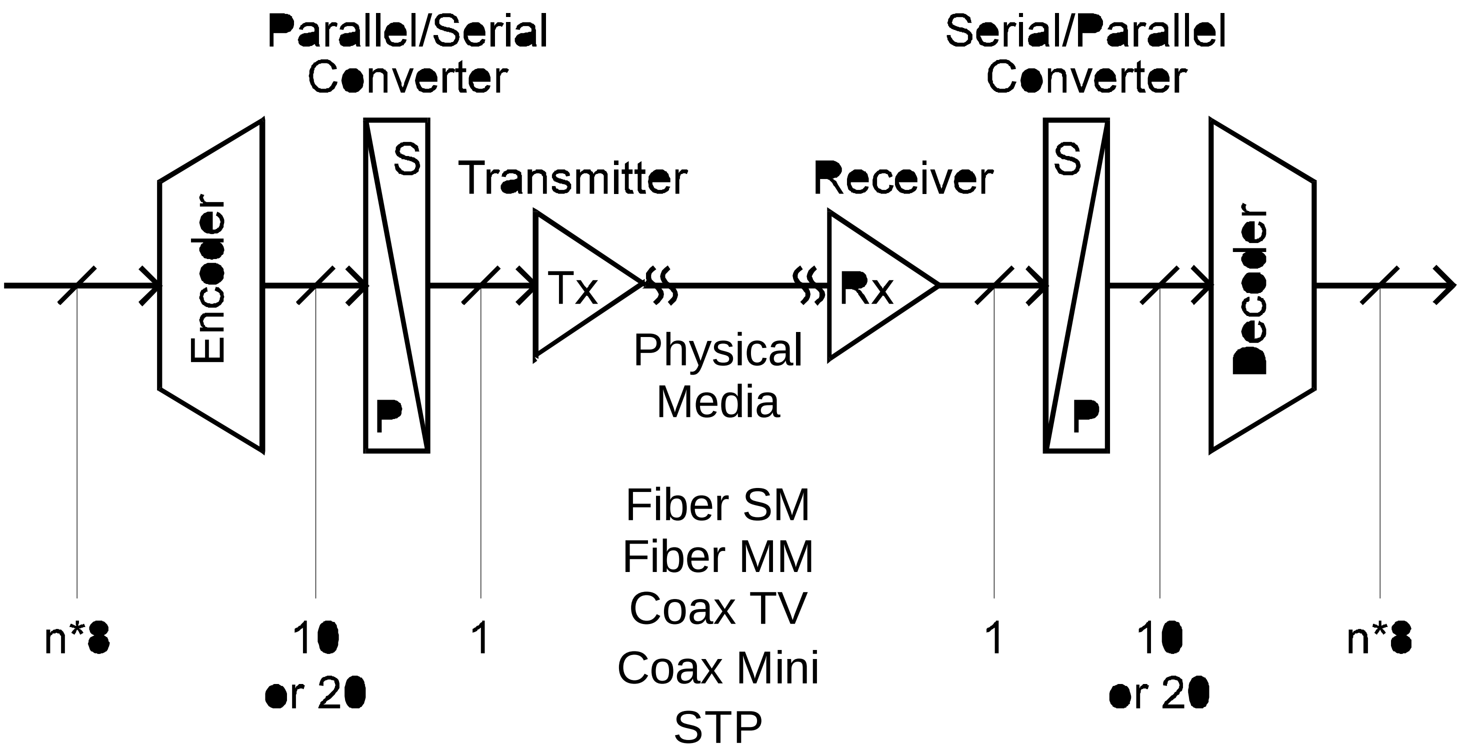
| Variant | Operating Range (kilometer) | Transmitter | Media Type |
|---|---|---|---|
| 25-SM-LL-L | 0.002 - 10 | 1300 nm laser | 9 um single mode |
| 25-SM-LL-I | 0.002 - 2 | 1300 nm laser | 9 um single mode |
| 25-M5-SL-I | 0.002 - 2 | 780 nm laser | 50 um multimode |
| 25-M6-LE-I | 0.002 - 1.5 | 1300 nm LED | 62.5 um multimode |
| 25-TV-EL-S | 0 - 0.075 | ECL | 75 ohm video coax |
| 25-MI-EL-S | 0 - 0.030 | ECL | 75 ohm miniature coax |
| 25-TP-EL-S | 0 - 0.050 | ECL | "Type 1" 150 ohm shielded twisted pair |

# 12.5 MB/S PHYSICAL VARIANTS

**132.813 megabaud serial signaling rate**

| Variant | Operating Range (kilometer) | Transmitter | Media Type |
|---|---|---|---|
| 12.5-M6-LE-I | 0.002 - 1.5 | 1300 nm LED | 62.5 um multimode |
| 12-TV-EL-S | 0 - 0.100 | ECL | 75 ohm video coax |
| 12-MI-EL-S | 0 - 0.040 | ECL | 75 ohm miniature coax |
| 12-TP-EL-S | 0 - 0.100 | ECL | "Type 1" 150 ohm shielded twisted pair |

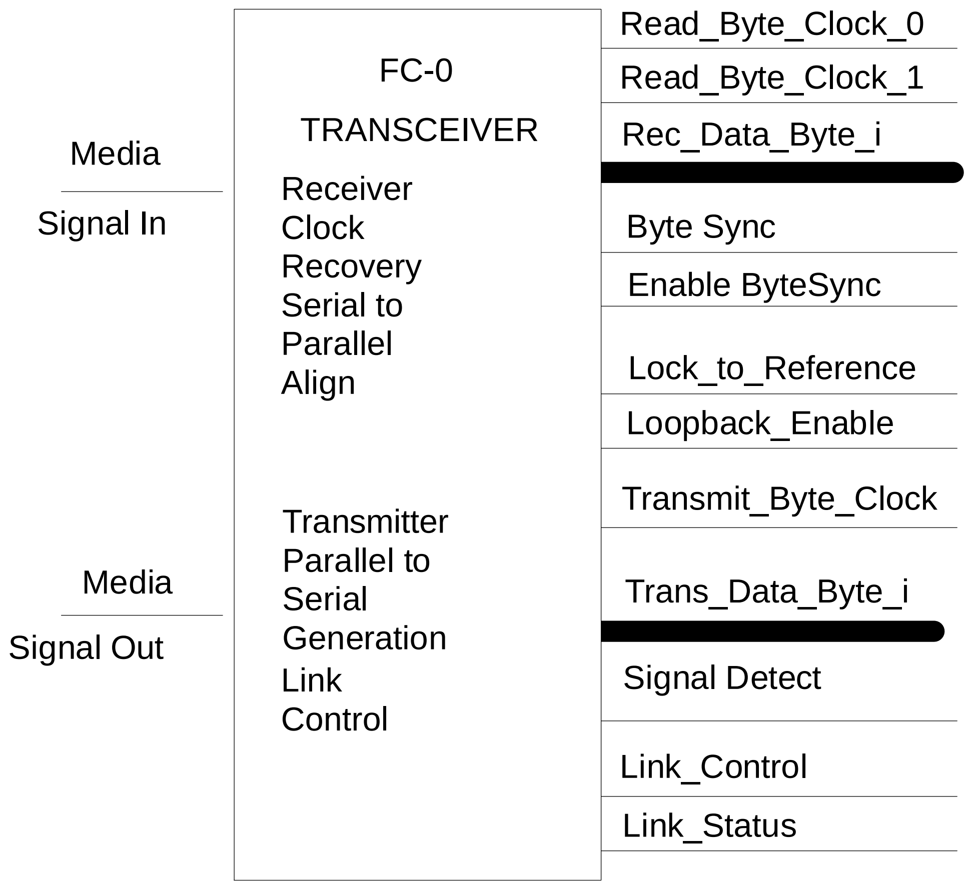
# INTERFACE OVERVIEW




| | | |
|---|---|---|
| 100 MegaByte/sec | = | 800 Megabaud |
| 8B/10B Encoded | = | 1000 Megabaud |
| +Overhead (6.2%) | = | 1.062 Gigabaud |

# PARALLEL INTERFACE

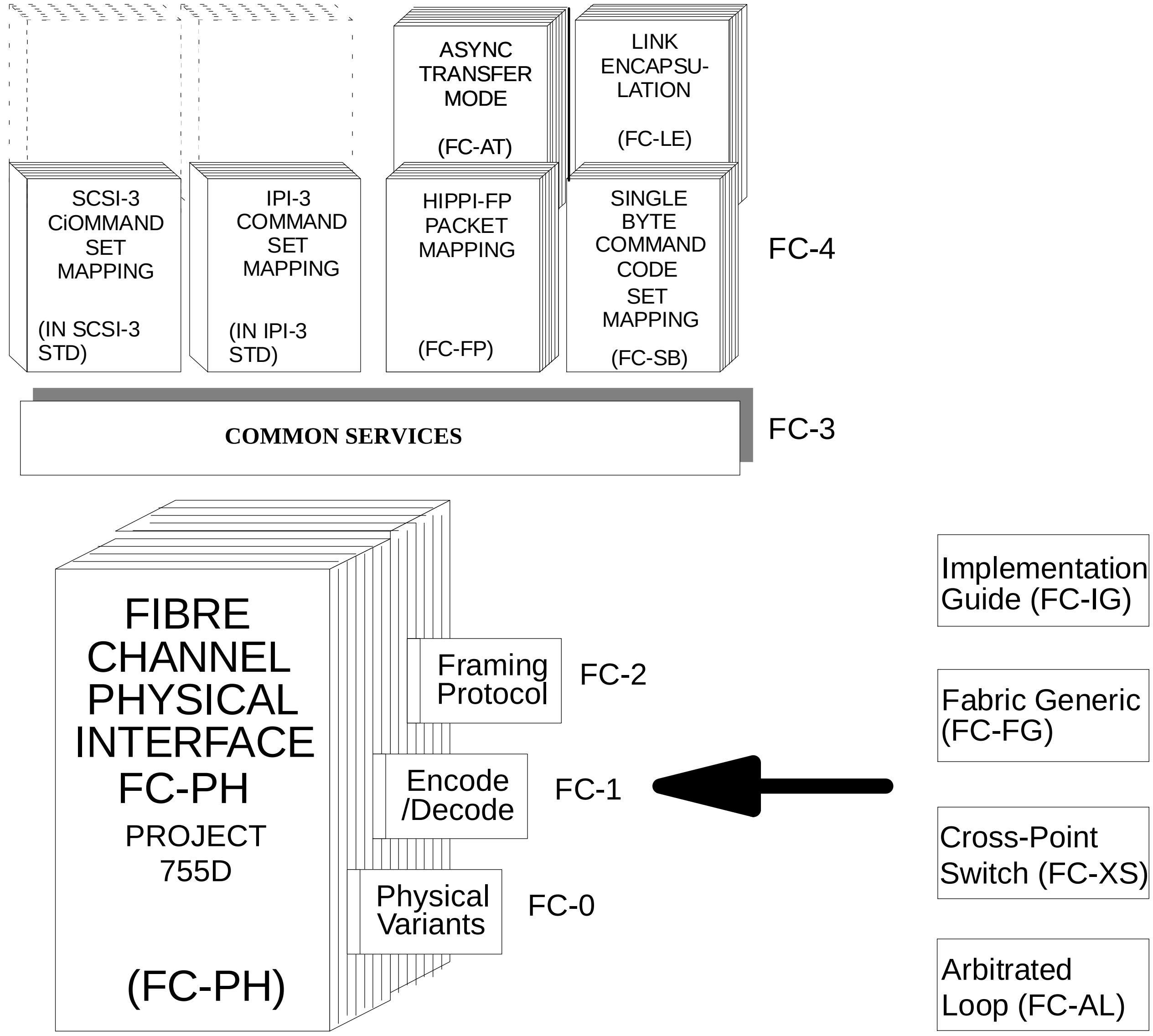
- **Recommendation only, not requirement**
- **Only covers data, not Link Control or Link Status**



# FC-1 ENCODE / DECODE

- Origin
- Notation
- Running Disparity
- Error Detection
- Special Characters
- Words
- Ordered Sets

# FIBRE CHANNEL LEVELS
# ENCODE/DECODE



# 8B/10B CODING SCHEME ORIGIN

- 8B/10B Scheme is patented by IBM
- Improve the transmission characteristics of data:
  - Maintains dc balance (same number of 1s and 0s) for easier receiver design
  - Provides good transition density for easier clock recovery
  - Allows error checking by unrecognized codes, Running Disparity (see below)
- Eight internal bits (one byte) transmitted as a 10 bit group over the media

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Internal Data Byte Bit Positions | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| 8B/10B Bit Designations | H | G | F | E | D | C | B | A | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Internal 8B/10B Character | A | B | C | D | E | F | G | H | | |
| Transmitted 10 bit character | a | b | c | d | e | i | f | g | h | j |

where:
a=A etc.
i is derived from ABCDE, j from FGH
i,j obtained from tables
a is transmitted first

# 8B/10B CODING SCHEME NOTATION

- Symbol  D25.7  where:
  - First character defines type:
    D for normal data characters
    K for special characters
  - Second and Third characters are decimal value of bits EDCBA for Data characters
  - Fifth character is decimal value of bits HGF for Data Characters

- Example:

| | H G F E D C B A |
|---|---|
| • Internal Data | 0 1 0 0 0 1 0 1  (hex 45) |
| • internal 8B/10B | 1 0 1 0 0 0 1 0 |
| • 10 bit character bits notation | 1 0 1 0 0 1 0 1 0 1<br>a b c d e i f g h j<br>A B C D E F G H |
| | E D C B A H G F<br>0 0 1 0 1 0 1 0 |
| • FC-PH notation | **D(0)5.2** |

  - Second, Third and Fifth characters for Special Characters are more complex
  - Decimal value of original data patterns which would produce similar bit patterns after encoding to those in special characters

# 8B/10B CODING SCHEME
# RUNNING DISPARITY

- Disparity is difference between number of 1s and 0s in character
- Running Disparity (RD) is cumulative Disparity of all previous characters
  - Calculated on sub-blocks
    - 6 bit (abcdei)
    - 4 bit (fghj) sub-blocks
  - +1 if more 1s than 0s (or 000111 or 0011)
  - -1 if more 0s than 1s (or 111000 or 1100)
  - Else previous RD unchanged
- Different encodes if RD if +ve or -ve at start of character
  - i,j chosen such that RD alternates across characters of non-zero Disparity
  - Allows error detection when number of errors 0 1 different from number of errors 1 0 (but does not necessarily identify character in error)

# 8B/10B CODING SCHEME
# ERROR DETECTION EXAMPLE

| | rd | | rd | | rd | | rd |
|---|---|---|---|---|---|---|---|
| Internal Data | | hex 35 | | hex 4A | | hex B7 | |
| Transmitted Character | - | D21.1 | - | D10.2 | - | D23.5 | + |
| Transmitted Bit Stream | - | 101010 1001 | - | 010101 0101 | - | 111010 1010 | + |
| Bit Stream after error | - | 101010 10*1*1 | + | 010101 0101 | + | 111010 1010 | + |
| Decoded Character | - | D21.0 | - | D10.2 | + | Code violation | + |

# 8B/10B CODING SCHEME
# SPECIAL CHARACTERS

- **Twelve special characters defined by 8B/10B scheme**
  - only K28.5 used by FC
- **K28.5 contains a "comma" - has special alignment characteristics:**
  - Bit pattern is
    - Current RD is -ve `001111 1010`
    - Current RD is +ve `110000 0101`
  - Run length of 2 followed by opposite polarity run length of 5 is the comma

***Comma is unique and defines byte synchronization***

- **Only one other usable special character contains the comma (K28.1)**

# 8B/10B CODING SCHEME
# WORD DEFINITIONS

- **Word is defined as group of four bytes**
- **Frame delimiters, Idle words etc., called Ordered Sets, have special structure**

| | | | |
|---|---|---|---|
| K28.5 | D21.4 OR D21.5 | Dxx.x | Dxx.x |

Left — Right

- K28.5 followed by three data characters
- Chosen for good spectral characteristics and coding distances
- Third and Fourth characters normally identical for checking
- Provide Low-Level Function and Signalling

# 8B/10B CODING SCHEME - WORD DEFINITIONS (Cont'd)

- **Frame-delimiter ordered sets are:**
  - SOF Connect Class 1 (SOFc1)
  - SOF Initiate Class 1 (SOFi1)
  - SOF Normal Class 1 (SOFn1)

  - SOF Initiate Class 2 (SOFi2)
  - SOF Normal Class 2 (SOFn2)

  - SOF Initiate Class 3 (SOFi3)
  - SOF Normal Class 3 (SOFn3)

  - SOF Fabric (SOFf)

  - EOF Normal (EOFn)

  - EOF Terminate (EOFt)
  - EOF Disconnect Terminate (EOFdt)

  - EOF Abort (EOFa)

  - EOF Normal Invalid (EOFni)
  - EOF Disconnect Terminate Invalid (EOFdti)

- **EOF Ordered Sets always result in negative Running Disparity**

- **SOF, EOF used to control functions of Fabric**

# 8B/10B CODING SCHEME - WORD DEFINITIONS (Cont'd)

- **Primitive Signal Ordered Sets are:**

  **Idle** | **Receiver Ready (R_RDY)**

  **Loop Primitive Signals (ARB, OPN, CLS, SYN)**

  - Have meaning when received singly
  - Idles may be inserted or deleted singly for clock elasticity purposes
  - R_RDYs used in buffer-to-buffer flow control (one R_RDY transmitted in reverse direction for each frame buffer freed)

- **Primitive Sequence Ordered Sets are:**

  **Offline (OLS)** | **Not_Operational (NOS)**

  **Link_Reset (LR)** | **Link_Reset_Response (LRR)**

  **Loop Primitive Sequences (LIP, PBD, PBE)**

  - Need to receive consecutive number before have meaning
  - Used in initialization and error recovery

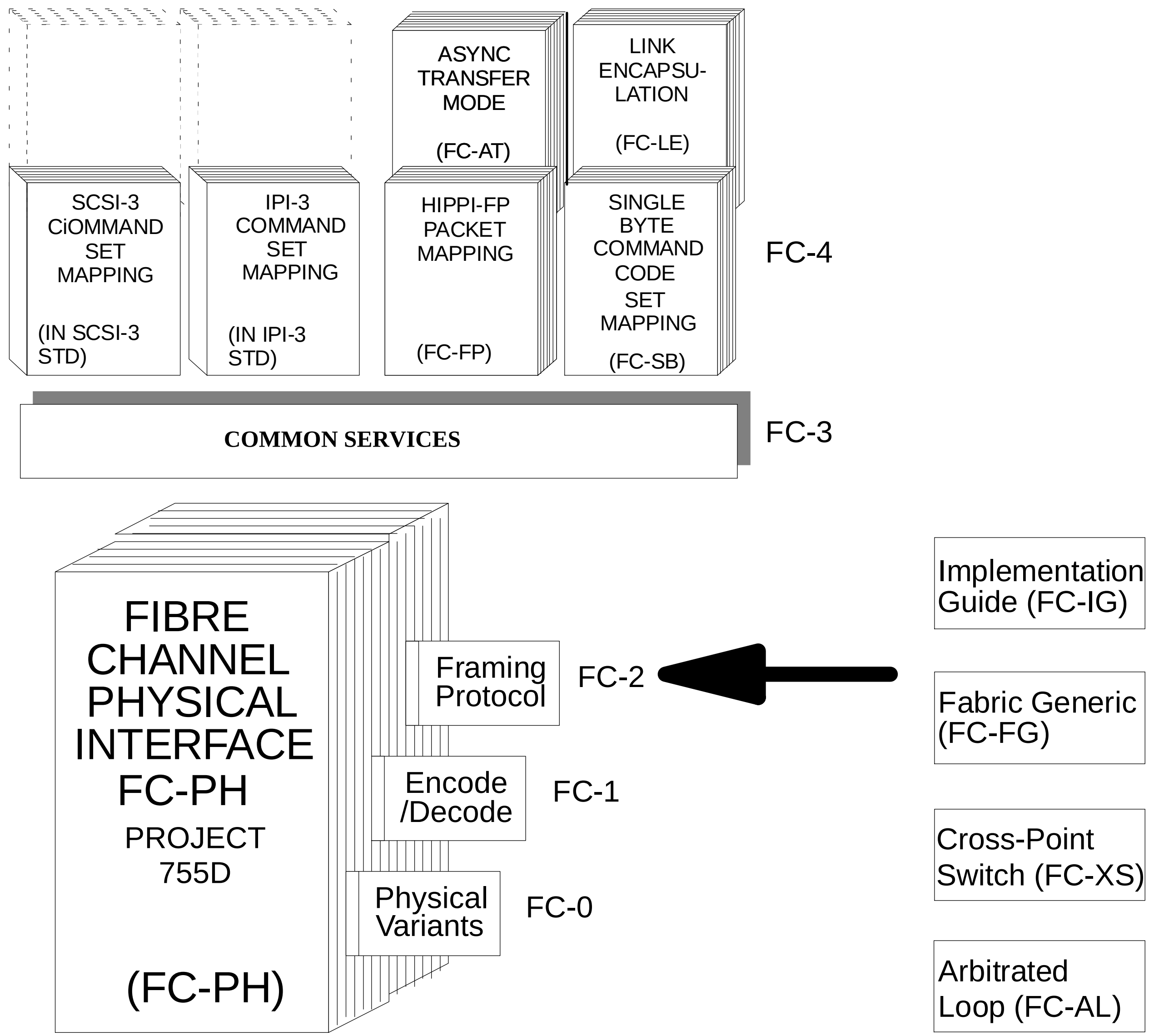
# 8B/10B CODING SUMMARY

- Robust scheme
- Simplifies receiver design
- Good spectral characteristics
- Limited run lengths
- Useful error detection capability
- Implementable in simple logic on byte-parallel data:
  - Encoder in the order of 100 gates
  - Decoder in the order of 85 gates
  - 50 gates for error detection

# FC-2 FRAMING PROTOCOL

- Major Architectural Features
- Constructs - Part One
  - Frames
  - Link Control Frames
  - Data Flow
- Constructs - Part Two
  - Exchanges
  - Multiplexing
  - Classes of Service
  - Common FC-4 Functions

# FIBRE CHANNEL LEVELS
# FRAMING PROTOCOL



# FRAMING PROTOCOL
# MAJOR PARTS

- Disassembly of information defined by FC-4s into frames and Reassembly from frames
- "Built-in" protocol to aid in managing link operation, control configuration, obtain status etc.
- Look-ahead flow control and error handling scheme for both of the above
- Key philosophy: disassembly/reassembly, flow control, error handling done in simple hardware independent of Upper Layers

# FRAMING PROTOCOL OVERVIEW

- Definition of Basic Components
- Frame Structure / Types
- Frame Header (Addresses, Sequence IDs, Exchange IDs)
- Optional Headers
- Link Level Control (ACK, Ready, Busy, Reject)
- Data Flow Control
- Segmentation, Recovery
- Sequence and Exchange management
- Multiplexing management
- Connection management
- Classes of Service
- Login / Logout

# FRAMING PROTOCOL - CONSTRUCTS

**FC-2 defines the following constructs:**

- **Ordered Set:** Consists of 4 10-bit characters
  - Combination of data and special transmission characters
  - Provide frame demarcation
  - Provide very low-level function and signaling between two ends of a link

- **Frame:** Consists of:
  - Start-of-Frame delimiter (SOF)
  - Frame Header
  - Optional Headers
  - Payload
  - CRC
  - End-of-Frame delimiter (EOF)
  - Each frame or group is frames is acknowledged:
    - Flow control
    - Delivery Notification (Class 1 and 2 only)
  - Frame is smallest unit of Transfer
    - Can be streamed within a Sequence under control of Credit
    - Size is a function of implementation - not of protocol

# FRAMING PROTOCOL - CONSTRUCTS (Cont'd)

- **Sequence**
  - Composed of 1-n Frames
  - Unidirectional set of frames for an operation
  - Recovery boundary
  - Each Sequence is identified by initiator: Sequence Identifier (SEQ_ID)
  - Each frame within a Sequence is numbered: Sequential Count (SEQ_CNT)

- **Exchange**
  - Composed of 1-n non-concurrent Sequences
  - Unit- or Bi-directional Flow of Sequences for an Operation
  - Exchange is Identified by each end:
    Originator / Responder Exchange IDs (OX_ID, RX_ID)

- **Operation**
  - Consists of 1-n Exchanges
  - Exchanges may be concurrent
  - Operation is optionally identified by each end:
    Originator / Responder Operation Associator (OO_AS, RO_AS)

# FRAMING PROTOCOL
# FRAME STRUCTURE

| | 4 | 24 | 0 to 2112 | 4 | 4 | Bytes |
|---|---|---|---|---|---|---|
| Idle | Start of Frame | Frame Header | Data Field (Optional Headers + Payload) | CRC | End of Frame | Idle |

- Frame Structure consists of:
  - Four byte SOF delimiter
  - Twenty-four byte fixed-format Header
  - Variable-size Data Field:
    - 0 to a maximum of 2112 bytes (2048+64)
    - May contain Optional Headers as well as payload
  - Four byte CRC (same as used in FDDI)
  - Four byte EOF delimiter

# FRAMING PROTOCOL
# FRAME HEADER STRUCTURE

<table>
<tr><th>Byte 0</th><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th></tr>
<tr><td>Bit 31</td><td>23</td><td>15</td><td>7 … 0</td></tr>
<tr><td>R_CTL</td><td colspan="3">DESTINATION_ID</td></tr>
<tr><td>rsvd</td><td colspan="3">SOURCE_ID</td></tr>
<tr><td>TYPE</td><td colspan="3">F_CTL</td></tr>
<tr><td>SEQ_ID</td><td>DF_CTL</td><td colspan="2">SEQ_CNT</td></tr>
<tr><td colspan="2">OX_ID</td><td colspan="2">RX_ID</td></tr>
<tr><td colspan="4">PARAMETER</td></tr>
</table>

# FRAMING PROTOCOL
# FRAME HEADER STRUCTURE

- Fixed-format Header structure consists of:
  - Routing Control (R_CTL) field, containing:
    - Routing bits for Device_Data, Link_Data, Link_Control etc.
    - Information Category (of payload) bits for Uncategorized, Solicited Data, Unsolicited Control, Solicited Control, Unsolicited Data (and others specific to protocols) etc.
  - Destination ID (D_ID) field, containing identifier of destination N_Port (or an Alias, or a well-known address)
  - Source ID (S_ID) field, containing identifier of source N_Port (or an Alias)
  - Data structure Type (TYPE) field, identifying protocol of frame payload contents
  - Frame Control (F_CTL) bit-significant field, identifying Exchange and Sequence Contexts, First and Last Frame of Sequence, End of Connection, Chained Sequence, Sequence Initiative, Last Sequence, New X_ID assigned, Invalidate X_ID, Continue Sequence condition, Abort Sequence condition, Relative Offset validity, Number of fill bytes
  - Sequence ID (SEQ_ID) field, containing unsigned binary number of Sequence.

# FRAMING PROTOCOL
# FRAME HEADER STRUCTURE (Cont'd)

- Sequence Count (SEQ_CNT) field, containing unsigned binary number identifying order of frame transmission
- Data Field Control (DF_CTL) field, indicating presence of optional headers (Expiration_Security, Network, Association, Device (size 16,32,64 byte) in Data field (and possible extension of basic frame header))
- Originator Exchange ID (OX_ID) field containing unsigned binary number by which the Originator identifies the Exchange.
- Responder Exchange ID (RX_ID) field containing unsigned binary number by which the Responder identifies the Exchange.
- Parameter field contents vary with type of frame:
  - Device_Data frames contains Relative Offset - offset of first byte of payload in bytes from a base address.
  - Link_Control frames contains specific information

# FRAMING PROTOCOL
# FRAME HEADER TYPE FIELD VALUES

- For Device_Data:
  - Fibre Channel Services
  - IS8802-2 LLC (in order)
  - IS8802-2 LLC/SNAP
  - IPI-3 Master
  - IPI-3 Slave
  - IPI-3 Peer
  - CP IPI-3 Master
  - CP IPI-3 Slave
  - CP IPI-3 Peer
  - SCSI FCP
  - SCSI GPP
  - HIPPI-FP
  - SBCCS - Channel
  - SBCCS - Control Unit
  - FC-ATM
  - Fabric Services FC-FG/FC-XS/FC-AL/SNMP

- For Link_Data
  - Basic Link Services
  - Extended Link Services
  - FC-4 Services

- For Link_Control
  - Reserved except F_BSY has reason code

# FRAMING PROTOCOL OPTIONAL HEADERS

- Four headers currently defined:
  - **Expiration Security Header** contains eight byte Expiration Time field, one byte Security type, two byte Security length, and four byte Security key fields
    - Expiration Timer used to identify stale frames
    - Security used to identify authorized frames
  - **Network Header** contains seven byte Network Destination ID field, seven byte Network Source ID field, one byte Destination Network Address authority, and one byte Source Network address authority fields:
    - ID fields contain global MAC addresses (world-wide name) in either 48 bit IEEE or 60 bit CCITT formats
    - Authority fields identify administering authority

# FRAMING PROTOCOL OPTIONAL HEADERS (Cont'd)

- **Device Header** specified by protocol being transferred of 16, 32, or 64 bytes in length
- **Association Header** contains 4 eight byte fields containing Originator and Responder Process and Operation Associator fields:
  - Process Associators used for routing behind N_Port
  - Operation Associators used for locating resources dedicated to an operation

- Optional headers are in fixed order to simplify handling in hardware

# FRAMING PROTOCOL
# FRAME TYPES

- Major types are Device_Data, Link_Data, Link_Control:
  - Device_Data frames carry payload associated with protocol identified in TYPE
  - Link_Data frames carry payloads associated with Link_Services
  - Link_Control frames implement flow control, link control, and error detection and handling functions

# FRAMING PROTOCOL
# LINK_RESPONSE FRAMES

- Error detection functions handled by:
  - N_Port Busy (P_BSY), N_Port Reject (P_RJT), Fabric Reject (F_RJT)
  - All contain separate Action code and Reason Code in Parameter field

    Action codes:
    - Retryable or Non-retryable
    - Abort sequence requested
    - etc.

    Reason Codes:
    - Invalid D_ID
    - Unable to accept frame
    - etc.
  - Fabric Busy (F_BSY) stores reason code in most significant 4 bits of TYPE field to allow any frame (Device_Data, Link_Data or other Link _Control) to be recreated from the F_BSY alone

- Error detection and handling intended to be performed in hardware alone

- Simple to create Link_Response frames from incoming frame:
  - Manipulation of frame header only - swap IDs, change Routing control and Type
  - Link_Response frames have no payload

# FRAMING PROTOCOL
# LINK_CONTINUE FRAMES

- Flow control functions handled by:
  - N_Port Acknowledgment of one, n, or all frames in a Sequence (ACK_1, ACK_N, ACK_0)
    - N carried in Parameter field
    - N of zero allowed in special case of "infinite" credit - it ACKs all frames previously received
- Flow control intended to be performed in hardware alone
- Simple to create Link_Continue frames from incoming Device_Data or Link_Data frame
- Link_Continue frames have no payload

# FRAMING PROTOCOL
# FLOW CONTROL

- Credit defines maximum number of frames to be transmitted before Link_Continue frame or R_RDY Primitive Signal received:
  - N_Port maintains separate Credit for Fabric and for each N_Port in communication
  - Credit_Count decremented for each frame transmitted and incremented for each Link_Control (end-to-end credit) or R_RDY (buffer-to-buffer credit) received
  - Credit allocated during Login (may be implicit)
- Estimate Credit procedure provides basis for allocation of end-to-end credit amongst communicating N_Ports:
  - Allocate on basis of distance
  - Intent to allow maintenance of frame streaming (i.e. credit greater than number of frames contained in round-trip delay)

# FRAMING PROTOCOL
# USE OF DEVICE_DATA FRAMES

- Device_Data frames used in Sequences which carry payload associated with and "application" or Upper Layer Protocol or ULP
  - ULP is a command set or protocol which runs on FC (e.g., SCSI, IPI-3, HIPPI-FP.)
  - FC-4 mapping defines Information Units (IUs) containing item(s) defined by application or ULP
  - Information Unit (IU) transferred across FC by one Sequence:
    - IU segmented into Frames by hardware
    - May contain multiple Information Categories
    - Protocol associated with block identified in Type field of each frame (does not change for Sequence)
    - Relative Offset allows block to be internally non-contiguous
  - There is no restriction on the Information Unit contents
  - SEQ_ID, SEQ_CNT, Relative Offset (if used) allow block to be reassembled by hardware at destination
- FC-2 has no connotation of Commands or Replies for Data frames
- Frame size is function of N_Port implementation - not known to application or ULP

# FRAMING PROTOCOL
# USE OF LINK_DATA FRAMES

- Link_Data frames used in Sequences which carry FC's built-in protocol (only Link_Data Types presently defined are Basic, Extended, and FC-4 Link Services):

- Command / Reply model used

- Basic Link Services Command types are:
  - Abort Sequence (ABTS), Remove Connection (RMC), No Operation (NOP)
    - ABTS aborts a failed Sequence
    - RMC abnormally removes a connection
    - NOP has no significance on Sequence
  - Identified by bits in R_CTL
  - May be inserted as single frames in Sequences of other Types

- Replies are Basic Accept and Reject (BA_ACC and BA_RJT)
  - Accept contains payload specific to the command
  - Reject contains reason codes (invalid command, logical busy etc.)

# LinkFRAMING PROTOCOL
# LINK_DATA FRAMES (Cont'd)

- Extended Link Services provide initialization, configuration and status retrieval functions for an FC-2 N_Port and system:
  - Same segmentation and reassembly process used as for Device_Data Information Units above
  - Login (LOGI) command carries service parameters (including world-wide name, classes supported, Credit etc.) to Fabric or N_Port
  - Logout (LOGO) command requests removal of service parameters
  - Abort Exchange (ABTX) command requests "abnormal" termination of other existing Exchange
  - Read Exchange Status Block (RES), Read Sequence Status Block (RSS), Read Connection Status (RCS), Read Link Status (RLS) commands cause status to be returned in Reply Sequence

# FRAMING PROTOCOL
# LINK_DATA FRAMES (Cont'd)

- **Estimate Credit** (ESTC) and Advise Credit (ADVC) used in scheme to estimate distance separating N_Ports. Establish Streaming (ESTS) obtains enough credit for the process to operate.

- **Read Timeout Value** (RTV) command cause two error-recovery time-out values to be returned in Reply Sequence

- **TEST** command contents have no significance

- **ECHO** causes payload to be returned in Accept

- **Request Initiative** (RI) requests that Sequence Initiative be passed for the target Exchange

- **Accept** (ACC) indicates acceptance of command and returns data where appropriate (Service Parameters and World-Wide Name for LOGI, Status Block contents for RCS, RES, RSS, RLS)

- **Link Application Reject** (LA_RJT) indicates rejection of command and returns a reason code and explanation

# FRAMING PROTOCOL LOGIN/LOGOUT

- **Login** allows an N_Port to establish operating environment with:
  - Fabric
  - Another N_Port
- Exchange of service parameters with Fabric:
  - FC-PH version
  - Initial Credit for buffer-to-buffer flow control
  - Largest frame that Fabric can buffer
  - Timeout values
  - Service Classes supported
  - Determine support of FC-PH options:
    - Intermix
    - Stacked Connect Requests
    - Sequential Delivery

# FRAMING PROTOCOL LOGIN/LOGOUT (Cont'd)

- Exchange of service parameters with other N_Port
  - FC-PH version
  - Initial Credit for end-to-end flow control
  - Largest frame that N_Port can receive
  - Number of concurrent Sequences
- Service Classes supported
  - Relative Offset capability
  - Intermix
  - Stacked Connect Requests
  - Sequential ACK transmission
  - X_ID reassignment
  - Initial Processor Associator
  - Information Category support
  - X_ID interlock
  - Error policy

# FRAMING PROTOCOL EXCHANGE USAGE

- **Mapping an Exchange to a ULP construct is optional**
  - Some protocols will not use Exchanges, or use Exchanges in unidirectional manner only (simple derivative case)
- **Exchange is unidirectional or bidirectional construct of non-concurrent Sequences (i.e., one Sequence at a time and one direction at a time):**
  - Required to:
    - Emulate existing half-duplex interfaces
    - Ensure operation of all protocols over FC is distance-insensitive
    - Ensure that generic N_Port hardware will support all protocols

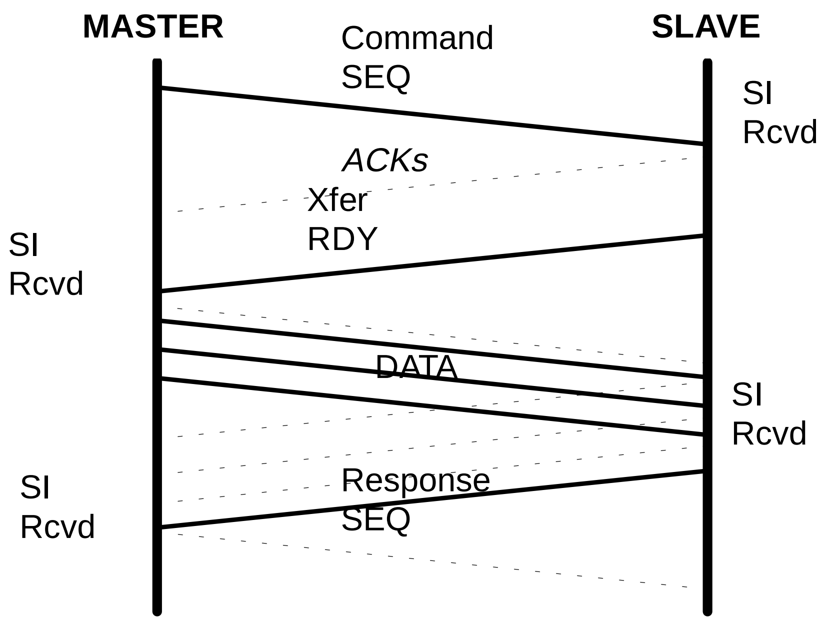
      Provide a generic mechanism for linking of incoming information with previously-sent information (e.g. responses with previously-sent commands)
    - Optimize performance in the case where multiple protocols are supported by a single N_Port

# FRAMING PROTOCOL
# EXCHANGE USAGE (Cont'd)

- Multiple concurrent Exchanges possible per N_Port
- Exchange uniquely identified by concatenation of OX_ID and RX_ID for a specific N_Port
- Normally related to interface control blocks
- Separate OX_ID and RX_ID to allow direct addressing of control blocks at both ends
- In bi-directional case, Sequence Initiative explicitly passed to allow other N_Port to allow generation of Sequence(s)

# FRAMING PROTOCOL
# EXAMPLE OF EXCHANGE USAGE

- In a peripheral command set, an Exchange may be mapped to a single Operation. Example is of Write-type command:




- Sequence Initiative gives explicit permission to transmit:
  - Always passed on last frame of Sequence
- OX_ID, RX_ID allow direct index into control block structure at both ends (allows single-interrupt Operations!)
- For Extended Link Services each  Command / Reply pair is separate Exchange

# FRAMING PROTOCOL
# FRAME MULTIPLEXING

- Conditions established by Login
- Frame size, Classes of Service negotiated
  - Maximum size, permitted size, granularity
- Frames, Sequences and Exchanges can all be multiplexed
  - Full-Duplex operation means Device_Data frames, Link_Data_Requests for one direction mix with Link_Continues, Link_Responses and Link_Data_Replies for other direction

# FRAMING PROTOCOL CLASS DEFINITIONS

- Wide Range of Requirements to address market needs:
- Class 1 is a Dedicated Connection (Circuit Switch):
  - Login with N_Port and Fabric required
  - Requires a turnaround delay to establish
  - Both N_Ports dedicated when Connection complete
  - Once established no F_BSYs or P_BSYs to frames N_Port to N_Port flow control, End-to-end delivery confirmation with ACK
  - Buffer-to-buffer flow control, R_RDYs only used for frame which establishes connection
  - Optional Intermix mode permits Class 2,3 frames if bandwidth available
  - Constructs and protocol defined to determine:
    - When a connection may be terminated
    - How a connection is terminated

# FRAMING PROTOCOL
# CLASS DEFINITIONS (Cont'd)

- Class 2 is a Frame Switch with confirmed delivery:
  - Login with N_Port and Fabric required
  - Connectionless service (no turnaround delay to establish)
  - N_Ports can support Class 2 to multiple other N_Ports simultaneously
  - N_Port to N_Port flow control, End-to-end delivery confirmation with ACK
  - Buffer-to-buffer flow control, delivery confirmation with R_RDY
  - Frame delivery failure notified with F_RJT or F_BSY, P_RJT or P_BSY
  - Each frame routed separately by Fabric
  - if multiple routes supported, frames may be delivered out of order

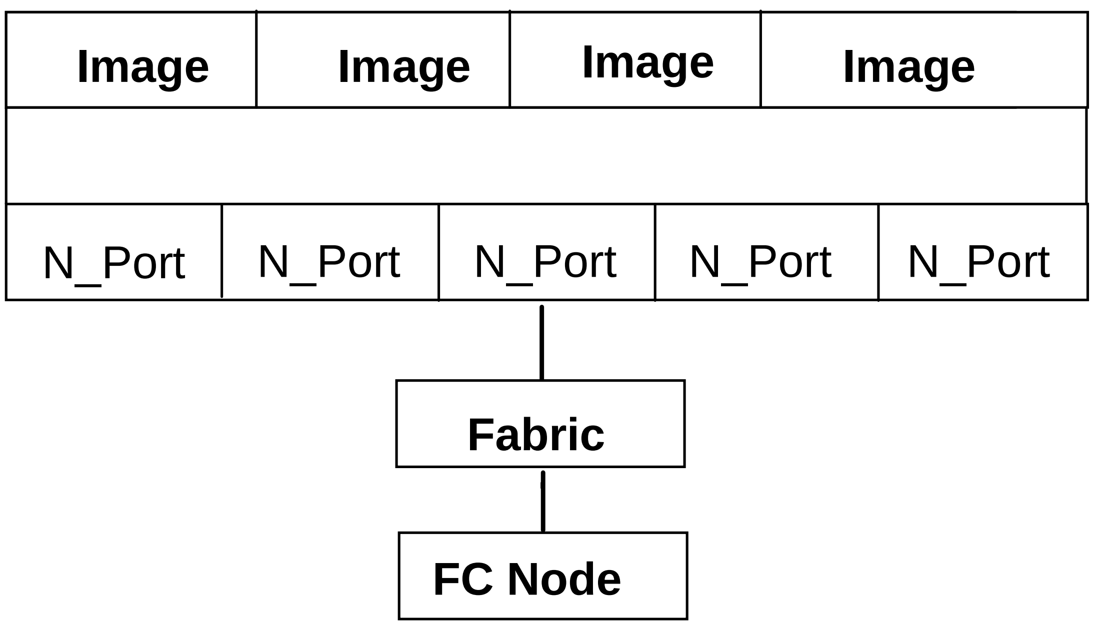
# FRAMING PROTOCOL
# CLASS DEFINITIONS (Cont'd)

- Class 3 is Datagram:
  - Login with N_Port and Fabric optional
  - Connectionless service (no turnaround delay to establish)
  - N_Ports can support Class 3 to multiple other N_Ports simultaneously
  - No N_Port to N_Port flow control, End-to-end delivery confirmation
  - F_Port flow control, delivery confirmation with R_RDY
  - Frame failure not notified (frame discarded and Sequence fails)
  - Detection of reassembly errors up to ULP

# FRAMING PROTOCOL
# USE OF PROCESS ASSOCIATOR

- Used for routing/identification behind N_Port
- Originator and Responder fields
- Identifies "group of processes" behind N-Port
  - E.g., single instance of an operating system
- Used to route incoming frames to correct target "image"
- Used to identify frames from source "image"

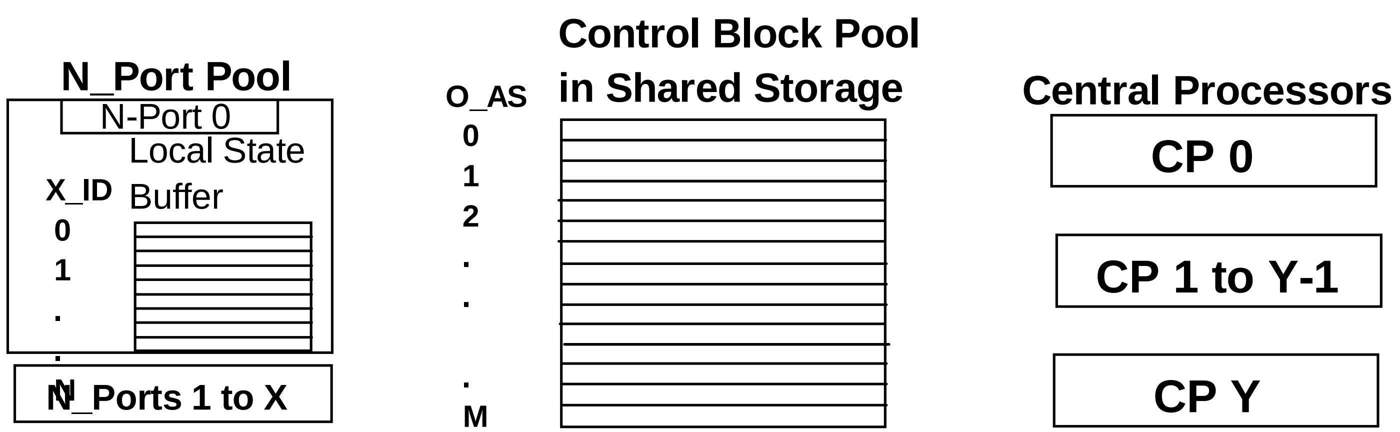
FC Node




Shared across all Images

# FRAMING PROTOCOL
# USE OF OPERATION ASSOCIATOR

- N-Ports may be shared, via control blocks, across physical central processors
- X_ID locates local (N_Port) Exchange resource
- Operation Associator locates shared control block
- Exchanges can become inactive
  - N_Port Resource stored into shared control block
  - X_ID is invalidated
- Allows reuse of local Exchange resources for another Exchange => efficient Exchange multiplexing
- When Exchange is to be reactivated
  - Operation Associator locates shared control block
  - Shared control block loaded into N-Port
  - X_ID is validated



# FRAMING PROTOCOL
# COMMON FC-4 FUNCTION

- Many protocols (IPI-3, SCSI, SBCCS) need same functions
- Common FC functions allow common implementations across protocols
- Common functions:
  - Entity address
  - Data Specification
  - Generic status
- Construct appears as a "header" at the beginning of the payload in the first frame of a Sequence
- Applicable to the following protocols:
  - SBCCS
  - SCSI-FCP
  - CP IPI-3
- Not applicable to the following protocols:
  - IP
  - SCSI-GPP
  - HIPPI-FP
  - IPI-3
- Presence of common FC-4 header is indicated by category field in frame header

# FRAMING PROTOCOL
# COMMON FC-4 - ENTITY ADDRESS

- Provides generic device addressing across FC-4s:
  - SBCCS: device address
  - SCSI-FCP: entity address
  - CP IPI-3: facility address

- Function:
  - When Exchange is Originated, Responder must locate constructs associated with Originator process
  - This information is currently defined by each FC-4
    - In different locations within a packet
    - With differing sizes
  - Entity Address allows TYPE-independent processing of this information

- Entity Address always present in information Category = 6 frame (Unsolicited Command) that is first frame of Sequence

- Entity Address is not present in frame with any other Information Category

# FRAMING PROTOCOL
# COMMON FC-4 - GENERIC STATUS

- Provides generic status handling across FC-4s
  - Command complete/not complete
  - Command successful/not successful

- Function: Allows some status handling to be performed by hardware in a TYPE-independent manner, e.g., chaining

| | |
|---|---|
| **Status** | **4 bytes** |
| **Reserved** | **4 bytes** |

- Status:
  - Bit 31: ULP Command Complete/Incomplete
  - Bit 30: ULP Command Failure/Success

- Reserved: For Transfer Count (proposed):

- Generic Status always present in Information Category = 7 frame (Command Status) that is first frame of Sequence

- Generic Status not present in frame with any other Information Category

# FRAMING PROTOCOL
# COMMON FC-4 - DATA DESCRIPTOR

- Provides generic "transfer ready" function across FC-4s:
  - SBCCS: Command Response
  - SCSI-FCP: Transfer Ready
  - CP IPI-3: TNR

- Function:
  - Indicates CU ready to receive write data or send read data
  - Indicates how many bytes are to be written or read
  - Can be used to initiate out-of-order transfers
  - Data Descriptor may be implicit/null for unambiguous data transfers

| | |
|---|---|
| **Length** | **4 bytes** |
| **Offset** | **4 bytes** |
| **Reserved** | **4 bytes** |

- Data Descriptor always present in Information Category = 5 frame that is the first frame of Sequence

- Data Descriptor not present in frame with any other Information Category

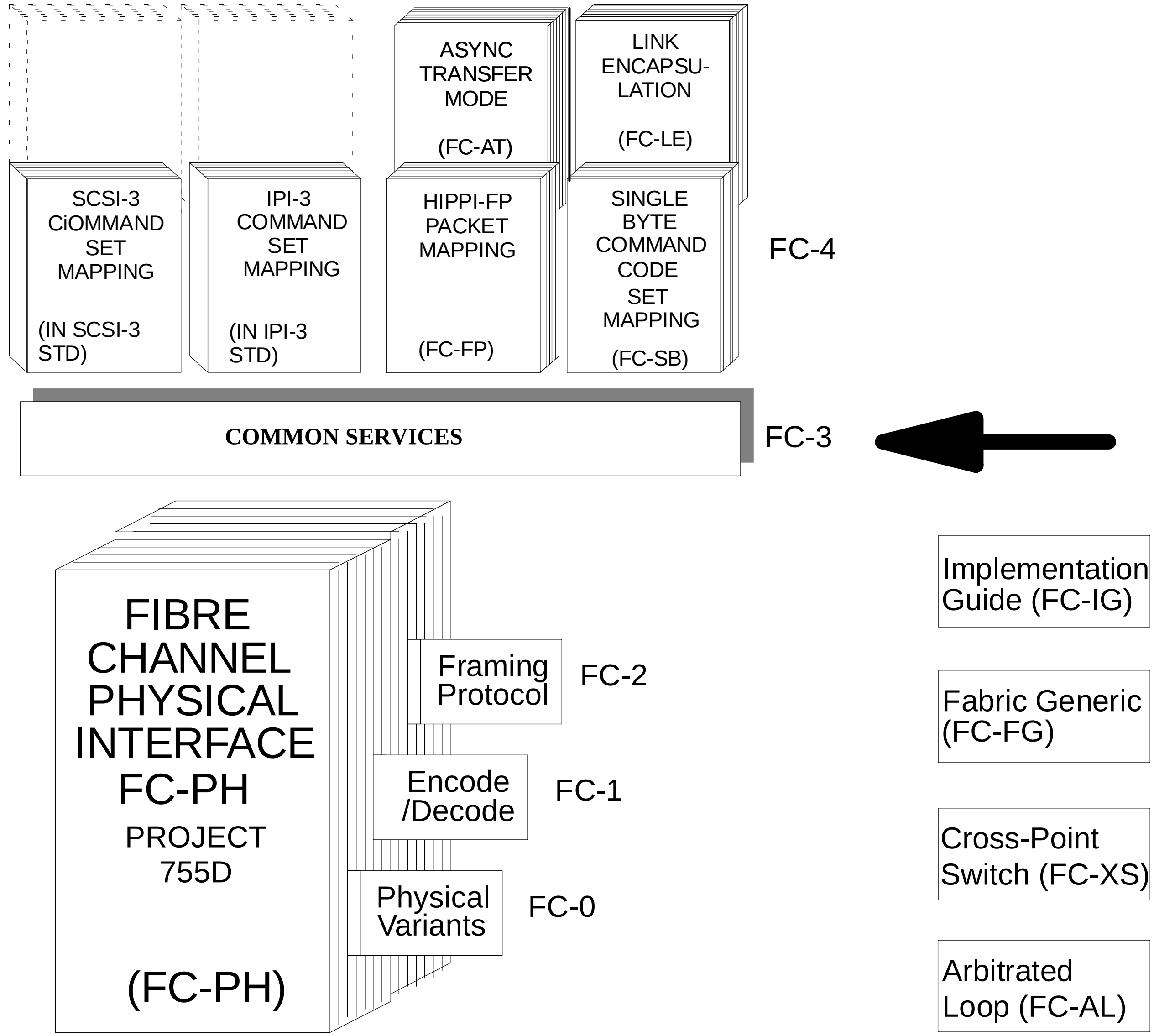
# FRAMING PROTOCOL SUMMARY

- Comprehensive support for a wide variety of protocols
- Multiple levels of control provided
- Support for multiple Classes of operation
- Built-in look ahead Flow Control
- Support for low-level error detection and recovery
- Not expected that any implementation will incorporate all features:
  - Choose appropriate subset (e.g., only Classes required by application)
  - Indicate subset supported by Login mechanism

# FC-3 COMMON SERVICES

- The Least Developed Part of Fibre Channel to Date
- Future Considerations for FC-3 Level
  - Hunt Groups
  - Striping
  - Broadcast
  - Multicast
  - Moviecast

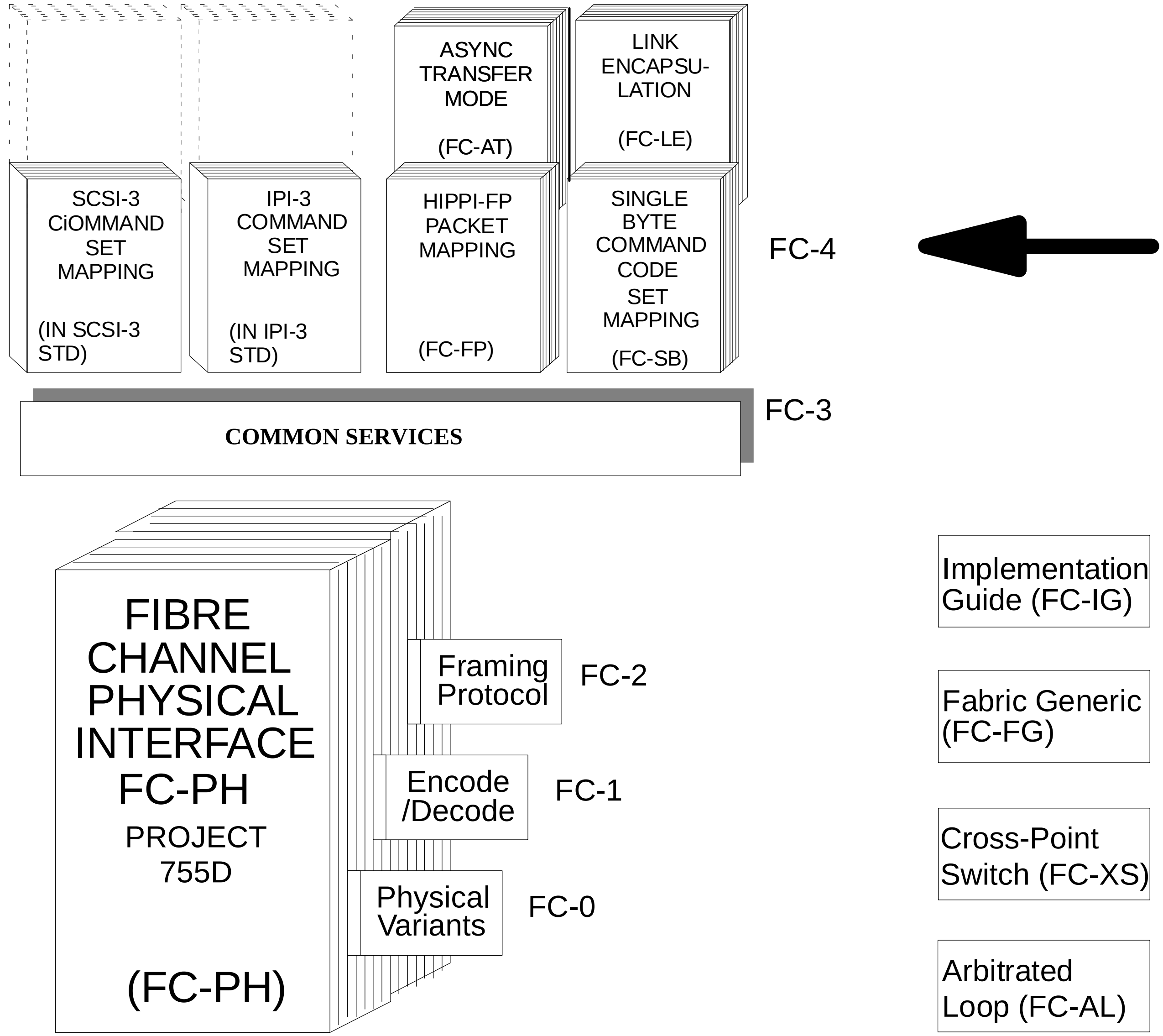
# FIBRE CHANNEL LEVELS
# COMMON SERVICES



# FC-4 PROTOCOL MAPPING

- Definitions
- Intelligent Peripheral Interface -3 (IPI-3)
- Small Computer System Interface (SCSI-3)
  - SCSI Fibre Channel Protocol (SCSI-FCP)
  - SCSI Generic Packetized Protocol (SCSI-GPP)
- Single Byte Common Command Set (FC-SB)
- High Performance Parallel Interface (HIPPI) (FC-FP)
- Link Encapsulation (IP Protocol) (FC-LE)

# FIBRE CHANNEL LEVELS
# PROTOCOL MAPPINGS



# FC-4 CHARACTERISTICS

- Define Information Units (IUs) composed of protocol constructs which are carried across FC:
  - All mappings support IUs of only one Information Category
  - Some mappings may define optional IUs which contain multiple Information Categories (e.g., a write command and the data for that command)
- IU definitions independent of Frame size, Sequence boundaries, Frame header
- IUs define use of some FC-PH constructs:
  - Relative Offset to aid in memory management
  - Sequence Initiative
  - Optional Headers
- Standard FC-4 definition:
  - IU Tables: define IU types and contents
  - IU flow: defines IU protocol flow
  - Service Interface definition: defines FC-4 usage of FC constructs via service interface defined in FC-PH Annex R
- Map existing, stable protocols to FC:

| | | |
|---|---|---|
| IPI-3 | SCSI | SBCCS |
| HIPPI-FP | IP/ARP | |

# FC-4 DEFINITION

Information Units (IUs) sent by a source to a destination defined by an IU table:

| IU Name | Operation Phase | Information_Set_1 | | Information_Set_2 | | FML | SI | M or O |
|---|---|---|---|---|---|---|---|---|
| | | Cat | Payload | Cat | Payload | | | |
| | | | | | | | | |
| | | | | | | | | |

- IU Name: Name of IU for FC-4 reference
- Operation Phase: description of where IU is used in ULP protocol
- Information_Set: Portion of IU that contains single Information Category
- Cat: FC-PH Information Category of Information_Set
- Payload: Description of ULP data carried
- FML: Indication of whether this is First/Middle/Last IU of operation
- SI: Indication of whether Sequence Initiative is Transferred/Held after IU transferred
- M or O: Indication of whether this IU is Mandatory or Optional

# IPI-3/CP IPI-3

- Defined by IPI-3 Working Group (ANSI X3T11)
- IPI-3 Command, Response and Asynchronous Packets mapped to IUs
- Data bursts mapped to IUs
- Classes 1 and 2 supported
- Device Headers not used
- Mappings for Master, Slave, and Peer defined
- IPI-3 Operation => Command, Data, Completion => One Exchange
- IPI-3 mapping does not use Common FC-4 Functions
- CP IPI-3 (Context Processing)
- CP IPI-3 mapping uses Common FC-4 Functions

# IPI-3/CP IPI-3
# INFORMATION UNITS TO SLAVE

| IU Name | Operation Phase | Information_Set_1 | | Information_Set_2 | | FML | SI | M or O |
|---|---|---|---|---|---|---|---|---|
| | | Cat | Payload | Cat | Payload | | | |
| S1 | Command | 2 | One Command packet | - | - | F | T | M |
| S2 | Data | 1 | First or Intermediate IPI-3 Burst | - | - | M | H | M |
| S3 | Data | 1 | Only or last IPI-3 Burst | - | - | M | T | M |
| S4 | Command /Data | 2 | One Write Command packet | 4 | First IPI-3 Burst | F | H | O |
| S5 | Command /Data | 2 | One Write Command packet | 4 | Only IPI-3 Burst | F | T | O |

Note: S1, S4, S5 have Cat=6 for CP IPI-3

# IPI-3/CP IPI-3
# INFORMATION UNITS TO MASTERS

| IU Name | Operation Phase | Information_Set_1 | | Information_Set_2 | | F M L | S I | M or O |
|---|---|---|---|---|---|---|---|---|
| | | Cat | Payload | Cat | Payload | | | |
| M1 | Data | 3 | TNR for Write | - | - | M | T | M |
| M2 | Data | 3 | TNR for Read | - | - | M | H | M |
| M3 | Data | 1 | IPI-3 Burst | - | - | M | H | M |
| M4 | Completion | 3 | One Command Completion Response packet | - | - | L | T | M |
| M5 | Completion | 2 | One Asynchronous Response packet | - | - | F & L | H | M |
| M6 | Data | 1 | Last IPI-3 Burst when IORS | - | - | L | T | O |

Notes: 1. M1, M2 have Cat=5 for CP IPI-3
2. M4 has Cat=7 for CP IPI-3

# IPI-3/CP IPI-3
# INFORMATION UNITS BETWEEN PEERS

| IU Name | Operation Phase | Information_Set_1 | | Information_Set_2 | | FML | SI | M or O |
|---|---|---|---|---|---|---|---|---|
| | | Cat | Payload | Cat | Payload | | | |
| P1 | Send Data | 2 | IPI-3 Peer header | - | - | F | H | M |
| P2 | Send Data | 1 | One IPI-3 Burst | - | - | M | H | M |
| P3 | Send Data | 1 | Only or last IPI-3 Burst | - | - | L | T | M |
| P4 | Send Data | 2 | IPI-3 Peer header | 1 | First IPI-3 Burst | F | H | O |
| P5 | Request Data | 2 | IPI-3 Peer header | - | - | F | T | M |

# SCSI-FCP

- Defined by SCSI Committee (ANSI X3T10)
- SCSI phases mapped to IUs
- Useful for short-haul applications
- Classes 1 and 2 supported
- Device Headers not used
- Mappings for Initiator, Target defined
- SCSI I/O Process => One Exchange
- Uses Common FC-4 Functions
- FCP Services allow enable/disable:
  - Transfer Ready
  - Immediate Write

# SCSI-FCP
# COMMAND/RESPONSE FORMATS

## Command IU format

| Field | Size | Description |
|---|---|---|
| FCP_ENT_ADDR | 8 bytes | Entity Address (Common FC-4) |
| FCP_CNTL | 4 bytes | Control (Queue type, flags) |
| FCP_CDB | 16 bytes | Command Descriptor Block |
| FCP_DL | 4 bytes | Data Length |

## Status IU format

| Field | Size | Description |
|---|---|---|
| Reserved | 8 bytes | |
| FCP_STATUS | 4 bytes | Validity, status flags |
| FCP_RESID | 4 bytes | Residual count |
| FCP_SNS_LEN | 4 bytes | Length of sense data |
| FCP_RSP_LEN | 4 bytes | Length of response data |
| FCP_SNS_INFO | n bytes | Sense data |
| FCP_RSP_INFO | m bytes | Response data (vendor unique) |

# SCSI-FCP
# INFORMATION UNITS TO TARGET

| IU Name | Operation Phase | Information_Set_1 | | Information_Set_2 | | FML | SI | M or O |
|---|---|---|---|---|---|---|---|---|
| | | Cat | Payload | Cat | Payload | | | |
| T1 | Command | 6 | FCP_CMND | - | - | F | T | M |
| T2 | Command | 6 | FCP_CMND | - | - | F | H | O |
| T3 | Command (Linked) | 6 | FCP_CMND | | | M | T | O |
| T4 | Command (Linked) | 6 | FCP_CMND | | | M | H | O |
| T5 | Data | 1 | Only Data | - | - | M | T | M |
| T6 | Data | 1 | Only Data | - | - | M | H | O |
| T7 | Command /Data | 6 | One Write command | 1 | Only Data | F | T | O |
| T8 | Command /Data | 6 | One Write command | 1 | Only Data | F | H | O |
| T9 | Command /Data (Lkd) | 6/1 | FCP_CMND | 1 | FCP_DATA | M | T | O |
| T10 | Command /Data (Lkd) | 6/1 | FCP_CMND | 1 | FCP_DATA | M | H | O |

# SCSI-FCP
# INFORMATION UNITS TO INITIATOR

## Information Units to Initiator

| IU Name | Operation Phase | Information_Set_1 | | Information_Set_2 | | FML | SI | M or O |
|---|---|---|---|---|---|---|---|---|
| | | Cat | Payload | Cat | Payload | | | |
| I1 | Data | 5 | FCP_XFER_RDY | - | - | M | T | M |
| I2 | Data | 5 | FCP_XFER_RDY | - | - | M | H | M |
| I3 | Data | 1 | FCP_DATA | - | - | M | H | M |
| I4 | Status | 7 | FCP_RSP | - | - | L | T | M |
| I5 | Status | 7 | FCP_RSP | | | M | Y | O |
| I6 | Data + Rap | 1 | FCP_DATA | 1 | FCP_RSP | L | T | O |
| I7 | Data + Rsp (Linked) | 1 | FCP_DATA | 1 | FCP_RSP | M | T | O |

# SCSI-GPP

- Defined by SCSI Committee (ANSI X3T10)
- Suitable for long hauls or complex topologies
- Information packets mapped to IUs
- Information packet consists of:
  - Interface Control Prefix
  - 1-255 Interface Logical Elements (ILE)
- Device Headers not used
- Two long-lived Exchanges established:
  - One for transfer in each direction
  - Sequence Initiative never passed
- Common FC-4 Functions not applicable
- ILEs:

| | |
|---|---|
| Message | Command Descriptor Block |
| Logical Data | Command Parameter Data |
| Status | Command Response Data |
| Autosense | |

# SCSI-GPP
# INFORMATION UNIT

## Information Unit

| IU Name | Operation Phase | Information_Set_1 | | FML | SI | M or O |
|---|---|---|---|---|---|---|
| | | Cat | Payload | | | |
| I1/T1- | - | 2 | One Information packet | FML | H | M |

# SBCCS

- Defined by FC Working Group (ANSI X3T11) - FC-SB
- Maps command sets currently carried on OEMI, ESCON channels
- Commands, data, status mapped to IUs
- Classes 1 and 2 supported
- Device Headers not used
- Mappings for Channel, Control Unit defined
- SBCCS channel program => One Exchange
- Uses Common FC-4 Functions

# SBCCS
# INFORMATION UNITS TO CONTROL UNIT

Partial List

| IU Name | Operation Phase | Information_Set_1 | | FML | SI | M or O |
|---|---|---|---|---|---|---|
| | | Cat | Payload | | | |
| CMD_1 | Command | 6 | Flags, Command | F | T | M |
| CMD_n | Command | 3 | Flags, Command | M | T | M |
| ACR | Accept Command Response | 3 | Parameters | M | T,H | M |
| ST_ACC | Accept Status | 3 | Parameters | L | H | M |
| CANCEL | Cancel operation | 6 | Parameters | F,M | T | M |
| RQS | Request Status | 6 | Parameters | F,M | T | M |
| SEL_RST | Selective Reset | 6 | Parameters | F,M | T | M |
| ST_STK | Stack Status | 3 | Parameters | L | H | M |
| SYS_RST | System Reset | 6 | Parameters | F,M | T | M |
| DATA_1 | First Data of multiple data IUs | 1 | Data | M | H | M |
| DATA_LW | Last Write Data of multiple data IUs | 1 | Data | M | T | M |
| DATA_n | Data of multiple data IUs | 1 | Data | M | H | M |

# SBCCS
# INFORMATION UNITS TO CHANNEL

| IU Name | Operation Phase | Information_Set_1 | | FML | SI | M or O |
|---|---|---|---|---|---|---|
| | | Cat | Payload | | | |
| CMR | Command Response | 5 | Data Descriptor | M | T,H | M |
| DLE | Device Level Exception | 3 | Parameters | L | T | M |
| DATA_1 | First Data of multiple data IUs | 1 | Data | M | H | M |
| DATA_n | Data of multiple data IUs | 1 | Data | M | H | M |
| STATUS | Status | 7 | Flags, status, count, supplemental status | M | T | M |
| DL_ACK | Device Level ACK | 3 | Parameters | L | T,H | M |

# HIPPI-FP

- Defined by FC Working Group (ANSI X3T11) - FC-FP
- HIPPI-FP Packets mapped to IUs
- Classes 1 and 2 supported
- Device Headers not used
- HIPPI Connection => One Exchange
- Common FC-4 Functions not applicable

# HIPPI
# INFORMATION UNITS

## Information Units

| IU Name | Operation Phase | Information_Set_1 | | Information_Set_2 | | F M L | S I | M or O |
|---|---|---|---|---|---|---|---|---|
| | | Cat | Payload | Cat | Payload | | | |
| H1 | - | 2 | HIPPI-FP header and D1 | 4 | HIPPI-FP D2 | F,M L | H | O |
| H2 | - | 2 | HIPPI-FP header and D1 | - | - | F,M L | H | M |
| H3 | - | 4 | HIPPI-FP D2 | - | - | M,L | H | M |

# LINK ENCAPSULATION

- Full 802.2 mapping defined by FC Working Group (ANSI X3T11) - FC-LE
  - Under development
- "IP/ARP on FC" defined by IETF
  - IP/ARP Packets mapped to IUs
  - With IEEE 802.2 LLC/SNAP headers (set to fixed values)
  - Classes 1 and 2 supported
  - Device Headers not used
  - One or more IP/ARP packets => One Exchange
  - Common FC-4 Functions not applicable
  - MTU = 65280 octets (64K + 256)
  - ARP Message format defined
  - Draft RFC available from Internet-Draft directory
    - draft-rekhter-fibre-channel-01.txt

# LINK ENCAPSULATION INFORMATION UNITS

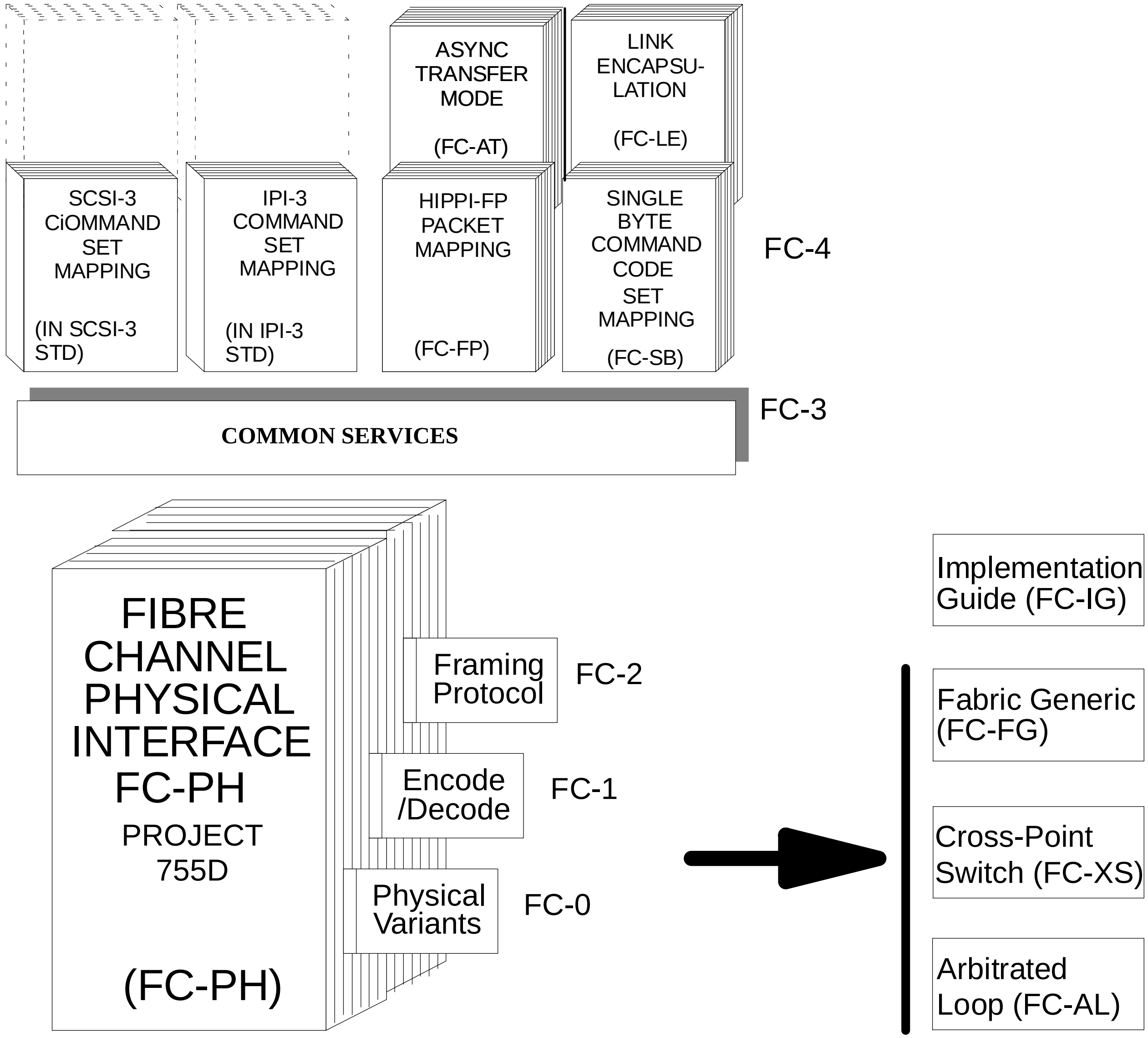
## Information Units

| IU Name | Operation Phase | Information_Set_1 | | FML | SI | M or O |
|---|---|---|---|---|---|---|
| | | Cat | Payload | | | |
| IP1 | | 4 | IEEE 802.2 LLC header<br>SNAP header<br>IP Packet | F, M, L | H | M |
| ARP1 | | 4 | IEEE 802.2 LLC header<br>SNAP header<br>ARP Packet | F, M, L | H | M |

# TOPOLOGIES

- Definitions and Requirements (FC-FG)
- Point-to-Point Topology (FC-PH)
- Cross-Point Switch Topology (FC-XS)
- Arbitrated Loop (FC-AL)

# FIBRE CHANNEL LEVELS
# TOPOLOGIES



# WHAT IS A FIBER CHANNEL TOPOLOGY?

A "SCHEME"

WHICH

INTERCONNECTS

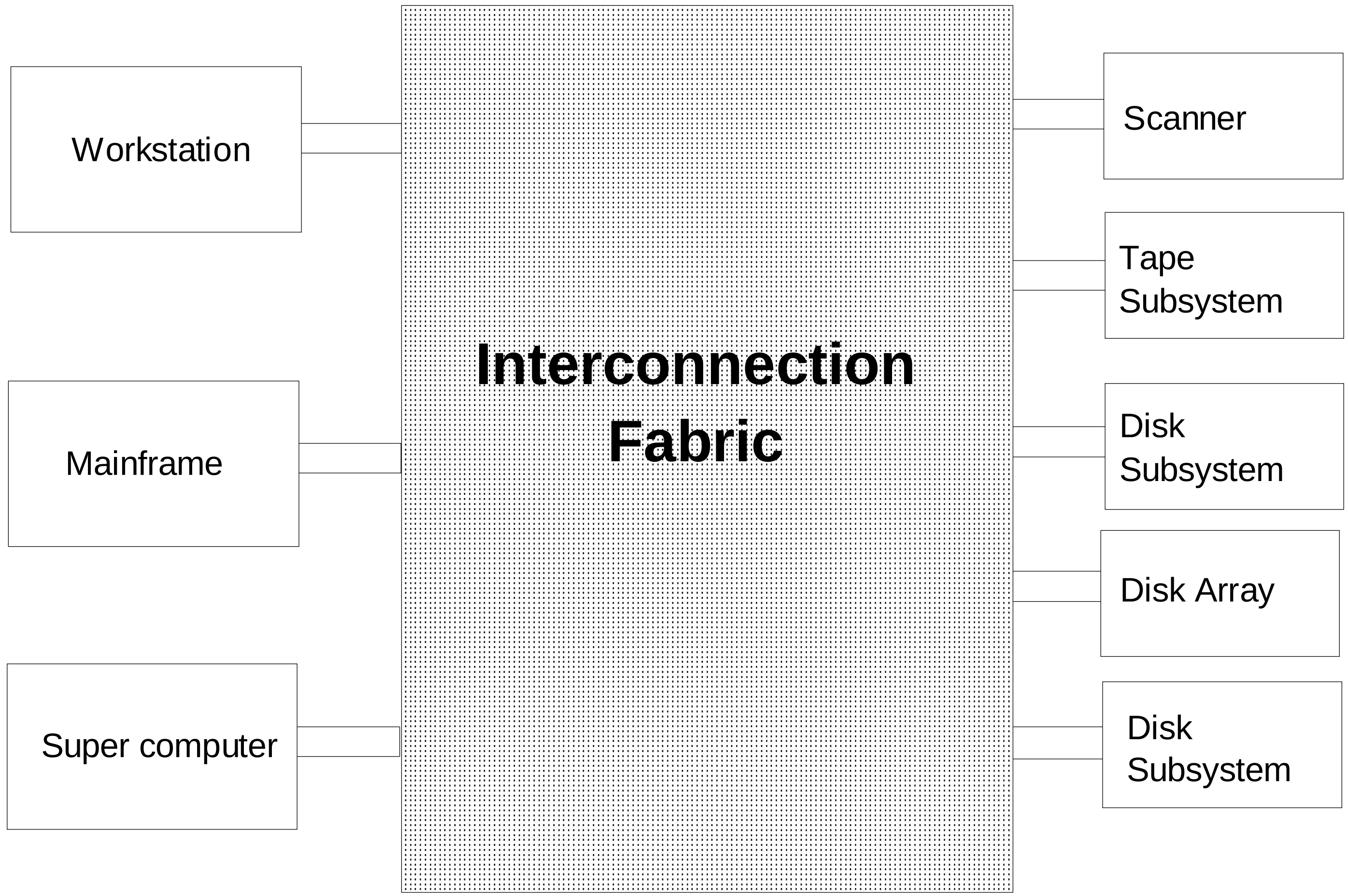
MULTIPLE

FIBRE CHANNEL INTERFACES

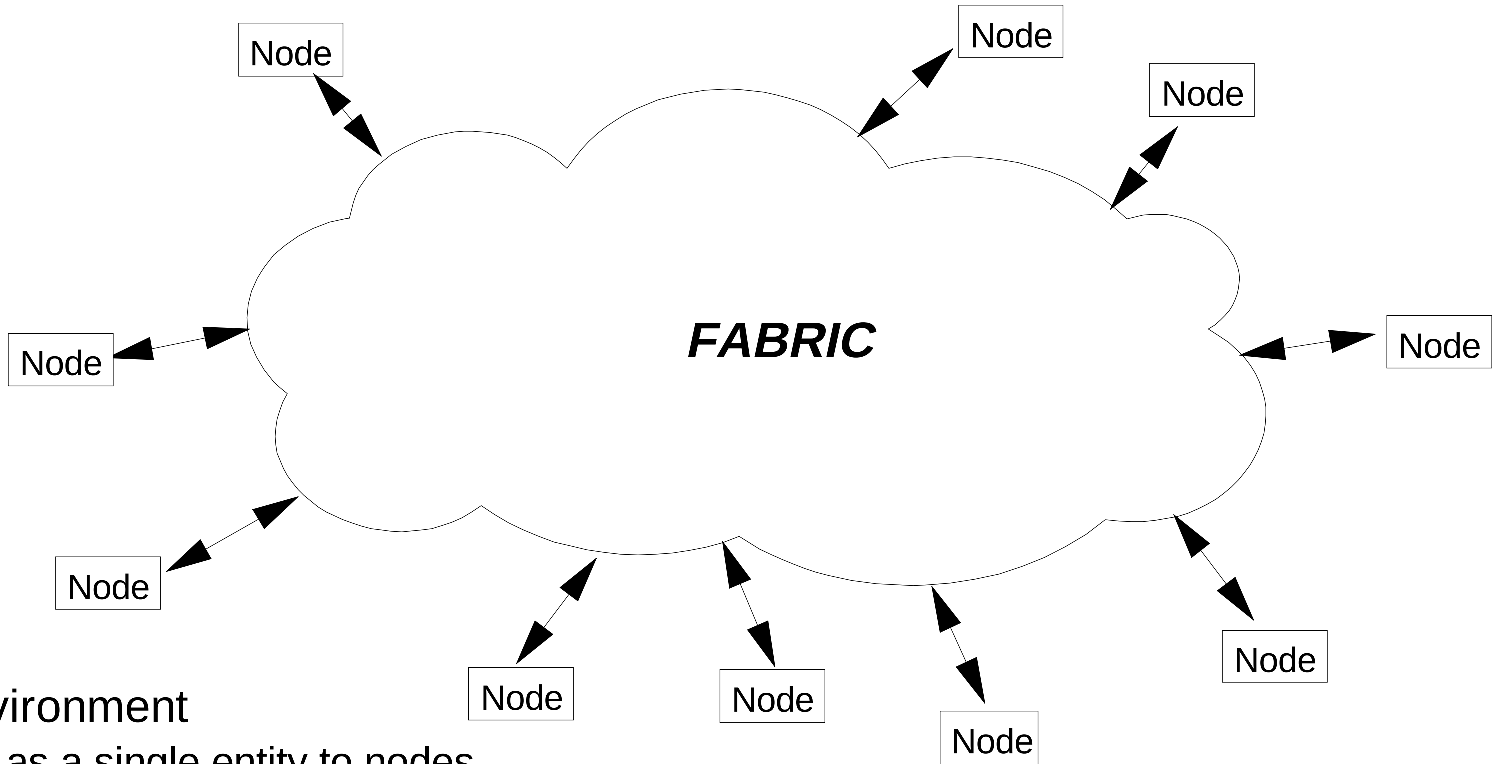
# FIBRE CHANNEL TOPOLOGIES

- Point-to-Point
  - Exactly two N_Ports connected together
  - No Fabric Elements present
  - No "Fabric" Services available
- Cross-Point Switch topology
  - May use entire 24-bit address space
  - Distributed Tree variant
- Arbitrated Loop topology
  - Private Loop up to 126 NL_Ports
  - May attach to a Cross-Point Switch topology through an FL_Port

# CROSS-POINT SWITCH TOPOLOGY BASICS



# CROSS-POINT SWITCH TOPOLOGY
# GENERIC MODEL




- Fabric Environment
  - Appears as a single entity to nodes
  - Consists of one or more fabric elements
  - Provides a homogeneous addressing space
- Transparency to nodes
  - Fabric topology
  - Routing path selection
  - Internal structure

# CROSS-POINT SWITCH TOPOLOGY
# GENERIC MODEL

- FC-PH does not address:
  - Inter-operation of heterogeneous fabric Elements
  - ID space partitioning
  - Rationalization of previously-separate ID spaces
- ANSI Project Proposals now in approval process to address these subjects:
  - FC-FG generic fabric requirements
  - FC-XS cross-point switch fabric requirements
  - FC-AL Arbitrated Loop topology requirements

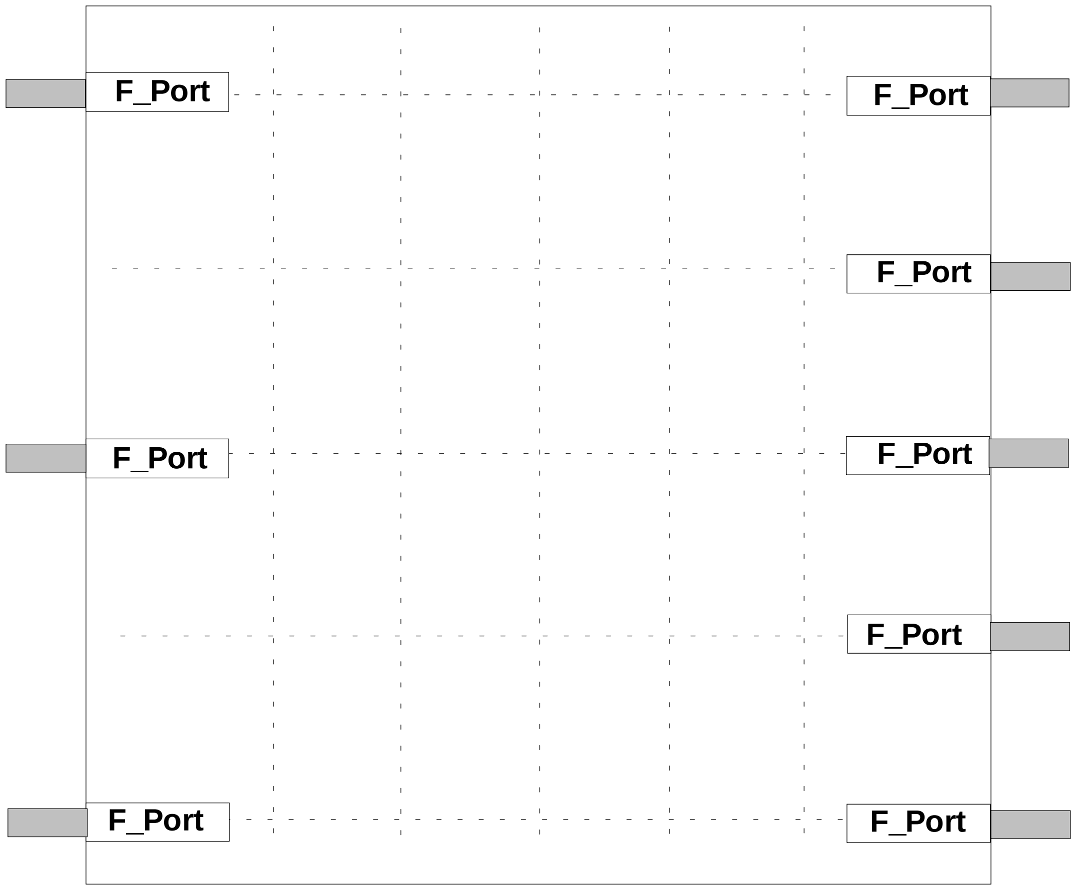
# CROSS-POINT SWITCH TOPOLOGY IMPLEMENTATIONS

- Many methods of designing and building Fibre Channel Fabrics:
  - Topology influenced in a major way by system performance requirements
  - Implementation a major function of packaging requirements and required growth capability

- When combined with choices of physical variants, offers unheard-of capability for product differentiation and system tuning to specific applications

- **KEY POINT IS THAT ONE "NODE DESIGN" IS COMPATIBLE WITH ALL FABRIC IMPLEMENTATIONS**

# CROSS-POINT SWITCH TOPOLOGY
# SWITCH STRUCTURE



# CROSS-POINT SWITCH TOPOLOGY ADVANTAGES

- Familiar type of operation from cross-point telephone exchanges
- Definitely the highest performance topology
  - Non-blocking architecture
- Choice of multiple paths between pairs of F_Ports
  - Moderate speed paths provide huge system bandwidths
  - e.g., 8x8 switch with 25 MB/s paths gives 200 MB/s system bandwidth (400 MB/s aggregate - 200 MB/s each direction)
- Topology able to support all classes of operation:
  - Class 1: dedicated connection and reliable
  - Class 2: "buffered" multiplexed and reliable
  - Class 3: "buffered" datagram

# CROSS-POINT SWITCH TOPOLOGY SWITCHING MODES

- Connection set up modes:
  - Circuit switched  (dedicated connection) mode
  - Frame switched (connectionless) mode

- Circuit switching
  - Pre-allocation of transmission bandwidth
  - Dedicated path connection on end-to-end basis
  - Information transmission until path is cleared

- Frame switching
  - Dynamic allocation of bandwidth on a link-by-link basis
  - Information transmission in blocks of limited size - 'frames'
  - Adaptive routing of individual frames
  - Buffering required for flow control

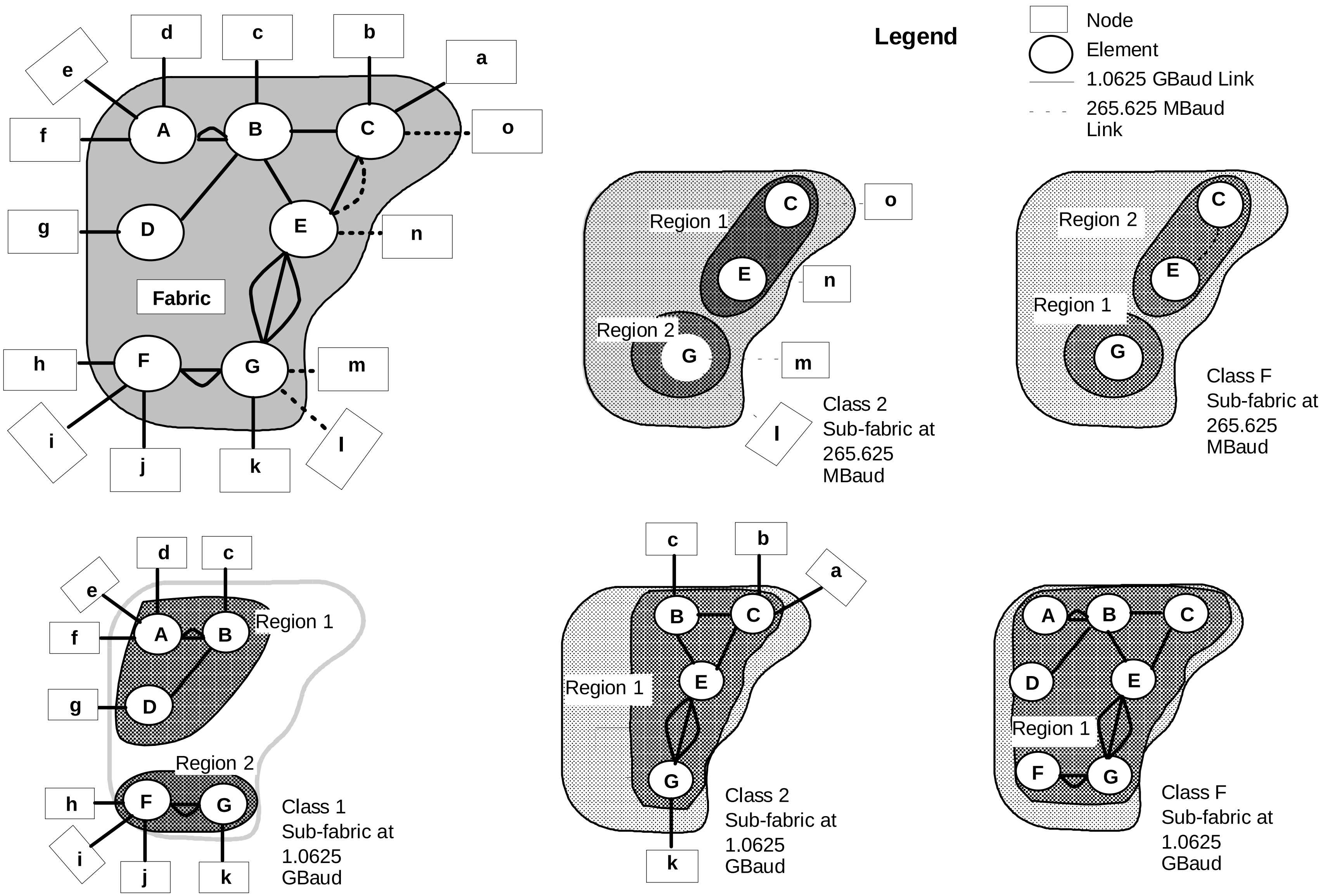
# CROSS-POINT SWITCH TOPOLOGY
# CIRCUIT SWITCHING OPTIONS

- Selection criteria for switching mode
  - Message length
  - Frame switching: queue delay; retransmission; reassembly
  - Circuit switching:  message transmission delay; circuit set up time
- Synchronous switching
  - Maintains bit and word alignment when path switching is performed
- Asynchronous switching
  - Bit and word alignment are lost any time a path change is made

# CROSS-POINT SWITCH TOPOLOGY CONCEPTS

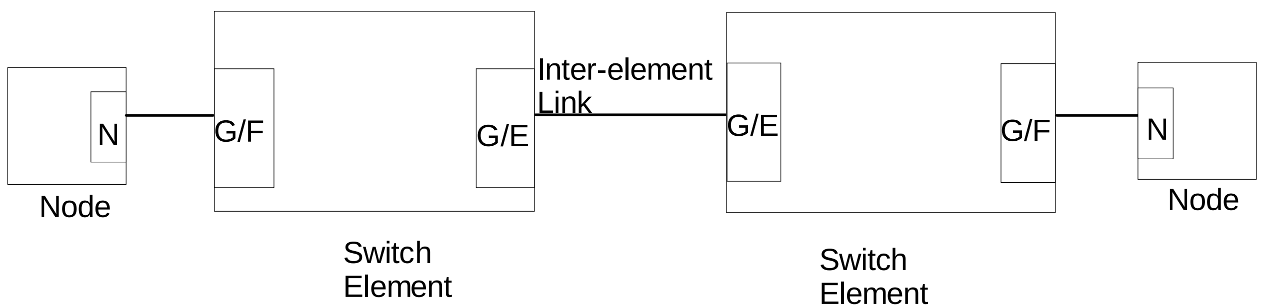
- XS fabric is a transport medium for fibre channel frames
- May comprise one or more switch elements connected via inter-element links
- May support all classes of services (1, 2, 3)
- Class F service to coordinate internal behavior of the fabric
- Within the fabric, each combination of a class of service and a data rate constitutes a logical 'sub-fabric'
- Two or more 'sub-fabrics' exist within the fabric
- One or more 'regions' exist within a 'sub-fabric'
- Homogeneous service and data rates within a 'region' are guaranteed
- Regions may also be created for topology management reasons
- Communication between different 'regions' require bridging through the intervening 'sub-fabrics'

# CROSS-POINT SWITCH TOPOLOGY PARTITIONING EXAMPLE



# CROSS-POINT SWITCH TOPOLOGY
# FABRIC PORTS

- **E_Port:**
  - An element port which is used to interconnect with another element
  - Shall carry both class F frame and class 1, 2 or 3 frames
- **F_Port:**
  - An element port which is used to interconnect with the N_Port of a node
  - Shall carry only class 1, 2 or 3 frames
- **G_Port:**
  - An element port that discovers whether to operate as an E_Port or an F_Port during login
  - If fabric login (LOGI) received, the port behaves as an F_Port
  - In response to the inter-element login (IELOGI) if "fabric presence" response is received, the port behaves as an E_Port;
  - Also, if an 'IELOGI' is received, the port behaves as an E_Port
  - May be implicitly assigned to behave as an E_Port or an F_Port



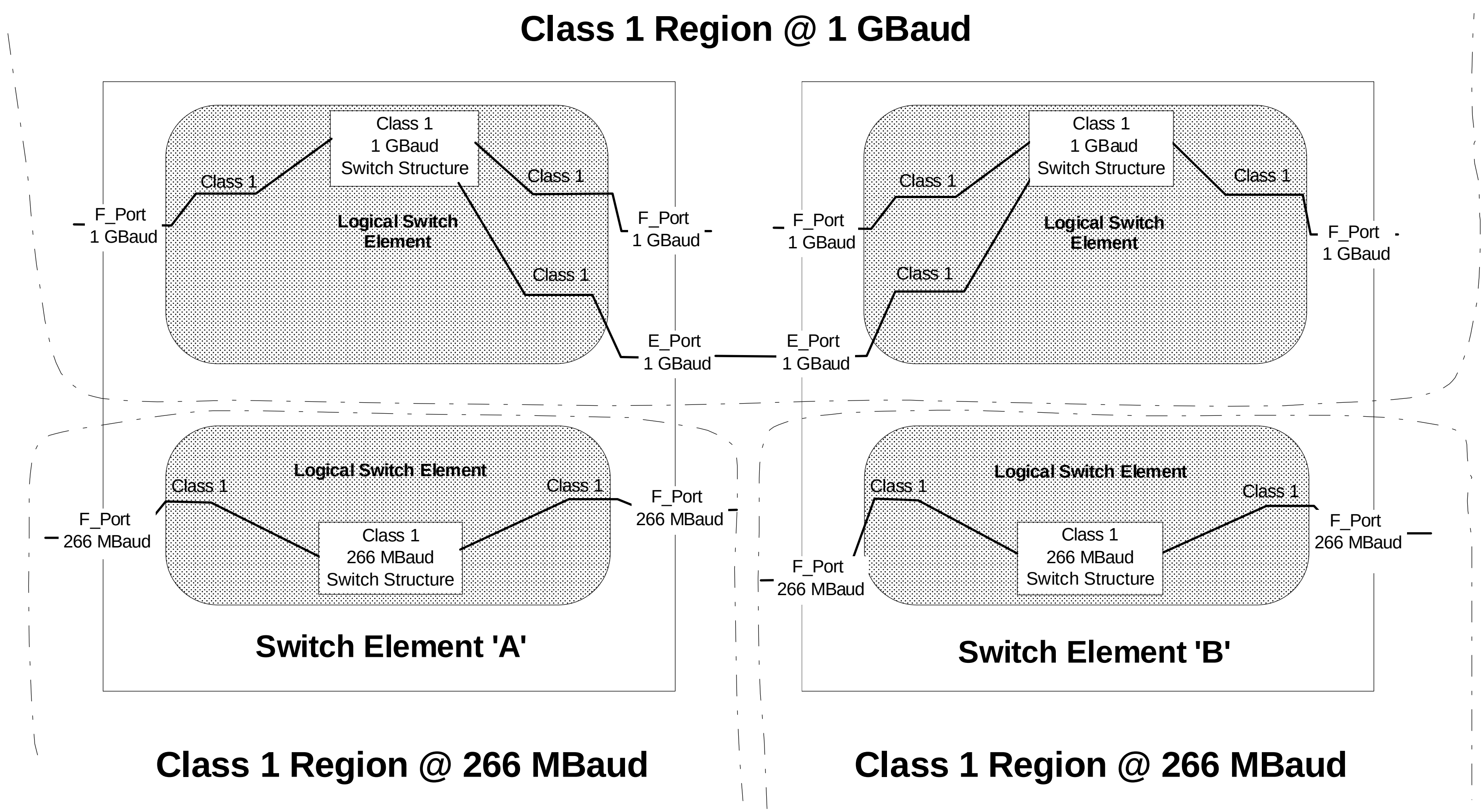
# CROSS-POINT SWITCH TOPOLOGY
# SWITCH ELEMENT

- XS fabric is made up of one or more physical switch elements
- Physical switch element has three or more ports.  Any mixture of E_Ports, F_Ports and G_Ports may be used
- Physical switch element supports multiple concurrent paths between different port pairs
- Each physical switch element has an element controller
- Global configuration parameters are maintained by the element controller
- Physical switch element consists of two or more logical switch elements

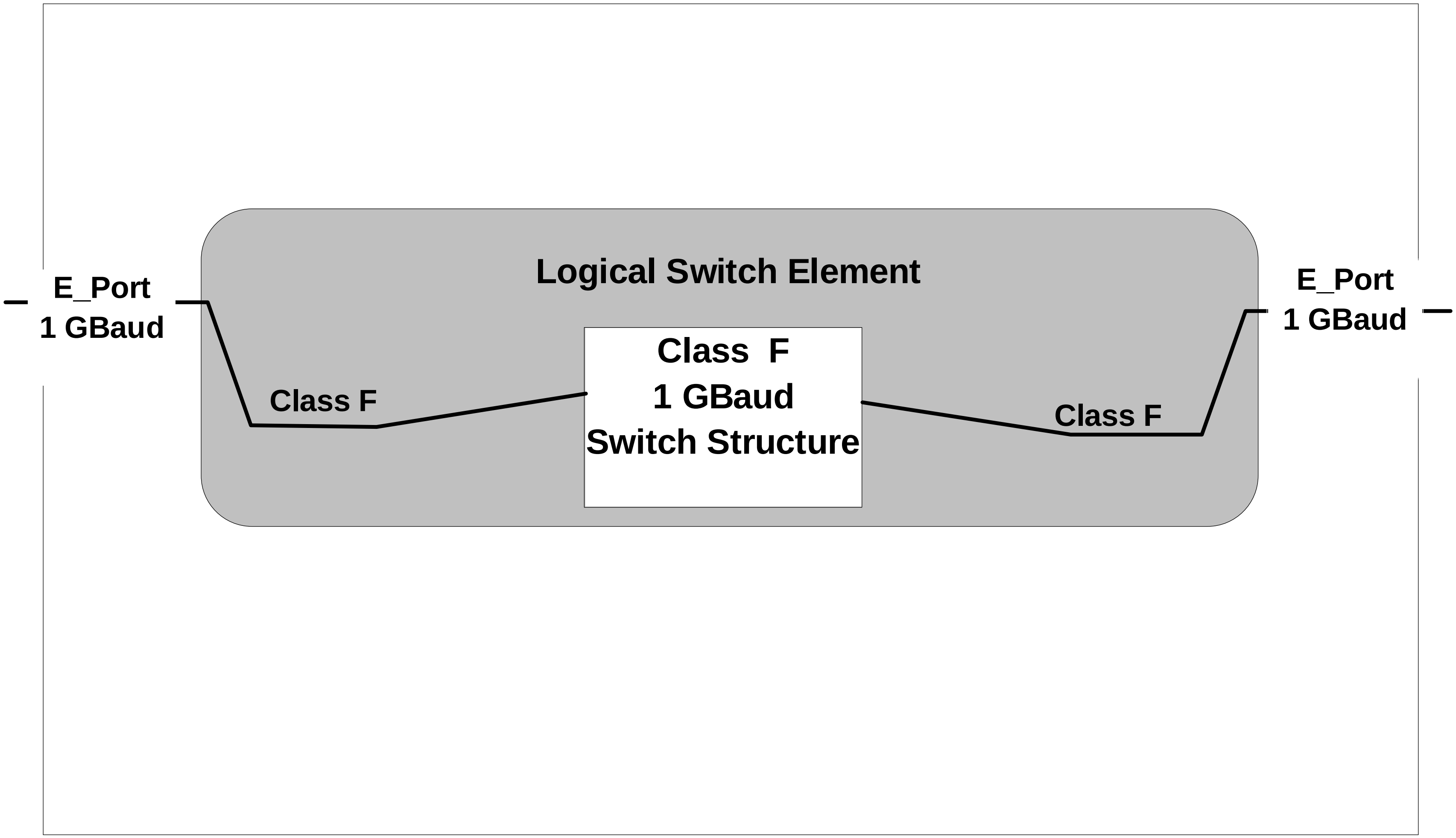
# CROSS-POINT SWITCH TOPOLOGY
# SWITCH ELEMENT (cont'd)

- Each combination of a class of service and data rate constitutes a logical switch element
- Logical switch element consists of indistinguishable ports joined internally by a switching structure
- Internal switching structure is invisible to the N_Ports
- Routing frames within the fabric transparent to the N_Ports
- Switch elements can be interconnected in an unrestricted topology via compatible media interface
- Switch element may optionally serve as 'root' element for distributed fabric
- Switch element may optionally serve as 'control' element for loop fabric

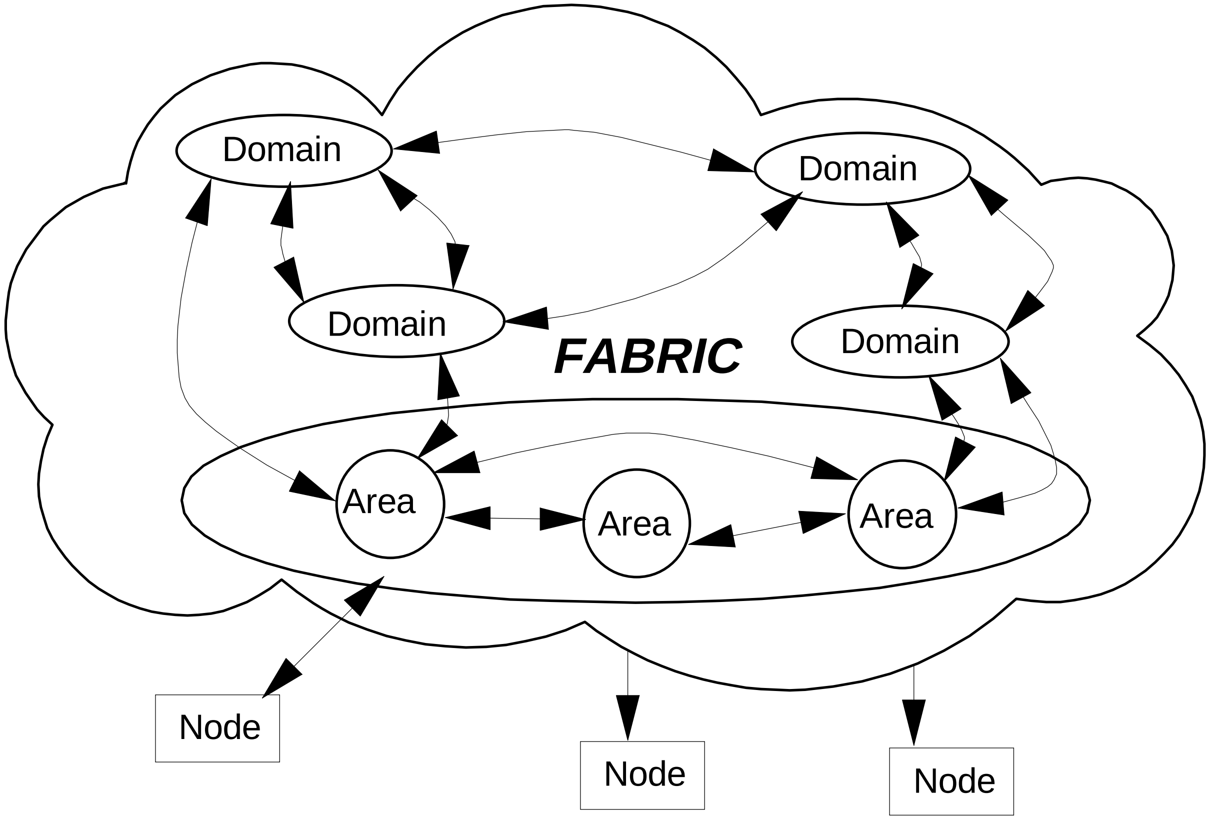
# CROSS-POINT SWITCH TOPOLOGY
# SWITCH ELEMENT EXAMPLE



# CROSS-POINT SWITCH TOPOLOGY
# SWITCH ELEMENT EXAMPLE



# CROSS-POINT SWITCH TOPOLOGY FABRIC ADDRESS PARTITIONING




- The address partitioning breaks the 24 bit address identifier space into a three level hierarchy
- Domains are the highest logical level constructs in the addressing hierarchy
- Areas are the middle logical level constructs in the addressing hierarchy
- Ports form the lowest logical level constructs in the addressing hierarchy
- Fabric elements acquire a contiguous range of address identifiers that may span multiple areas

# CROSS-POINT SWITCH TOPOLOGY
# FABRIC ADDRESS PARTITIONING (cont'd)

- Address partitioning is made to aid routing by limiting resource requirements
- The address partitioning is invisible to the N_Ports
- Fabric elements are not required to recognize the address partitioning
- Space reserved for future use
- Vendor unique address partitions for special requirements not defined in the standard
- Recognized address types are:
  - FC-PH and FC-EP assigned addresses
  - Address assignments which require no special attention for the purpose of routing
  - Address which may require special attention to perform routing
  - Addresses acquired for XS fabric management

# CROSS-POINT SWITCH TOPOLOGY FABRIC ADDRESS PARTITIONING (cont'd)

- Port identifier partition
  - E_Port identifiers (Inter-element link ports)
  - N_Port (F_Port) identifiers
    - Native address identifiers
    - Simple alias address identifiers
    - Element-wide hunt groups
- Special Identifier Partition
  - Reserved for fabric assisted routing function
    - Hunt Groups (restricted to a single domain)
    - Multicast (restricted to a single domain)
  - Domains not requiring any fabric-assisted routing functions may use the special identifier partition as a port identifier partition

# CROSS-POINT SWITCH TOPOLOGY ADDRESS ASSIGNMENT

- Each switch element assigned a contiguous address pool
- The address pool size is variable
- World wide name may be attached to each address within pool
- All operating F_Ports assigned addresses from the pool
- E_Ports connected to other E_Ports are also assigned addresses from the same pool
- The N_Ports inherit the addresses of the F_Ports that they are connected to during 'login'

# CROSS-POINT SWITCH TOPOLOGY ELEMENT CONTROLLER

- Each switch element is associated with an element controller
- Logically responsible for accepting and generating frames destined to or expected from the switch element
- Logically accepts, as an alias, well-known fabric F_Port address, fabric controller address and any unused address identifier managed by the element
- Optionally can have a unique N_Port address ID; normal N_Port protocol applies; may support classes 1, 2, or 3
- Class F frames destined for a port maintained by the element always forwarded to the element controller
- Regional configuration parameters are acquired, retained or re-acquired by the element controller

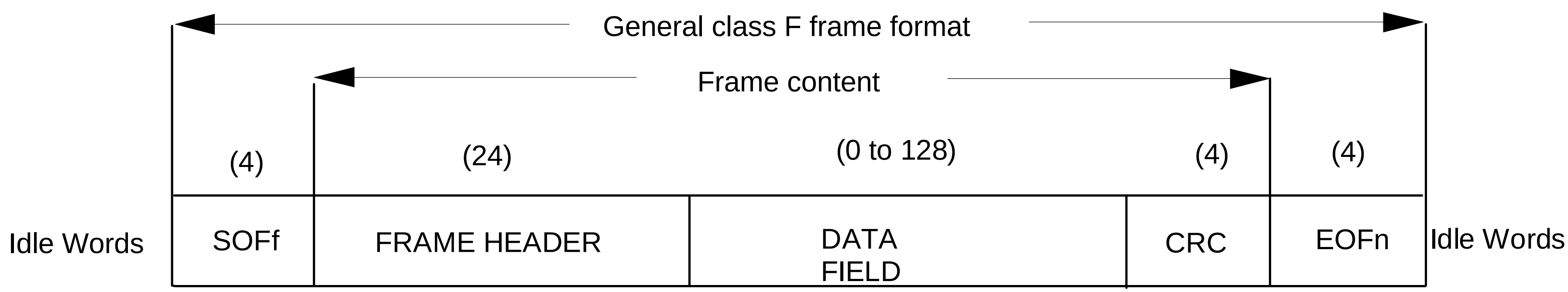
# CROSS-POINT SWITCH TOPOLOGY
# CLASS F SERVICE

- Provides connectionless service
- Controls and coordinates internal behavior of the fabric
- Usage fabric-wide across all regions and sub-fabrics
- Not applicable outside of the fabric
- Based on a simplified class 2 model
- Supports end-to-end (element controller-to-element controller) flow control using ACK_1 frame only
- Supports buffer-to-buffer (E-Port-to-E_Port) flow control using R_RDYs
- Supports use of sequences and exchanges
- Class F frames are used on inter-element link (IEL) to perform IEL services :
  - E_Port-to-E_Port login
  - E_Port-to-E_Port logout
  - Other IEL services

# CROSS-POINT SWITCH TOPOLOGY
# CLASS F SERVICE (Cont'd)

- Allows in-order and out of order delivery of class F frames
- All class F frames start with an SOFf delimiter
- All class F frames are terminated by an EOFn delimiter
- Class F frames use the standard FC-PH frame header
- Class F frames do not use optional headers
- Class F frame data field size is fixed at 128 bytes
- Class F frames may always be intermixed onto class 1 dedicated connection
- Class F frames destined to F_Ports maintained by an element are logically forwarded to the element controller
- Fabric guarantees notification of non-delivery of class F frames to the originator

# CROSS-POINT SWITCH TOPOLOGY
# CLASS F FRAME FORMAT



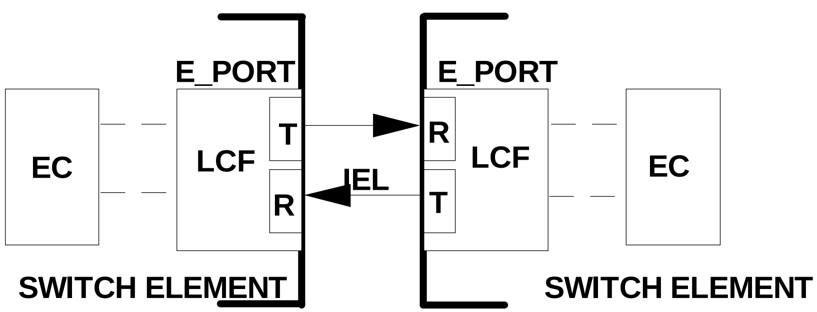
# CROSS-POINT SWITCH TOPOLOGY
# INTER-ELEMENT LINK (IEL)

- Represents link between E_Ports belonging to two adjacent elements

- May support FC-0 and FC-1 compliant media, interface and transmission protocols as per FC-PH; and other means also possible

- IEL frame traffic:
  - Class F frames
  - Class 1, 2 or 3 frames

- IEL flow control:
  - Buffer to buffer flow control between E_Ports using R_RDY primitive; similar to N_Port to F_Port
  - End-to-end flow control between the element controllers logically connected to the communicating E_Ports across one or more IEL's using ACK_1 frame; similar to N_Port to N_Port




**LEGEND**
EC: Element Controller
LCF: Link Control Facility
T: Transmitter
R: Receiver

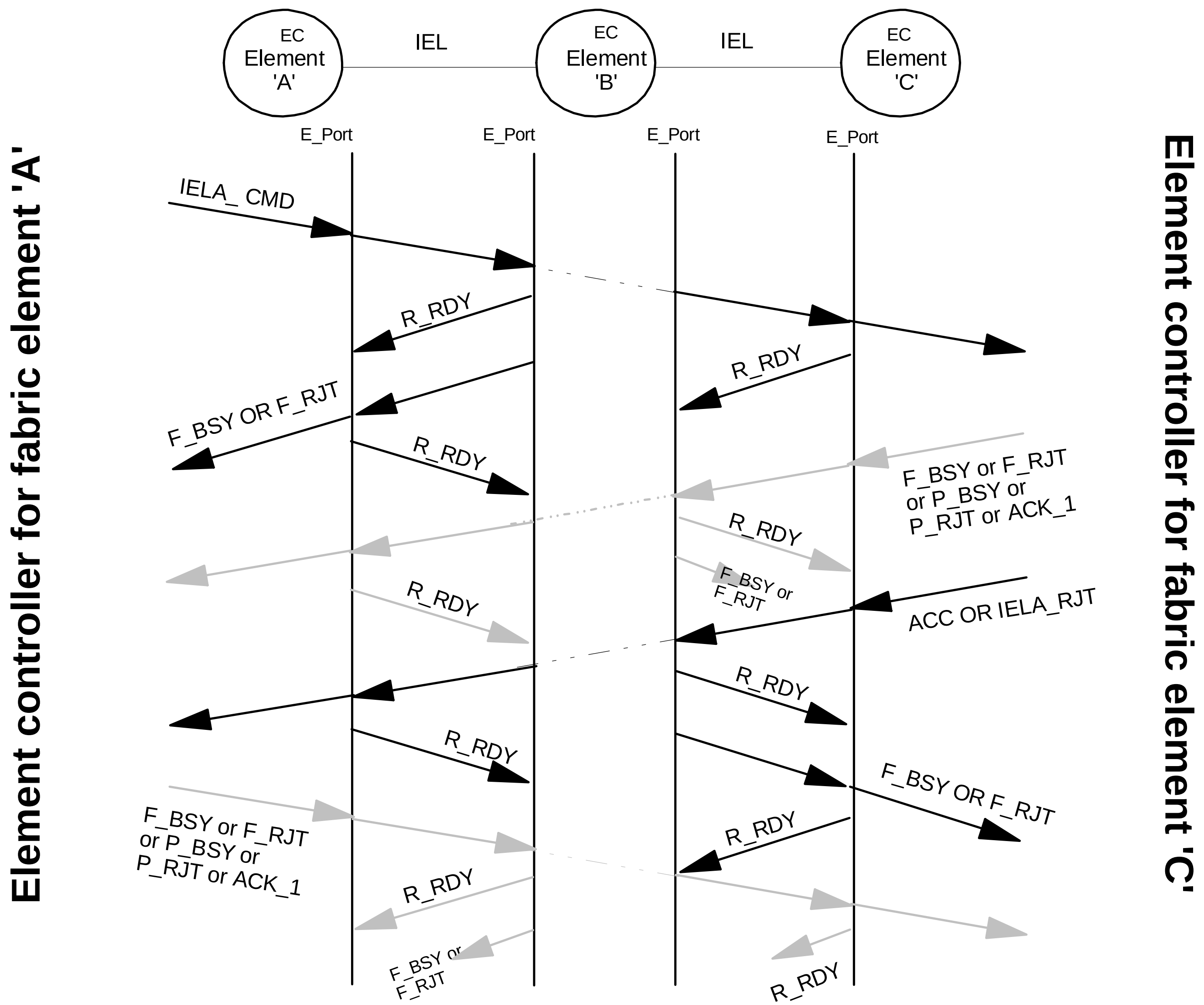
# CROSS-POINT SWITCH TOPOLOGY INTER-ELEMENT LINK (IEL) SERVICES PROTOCOL

- IEL services protocol interoperable between fabric element over inter-element links (IEL)
- Frame/Sequence exchange between element controllers (EC) via fabric E_Ports
- Classes of service supported
  - Class F:  mandatory; class F frame
  - Classes 1, 2, 3: optional;  FC-PH frame
- Type of frame:  link data

# CROSS-POINT SWITCH TOPOLOGY
# IEL SERVICES PROTOCOL (cont'd)

- IEL services to be defined
  - IEL_Login
  - IEL_Logout
  - IELA_RJT
  - others (TBD)
- Flow control: based on FC-PH model
  - R_RDY: buffer to buffer
  - ACK_1: end to end
- Link response rules:  based on FC-PH model

# CROSS-POINT SWITCH TOPOLOGY
# IELA FLOW CONTROL MODEL



# CROSS-POINT SWITCH TOPOLOGY SWITCH CONSIDERATIONS

- Data rates:
  - N_Ports with dissimilar data rates:  advantageous to support mixed rate environments (class 2 and class 3 only)
- Media choices:
  - N_Ports with dissimilar media: advantageous to support mixed media environments
- Services classes:
  - N_Ports supporting one or more service classes:  advantageous to support multiple service classes as well as mixed class environments
- Size:
  - Installation differing in the number of ports required:  advantageous to make a switch design supporting incremental growth
- Configuration control:
  - Fabric administrative function to support
  - Utilities to control system variables
  - Initial F_Port configuration
  - F_Port privileges
  - Initial routing information
  - Alternate routing information

# GENERAL FABRIC SWITCH CONSIDERATIONS (cont'd)

- Performance monitoring
  - Collection of utilization data allows efficient switch management
- Error logging
  - Collection of error information allows efficient switch management
- Common services:
  - Inclusion of common server functions:  name, time, management server and fabric controller within fabric
  - Larger installations may augment or replace the internal servers with servers located outside the fabric
  - Login service is mandatory and must be present in the switch
- Reliability:
  - Redundant designs of both switches and cascading links for desired level of availability

# CROSS-POINT SWITCH TOPOLOGY ADVANTAGES

- Topology:
  - Allows performance and bandwidth to be tailored to specific requirements
- Switching flexibility:
  - Frame switching for time sharing systems, where average message length is on order of 1000 bytes
  - Circuit switching for file transfers, where messages are several thousand bytes long or more and should be sent at the same time
- No inherent limitations to the performance:
  - An unlimited number of data paths can be provided
- Varying degrees of fault tolerance:
  - Combination of redundant links, switches and the provision for alternate paths in the fabric
- Current VLSI and packaging technologies:
  - Makes the fabric switch a cost effective solution in the high speed interconnection market

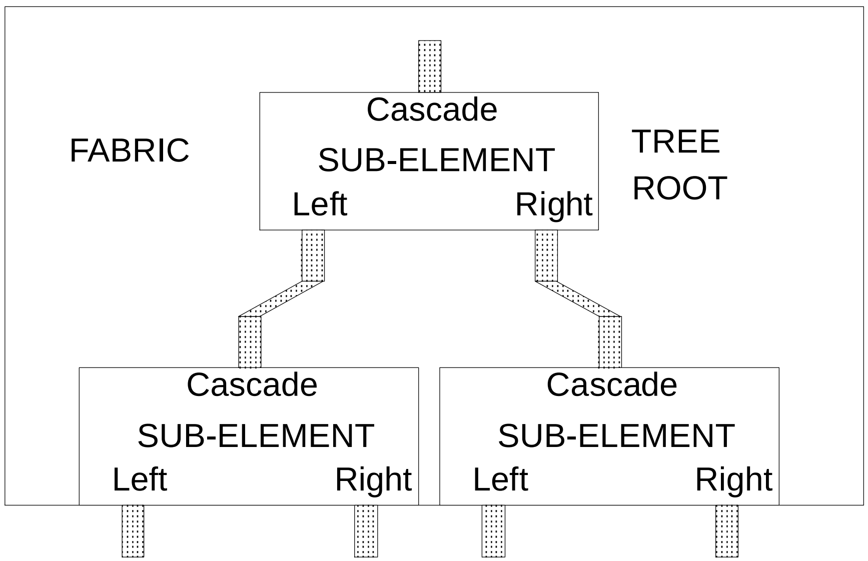
# CROSS-POINT SWITCH TOPOLOGY IMPLEMENTATION EXAMPLE

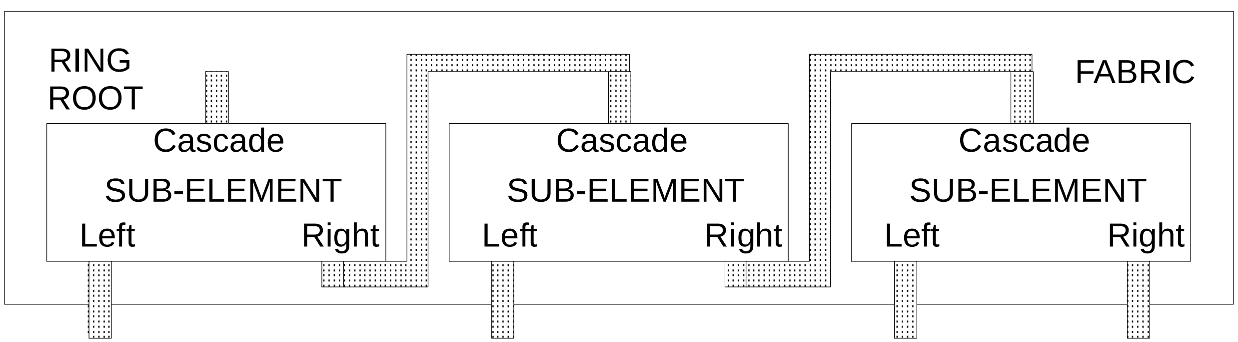
# DISTRIBUTED FABRIC

- Variant of general topology that functions as a tree structure
- Cross-point Switch topology is generally considered a mesh
- This simplification of structure permits less expensive switches
- Lacks the redundancy of the general topology

# CROSS-POINT SWITCH TOPOLOGY
# DISTRIBUTED FABRIC CONNECTIONS

Cabling of ports dictates actual topology



# DISTRIBUTED FABRIC SUB-ELEMENT

- Requires one Frame Buffer per port and small amount of Frame routing logic
- Power-failure resilience is an issue
  - External Sub-Elements with separate power supplies make for a resilient system
  - Sub-Elements are lower cost if left port is integrated with the Node but less resilient

# CROSS-POINT SWITCH TOPOLOGY
# DISTRIBUTED FABRIC ADVANTAGES

- Flexible topology and connectivity
- Fully-compliant with FC-PH F_Port definition
- Single configuration can have mix of different media and rates
- Easily expandable
- Interoperability of Sub-Elements from different sources easy to define

# CROSS-POINT SWITCH TOPOLOGY
# DISTRIBUTED FABRIC ARCHITECTURE

- Composed of multiple identical Sub-Elements
- Each Sub-Element has three F_Ports:
  - Cascade, Left and Right
  - Each fully compliant with FC-PH
- Sub-Elements expected to cost more than L_Ports
- Definition is a mouthful:
  - Mesh topology (generalized)
  - Blocking switch
  - Non-network (only one route between any pair of F_Ports)
- Within restrictions of definition, several topologies possible

# CROSS-POINT SWITCH TOPOLOGY DISTRIBUTED FABRIC PRINCIPLES

- Only one possible route between any pair of Sub-Elements:
  - Sequential frame order guaranteed
- ID assignment performed at each power-on:
  - Every Sub-Element tests for attachment to Cascade port (if no attachment then Root)
  - ID assignment begins with broadcast initiated by the Root:
    - As broadcast migrates IDs are assigned
    - A particular F_Ports does not necessarily get the same ID at every power-on
- If cabling rules not followed, end up with independent Fabrics:
  - Not a failure and helps diagnosis
- Limitation of one Frame buffer per F_Port can cause problems:
  - One lost R_RDY causes loss of credit
  - Link Reset recovers successfully
  - Not a major issue

# FABRIC TOPOLOGY SUMMARY

- Fiber Channel Fabric is a significant tool for the Systems Architect:
  - Tailor system performance with Fabric topology
  - Tailor system cost/performance with Fabric implementation and FC-PH Physical Variant definitions
  - Tailor expandability with Fabric architectures, etc.
- Fiber Channel Fabric definition allows many different topologies, architectures, implementations
- All WITHOUT requiring hardware or software changes to computers, controllers or peripherals!!
- Present Fabric definition is just the beginning!
  - Generalized heterogeneous Fabric operation will take significant work

# FIBRE CHANNEL SERVICES

- Services for reliable transfer of data
- Well-known addresses for the servers
  - Login Server: FFFFFE
  - Fabric Controller: FFFFFD
  - Name Server: FFFFFC
  - Time Server: FFFFFB
  - Management Server: FFFFFA
- Logical N_Ports
- All services share common protocol
- Support of all classes of service
- All services not mandatory

# FIBRE CHANNEL SERVICES
# SERVER FUNCTIONS

- Login Server
  - Helps to discover operating characteristics
  - Provides N_Port address assignment
  - Mandatory  within fabric
- Controller
  - Assists initialization / configuration
  - Routing management
  - Optional fabric-assisted services
  - Mandatory within fabric

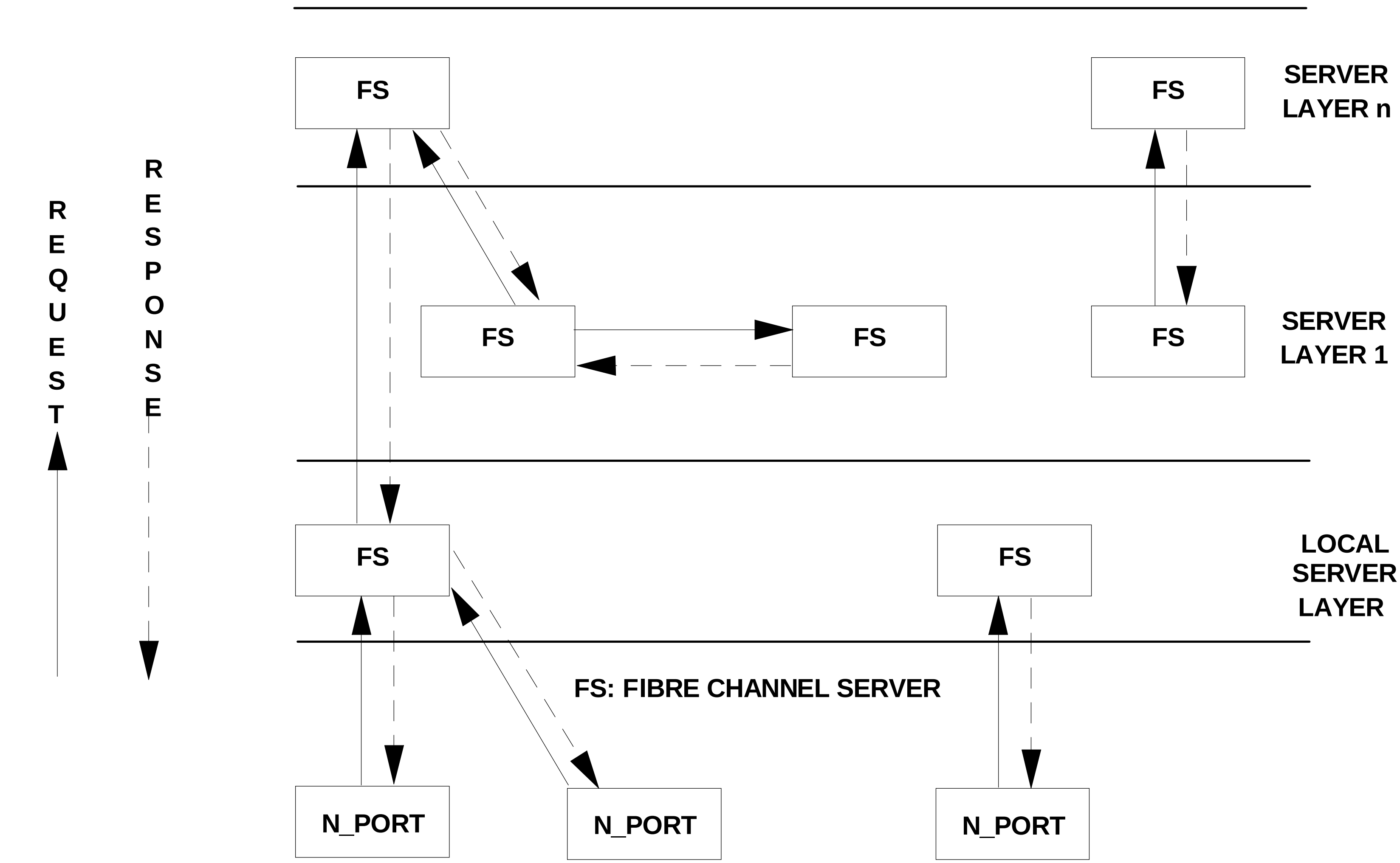
# FIBRE CHANNEL SERVICES SERVER FUNCTIONS

- Name server (optional)
  - Translation for N_Port ID's
    - IEEE/CCITT addresses (worldwide unique)
    - IP addresses
    - Symbolic names
  - Vital product data
- Time server (optional)
  - Management of expiration timers
  - Time synchronization
  - Time of Day
- Management server (optional)
  - Configuration management
  - Performance management
  - Fault management
  - User diagnostics
  - Accounting

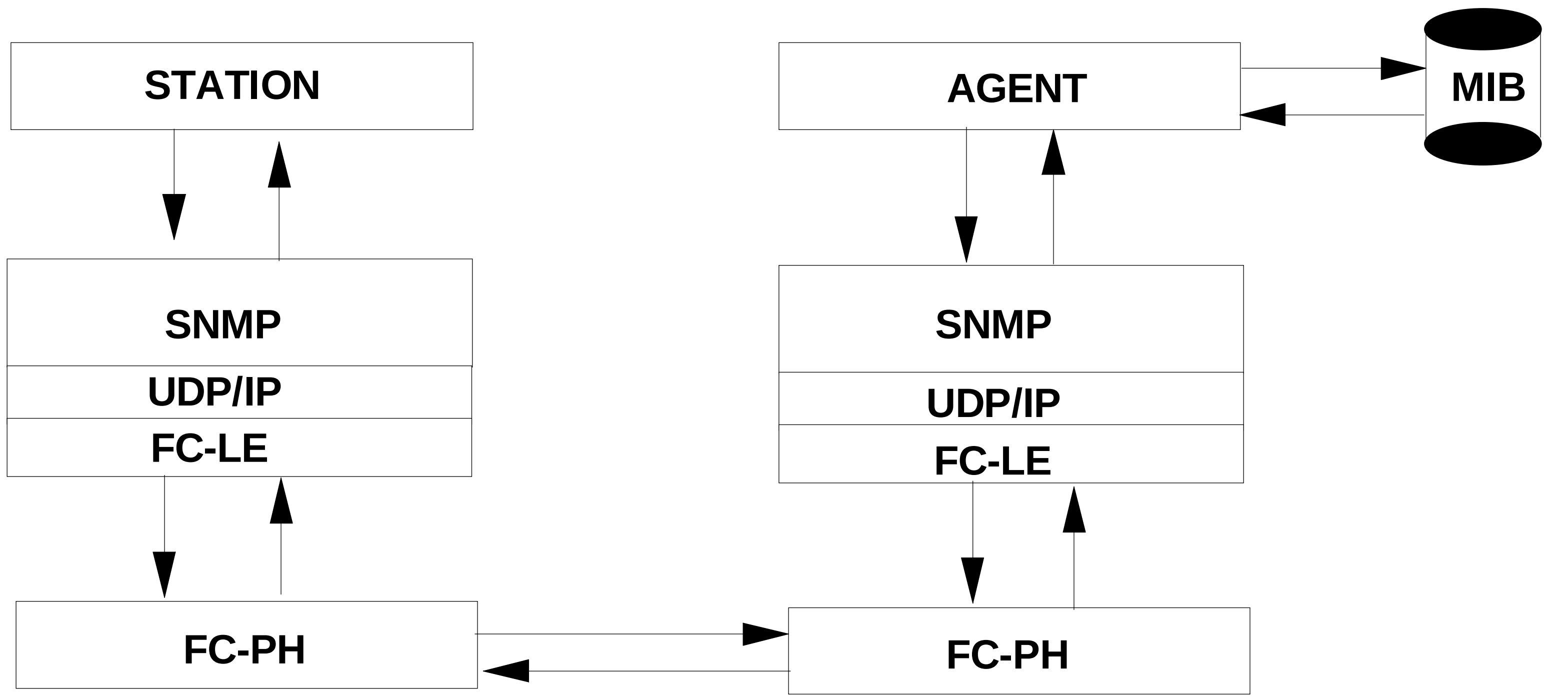
# FIBRE CHANNEL SERVICES SERVER STRUCTURE

- Flexibility of location
  - Within fabric
  - Within nodes external to fabric
- Flexibility of configuration
  - Centralized
  - Distributed
  - Hierarchical
- Appears as a single entity to N_Ports
- Appears as locally connected to N_Ports
- May exist in different areas / domains

# FIBRE CHANNEL SERVICES SERVER CONFIGURATION

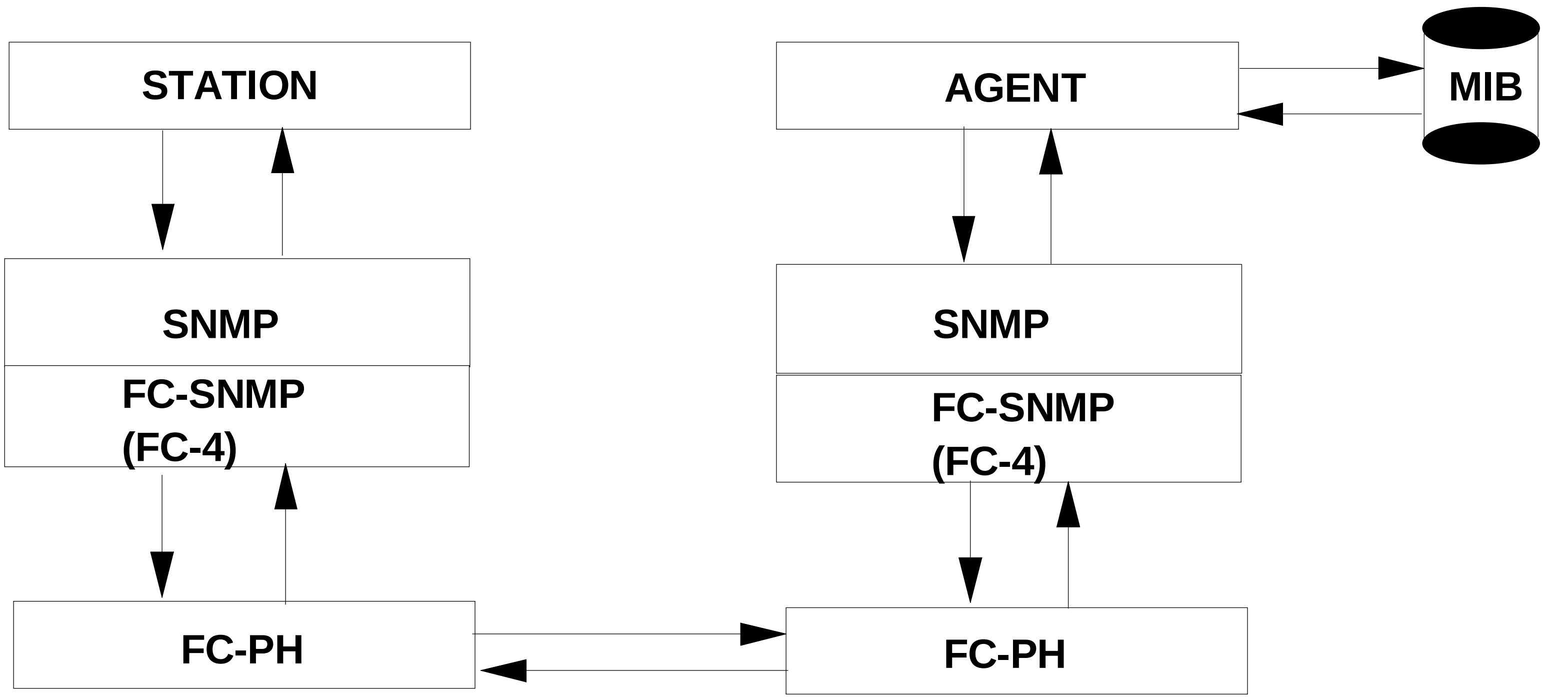


# FIBRE CHANNEL MANAGEMENT USING SNMP OVER FC-LE (CONVENTIONAL)




- A collection of management stations (managers) and management elements (agent)
- A manager to monitor and control the agents
- An agent to perform the network management functions requested by the manager
- Management Information Base (MIB)

# FIBRE CHANNEL MANAGEMENT USING NATIVE SNMP OVER FC-PH

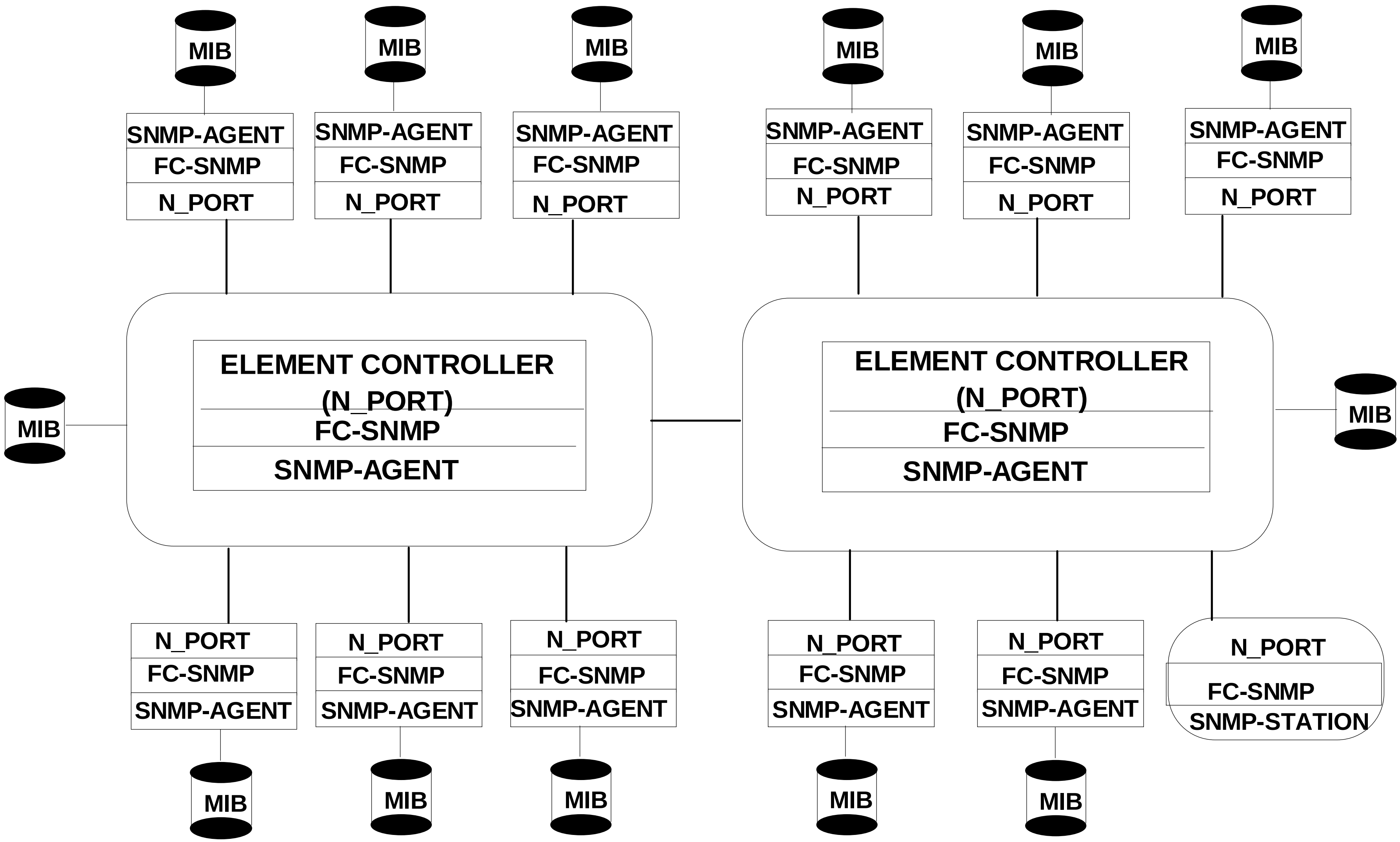



- **SNMP over UDP / IP is too restrictive**

  -Peripheral systems do not support UDP/IP

- **A generic SNMP solution to manage Fibre Channel network consisting of both computing and peripheral devices**

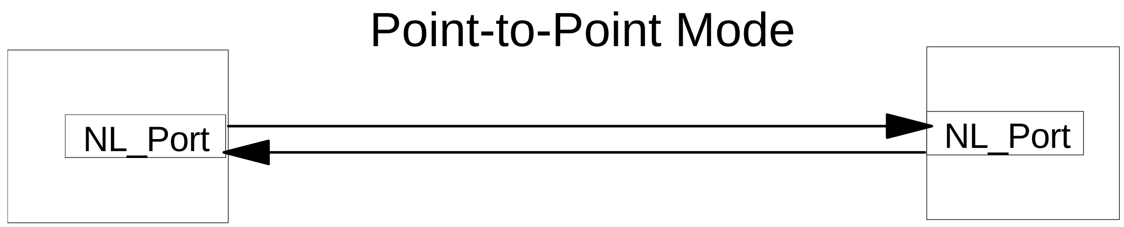
# FIBRE CHANNEL SERVICES
# FC-SNMP MANAGEMENT MODEL

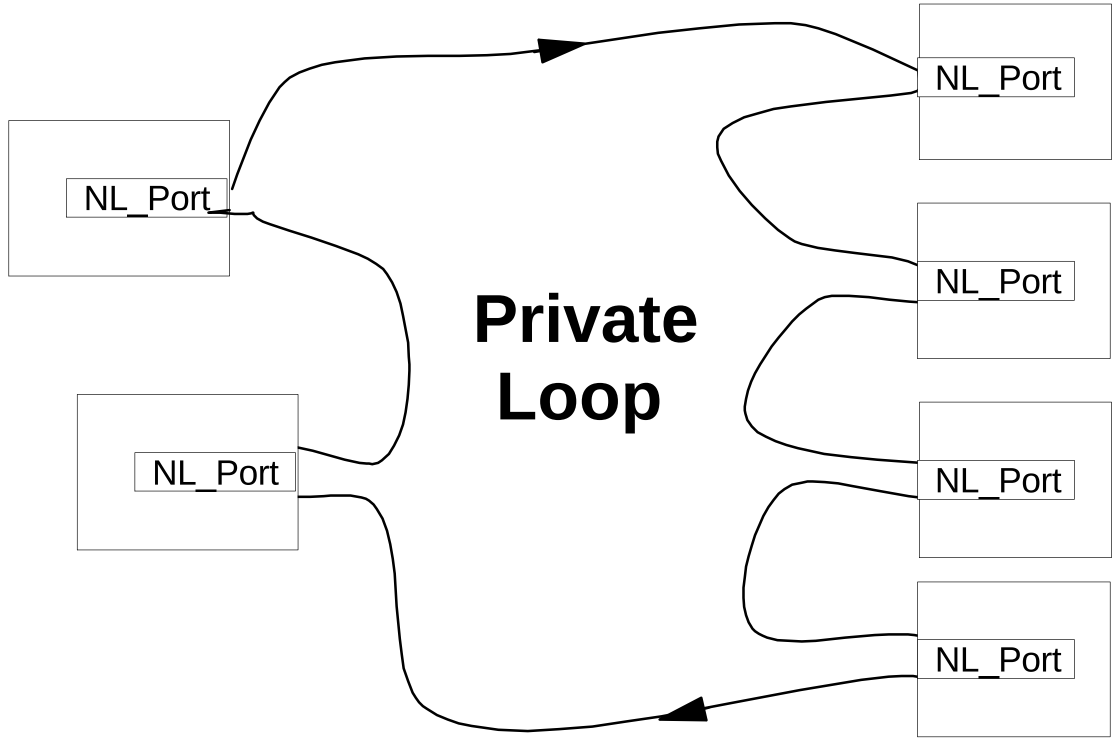


# ARBITRATED LOOP TOPOLOGY

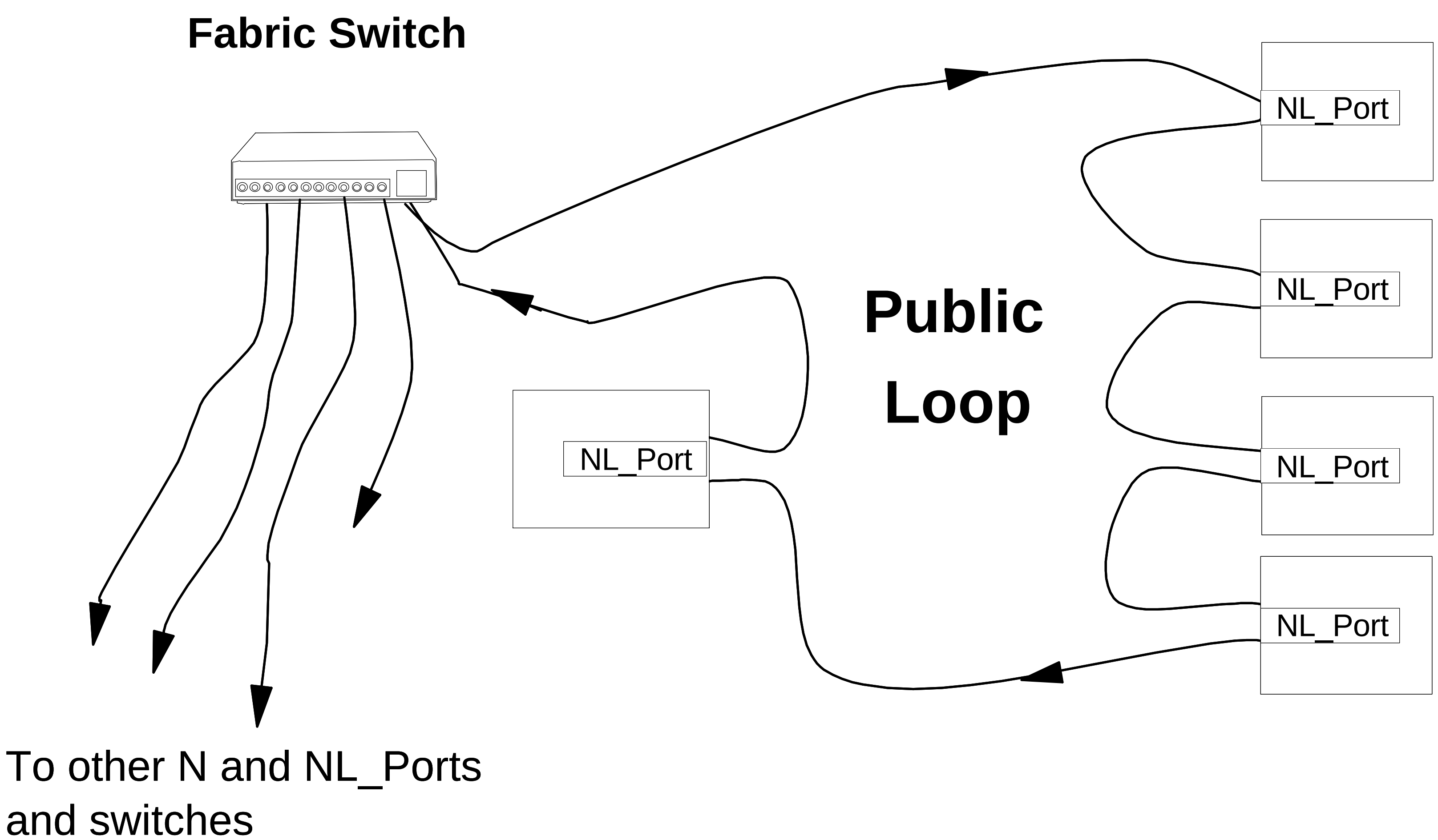
- Requirements are
  - Low cost - allow small configuration to operate without separate cross-point switch
  - Self discovery of connection to loop, else revert to point to point mode
  - No Loop Master (distributed control)
  - Loop can "grow" to attach to a single fabric FL_Port
- Key point
  - Simple protocol addition to FC-PH
- Other characteristics:
  - 126 NL_Ports + 1 FL_Port max
  - Supported by all FC-0 variants
  - Supports all FC-PH protocols and classes
  - Arbitrated access to Loop
  - Access Fairness
  - Preferred address selection supported
- Loop port (NL_Port and FL_Port) is a superset of an N_Port and F_Port:
  - One chip(set) with NL_Port and N_Port functionality is key
  - Simple state machine controls normal operation (11 states)
  - Requires "intelligence" for processing initialization

# ARBITRATED LOOP TOPOLOGY
# PRIVATE LOOP

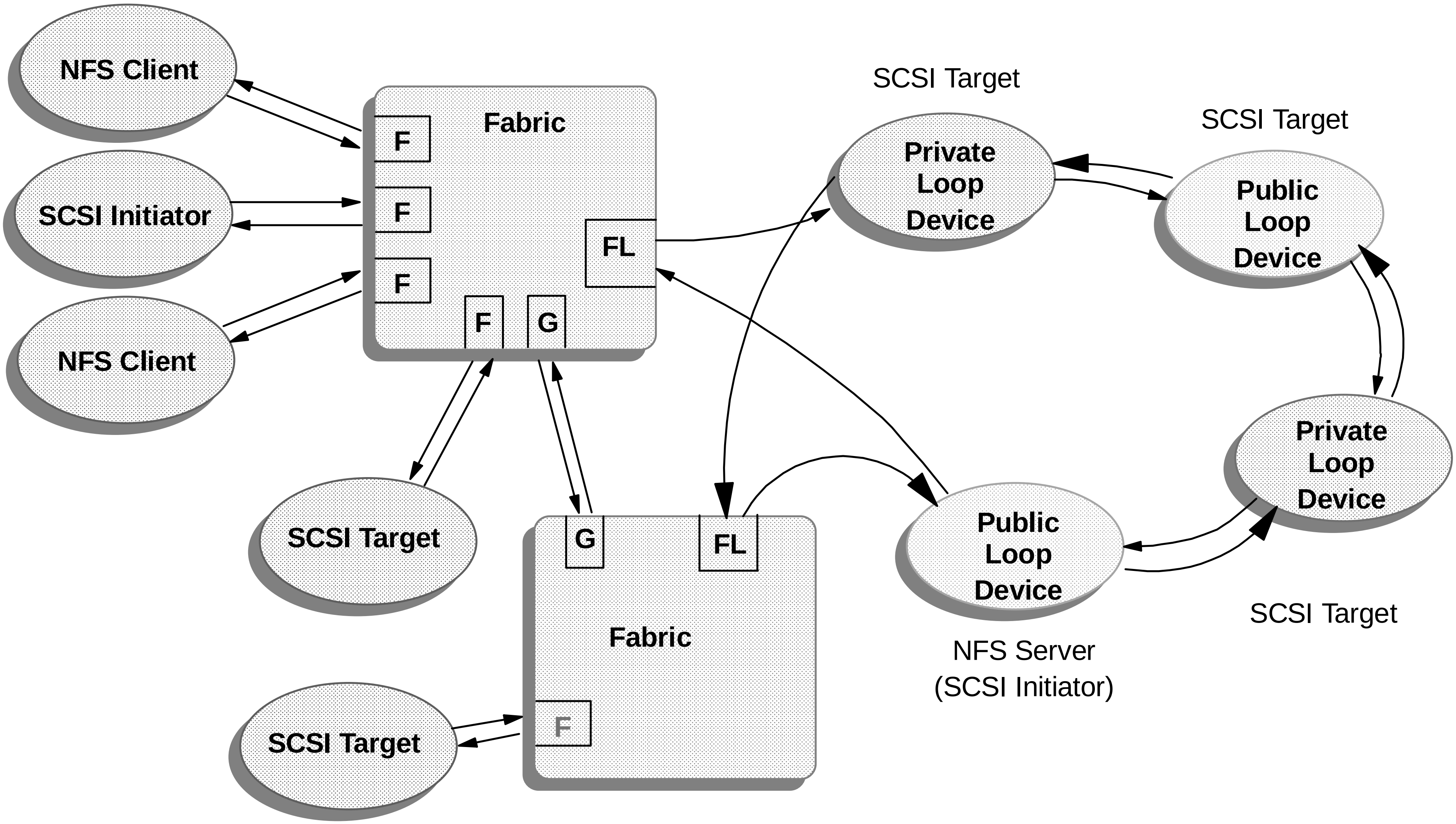


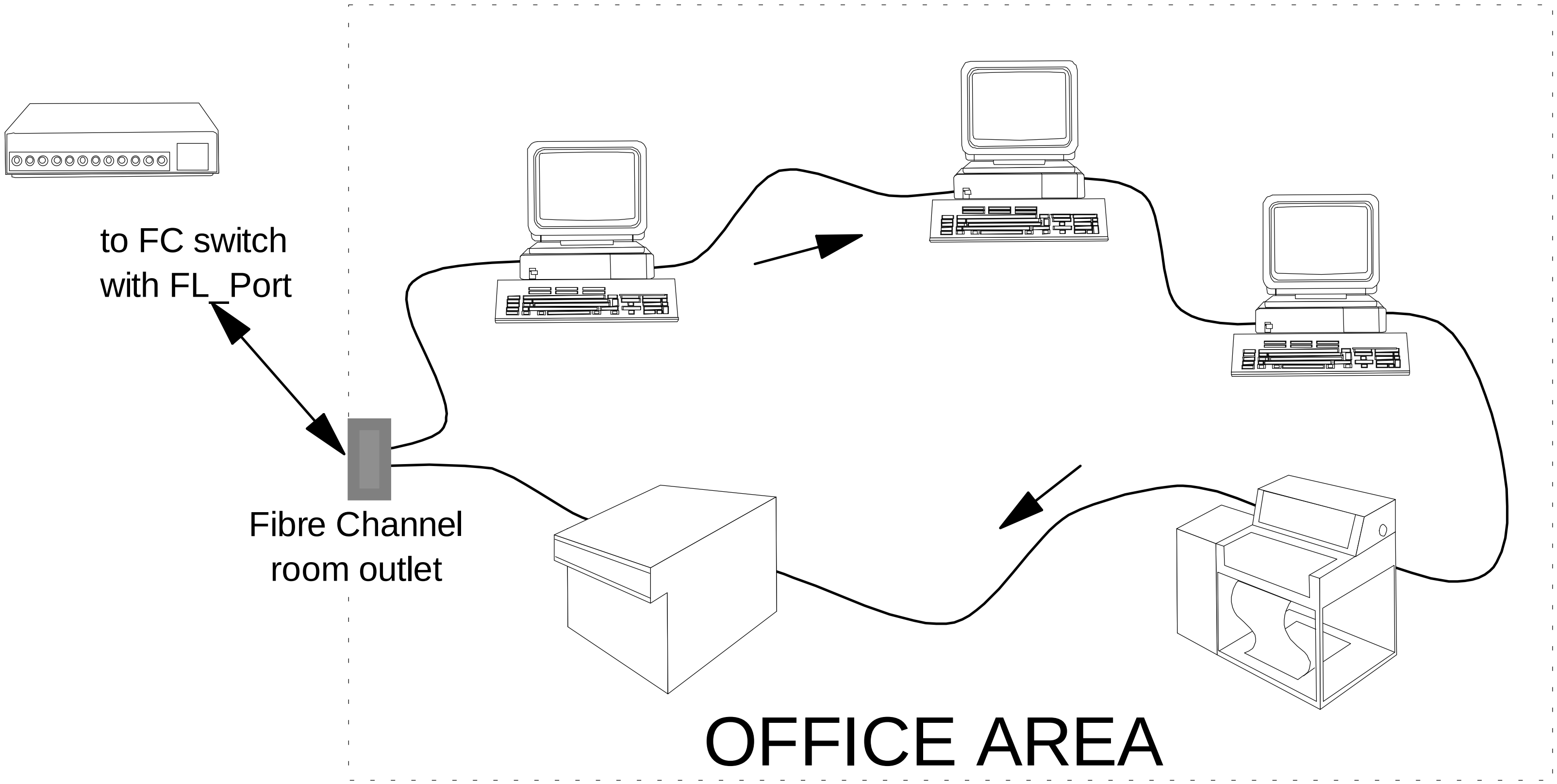
# ARBITRATED LOOP TOPOLOGY
# PUBLIC LOOP



# PUBLIC / PRIVATE NL_PORTS

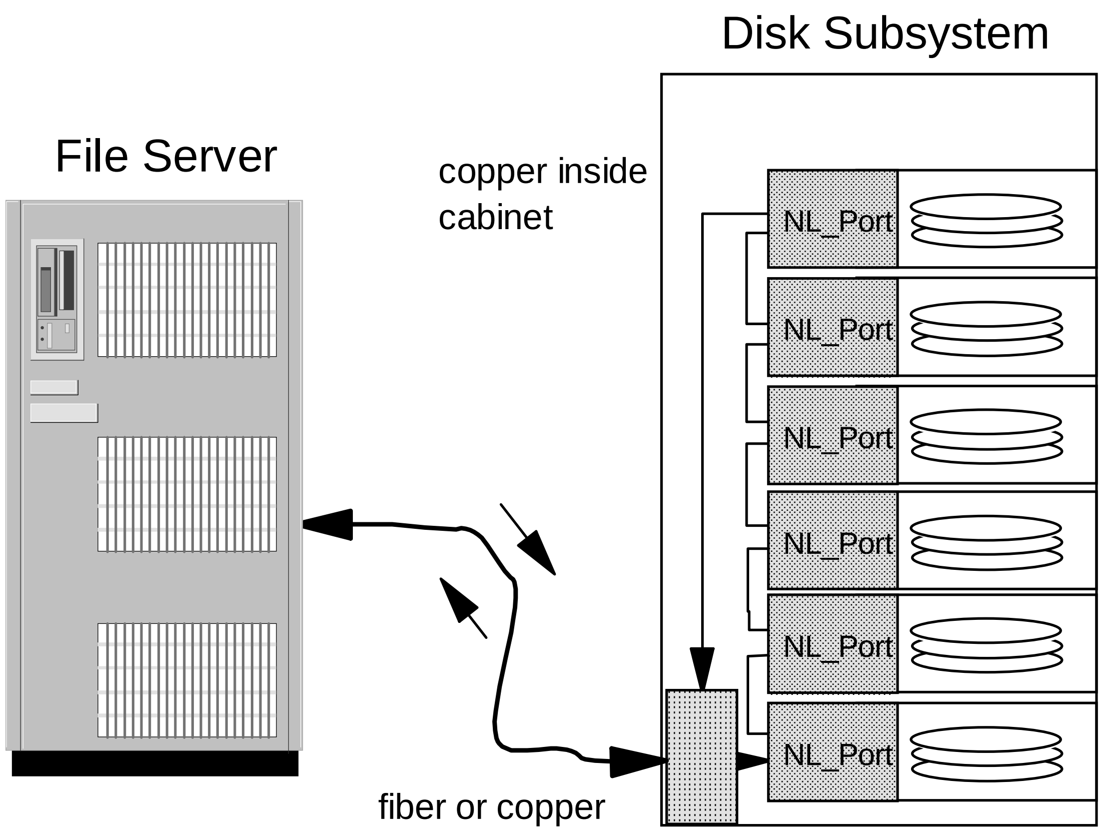


# ARBITRATED LOOP TOPOLOGY
# "OFFICE" LOOP




- Additional NL_Port Fibre Channel devices can be added to office area without need to run additional wiring to FC switch and add F_Ports
- All devices share bandwidth of Loop
- Public Loop comprised of one FL_Port and multiple NL_Ports

# ARBITRATED LOOP TOPOLOGY
# "DISK" LOOP




- Use Loop topology to connect a number of high performance disks
- Highest performance/smallest footprint disk interface available
- High bandwidth for large connectivity
- Private Loop comprised of only NL_Ports

# ARBITRATED LOOP TOPOLOGY ADDITIONAL PROTOCOL

- Simplicity the key:
  - An  L_Port is normally in MONITORING state, looking for an OPN(y) and retransmitting all incoming data
  - When an NL/FL_Port wishes to transmit, it begins replacing IDLE and lower priority ARBx primitive signals with ARB(x), where x is its unique loop address
  - When ARB(x) is received by it's sender, the port has won arbitration and opens its half of the loop (stops retransmitting incoming data)
  - The L_Pport sends an OPN(y) to the port it wishes to communicate with
  - When port Y receives an OPN(y), it opens its half of the loop.  OPN(yx) allows full duplex; OPN(yy)allows half-duplex
  - **Two OPEN NL/FL_Ports transmit and receive as per normal FC rules**
  - Either L_Port may initiate a close by sending a CLS to the other port
  - The other port must also send a CLS back to fully close the loop to allow another port to win arbitration
- Similar to arbitration and selection on existing channels

# ARBITRATED LOOP TOPOLOGY CHARACTERISTICS

| | |
|---|---|
| ▪ Connectivity: | n Number of L_Ports<br>127 Active Addresses |
| ▪ FC-PH classes: | Class 1, 2, and 3 (4 and 5 in FC-PH-2) |
| ▪ Low Cost: | Requires no fabric; reduces transmitters / receivers |
| ▪ End Node: | Can operate as both N_Port and NL_Port;<br>may connect to an F_Port |
| ▪ Fabric Node: | Can operate as both F_Port and FL_Port;<br>may connect to an N_Port |
| ▪ Fairness: | L_Port cannot arbitrate again until all other arbitrating<br>L_Ports have won arbitration |
| ▪ Addressing: | Assigned by switches, hard-coded, or Fabric |
| ▪ Replicate: | Supports broadcast, multicast, selective multicast |

# ARBITRATED LOOP TOPOLOGY
# ADDITIONAL LOOP ISSUES

- Resilience of Loop to power failure is procurement issue:
  - Active bypass circuits are available
  - In low-cost procurements,  a power failure in a single Node may cause Loop to fail
- Various degrees of resilience possible
  - Redundant Loops
  - Coax relays, optical by-pass switches
  - Active Tee's
  - Active Hubs
- More than 126 NL_Ports and more than 1 FL_Port may be connected in the Loop, but only in non-participating mode (repeating - no Loop address)

# ARBITRATED LOOP TOPOLOGY PRIMITIVES

- **New FC-PH Primitives:**

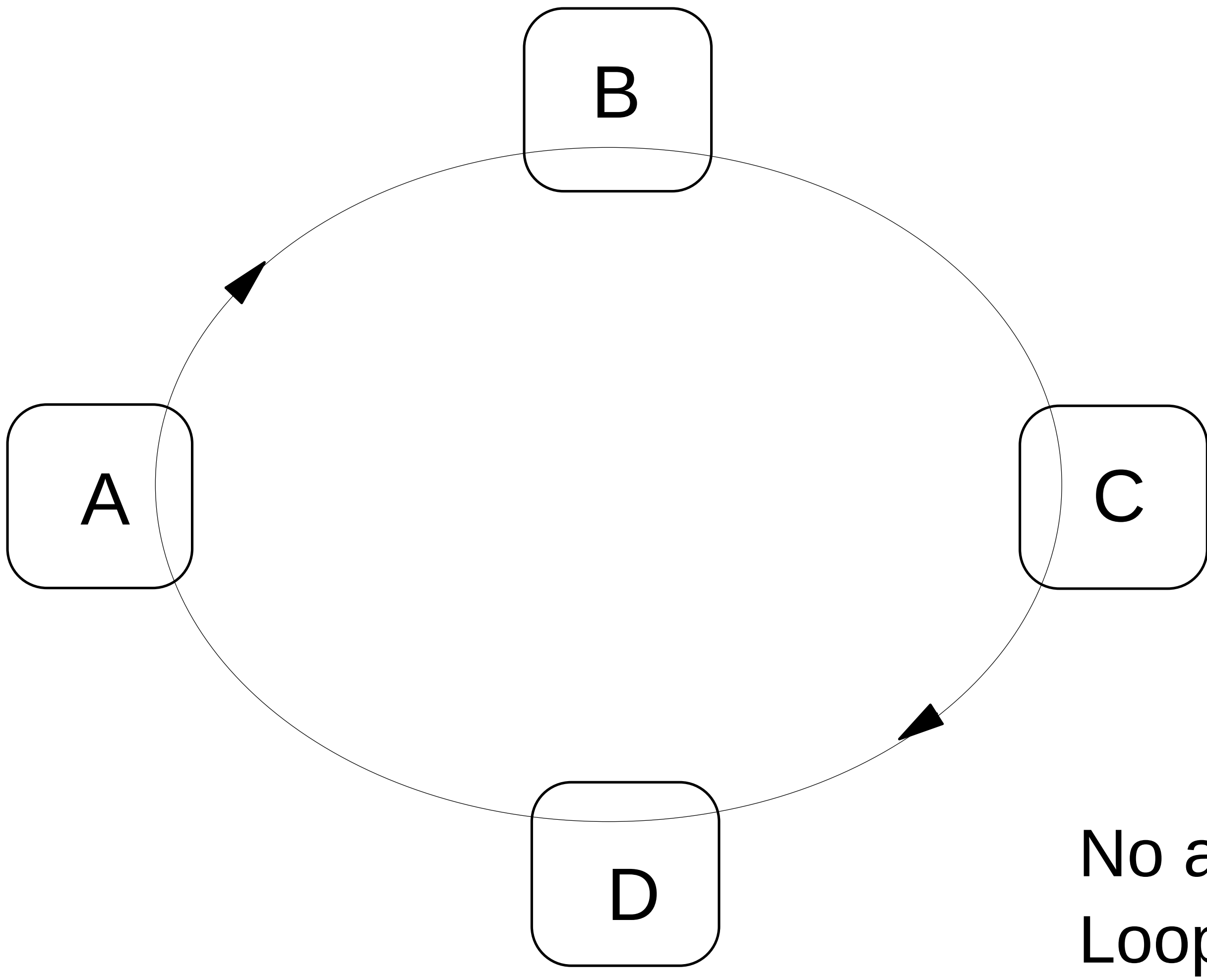
  - **ARBx** Arbitrate Address x
  - **OPNyx** Open Address y - Full Duplex Mode
  - **OPNyy** Open Address y - Half Duplex Mode
  - **OPNfr** Open Broadcast Replicate
  - **OPNyr** Open Selective Replicate
  - **CLS** Close receiving Node
  - **LIP** Loop Initialization - Five Variations
  - **LPByx** L_Port Bypass
  - **LPEyx** L_Port Enable (includes ALL)
  - **MRKtx** Mark (Synchronization)

# ARBITRATED LOOP TOPOLOGY ADDRESSING

– One byte Arbitrated Loop Physical Address (AL_PA) in hex:

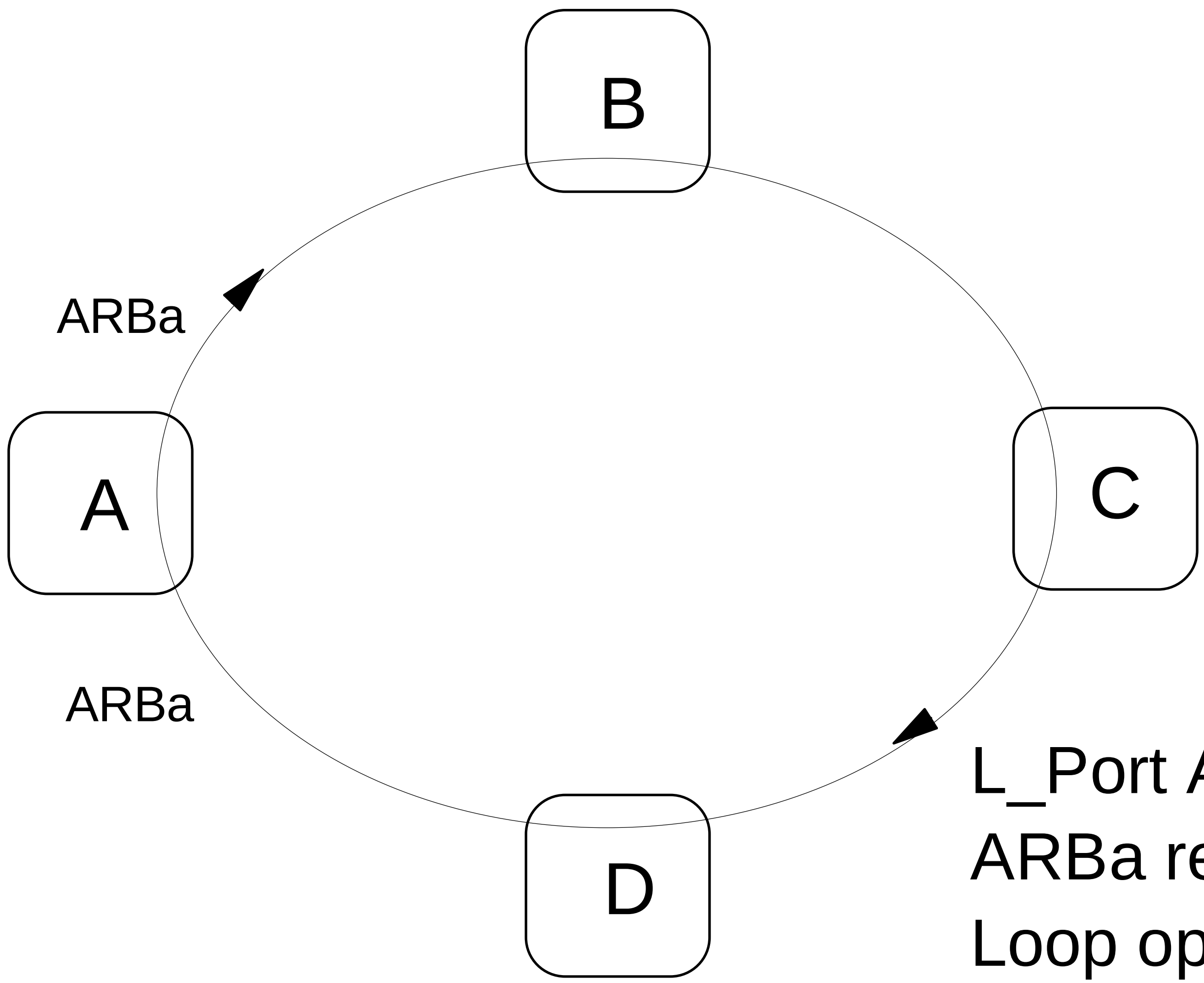
| | |
|---|---|
| **00** | Reserved for FL_Port |
| **01-EF** | Available for active nodes (126 neutral disparity values). **Extra L_Ports may be connected in the Loop, but in n-participating mode (no Loop address); however, addresses may be shared.** |
| **F0** | Reserved for access fairness algorithm |
| **F1-F6** | Invalid |
| **F7-F8** | Flags used with LIP |
| F9-FE | Reserved |
| **FF** | Used in LIP, OPN, and LPE to address ALL L_Ports |

# ARBITRATED LOOP TOPOLOGY
# LOOP CLOSED (IDLE)




No activity
Loop is closed
Nodes retransmitting

# ARBITRATED LOOP TOPOLOGY
# 'A' ARBITRATING



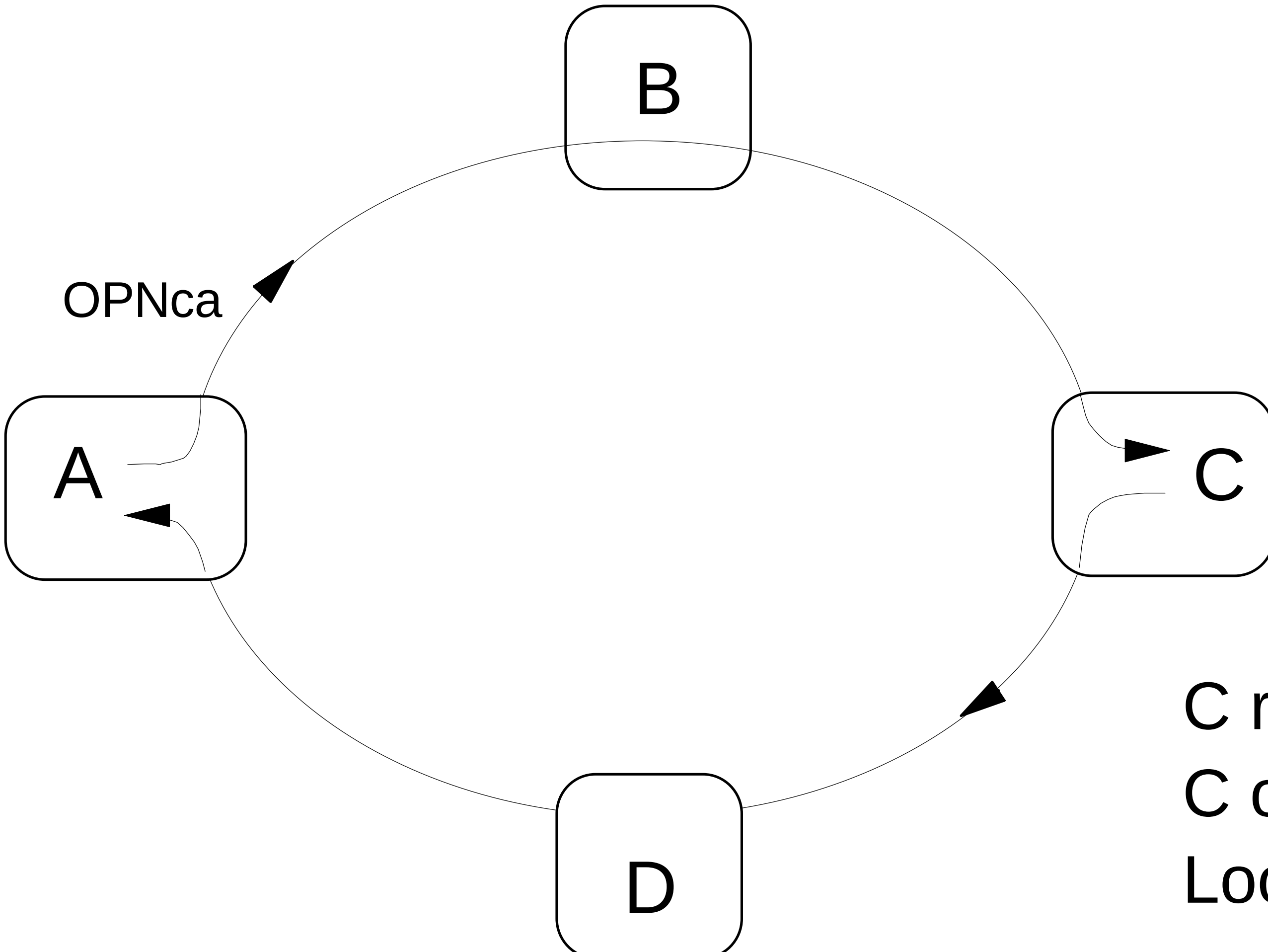

L_Port A transmits ARBa

ARBa received

Loop open at A

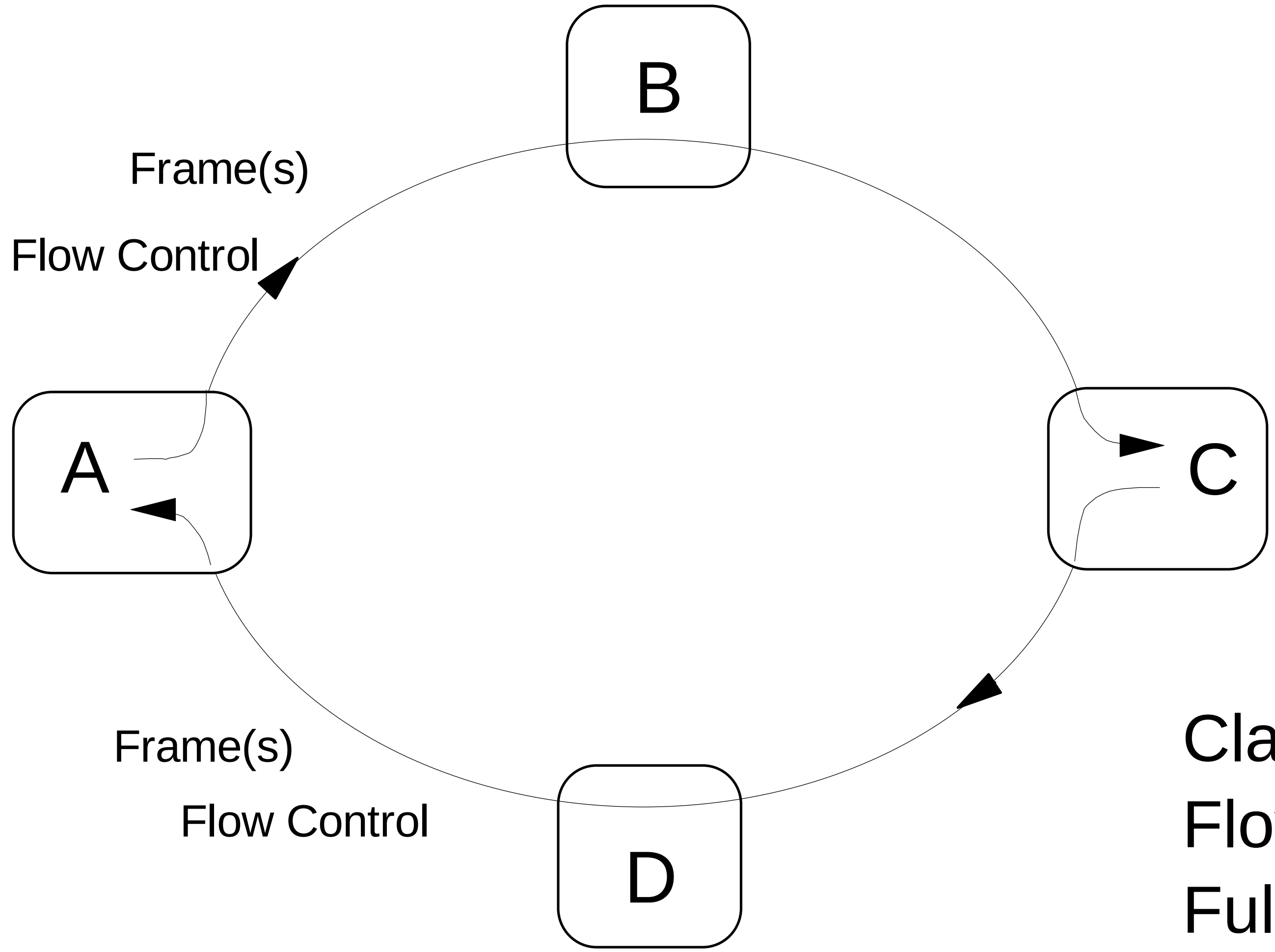
L_Port A can send OPNca

# ARBITRATED LOOP TOPOLOGY
# 'A' OPENS 'C'



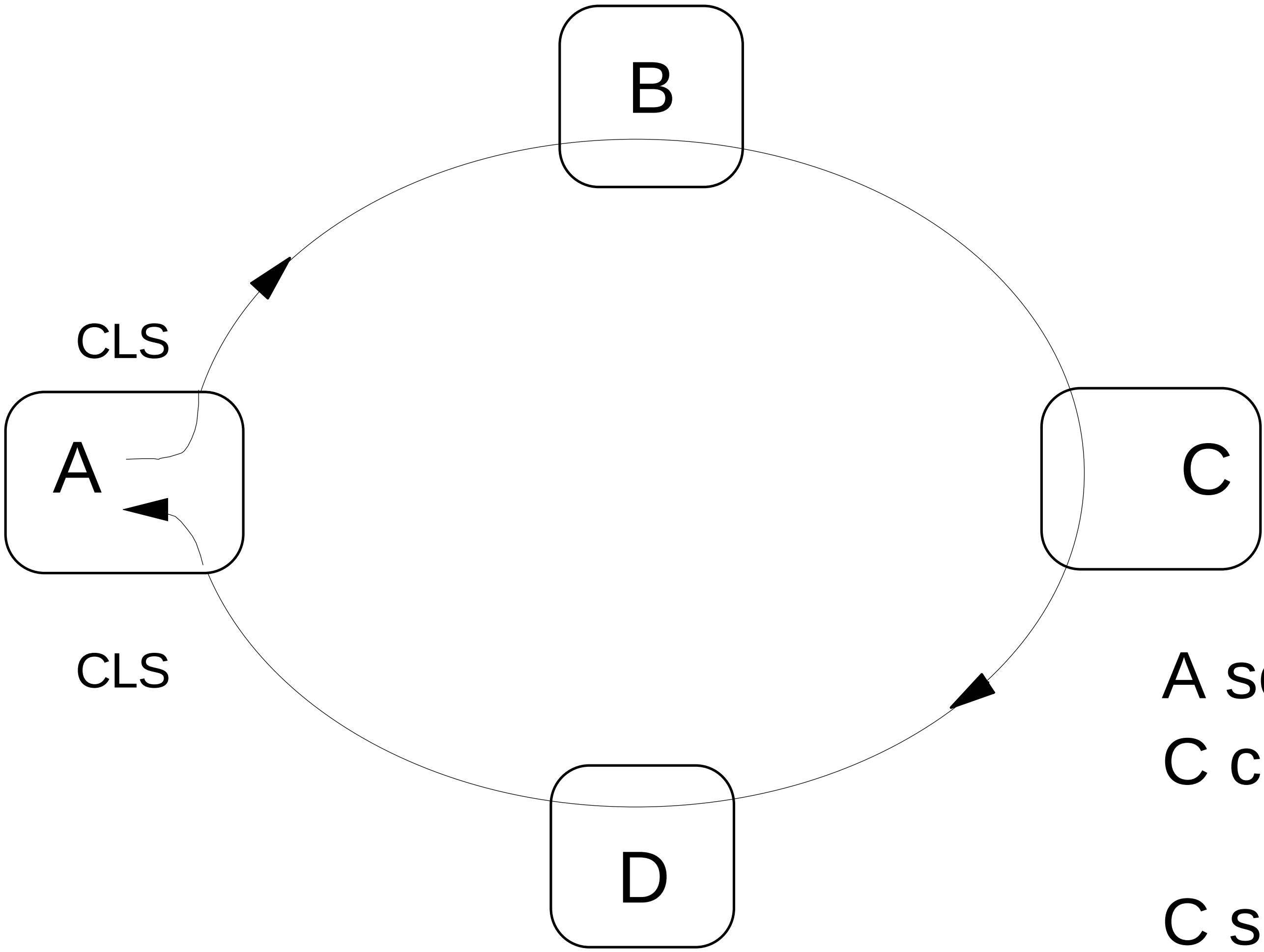

C receives OPNca
C opens
Loop now is FC-PH
point-point A to C

# ARBITRATED LOOP TOPOLOGY 'A' AND 'C' USING POINT TO POINT TOPOLOGY RULES




Class 1, 2, 3, (4, 5)
Flow control
Full-Duplex

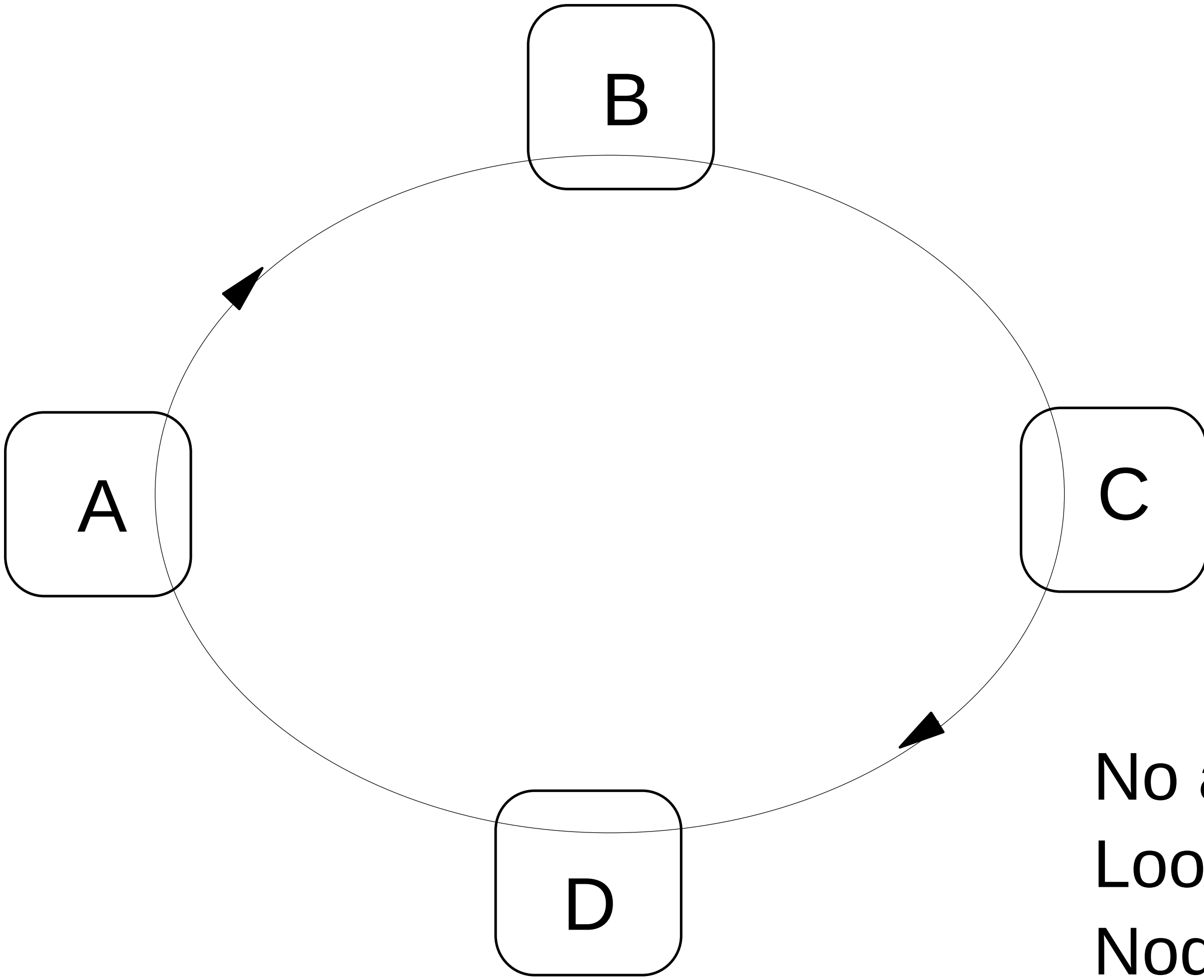
# ARBITRATED LOOP TOPOLOGY
# 'A' CLOSES 'C'




A sends CLS
C closes

C sends CLS
A closes

# ARBITRATED LOOP TOPOLOGY
# LOOP CLOSED (IDLE)




No activity
Loop is closed
Nodes retransmitting

# ARBITRATED LOOP FLOW CONTROL

- **Buffer-to-Buffer Credit (BB_Credit)**

| | | |
|---|---|---|
| Login BB_Credit | - | default is 1, but controlled by each L_Port value, or by lowest value of any L_Port; used at OPN only |
| BB_Credit = 0 | - | requires a handshake |
| BB_Credit = 1 | - | allows OPN, Frame, CLS |
| BB_Credit > 1 | - | allows OPN, n Frames, CLS |

# ARBITRATED LOOP TOPOLOGY ADVANTAGES

- Lowest cost solution when combined with copper physical variants, requires half the number of transmitter/receivers.
- No separate Fabric required to connect  n Nodes (but is allowed)
- Back-panel daisy-chain scheme eliminates wiring clutter in configurations with many peripherals (e.g., disk arrays)
- Bypass switch design permits minimal power operation with functional Loop
- Simple additions to current FC-PH protocol defined in FC-AL
- Easy expands by addition of a Fabric switch

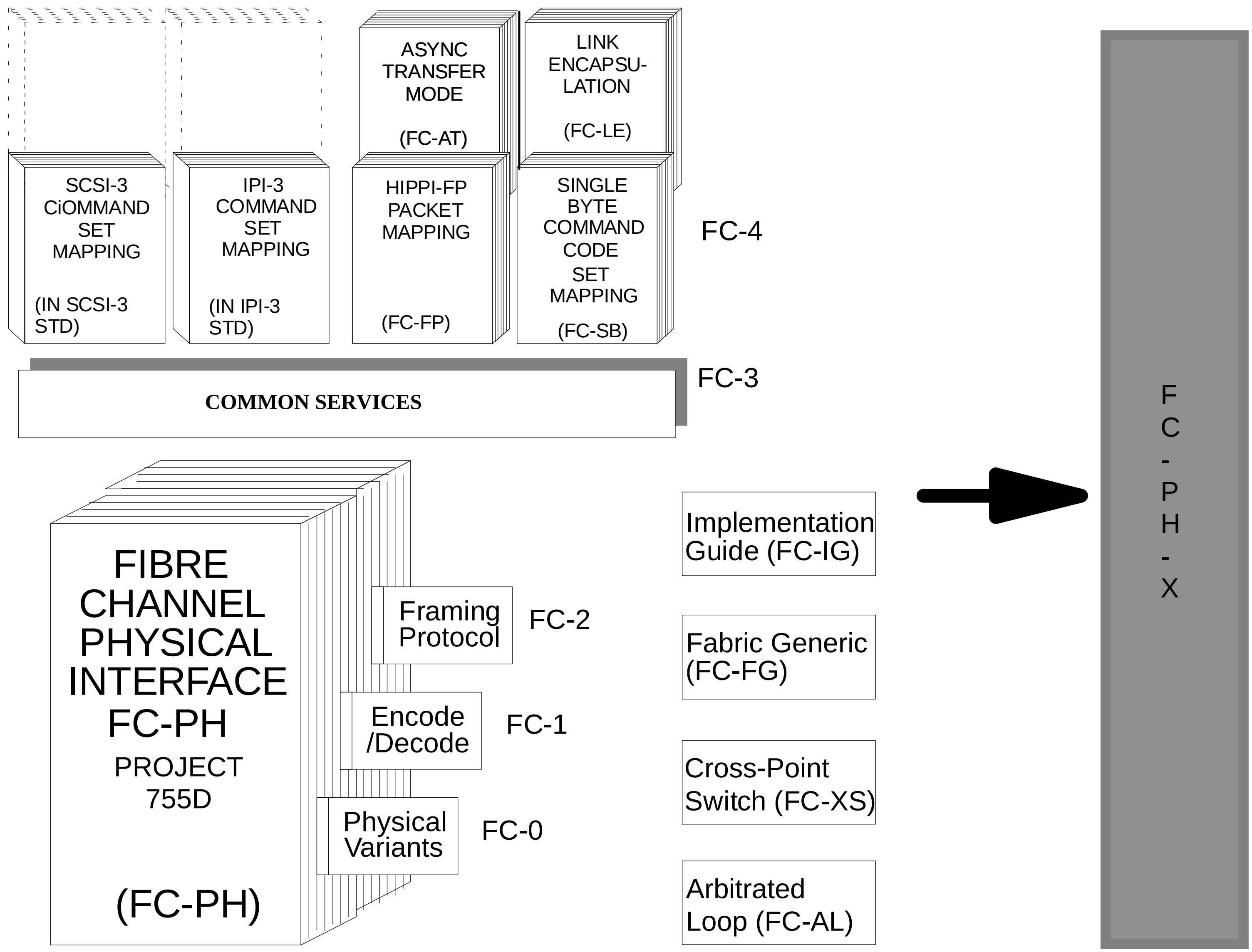
# ARBITRATED LOOP TOPOLOGY SUMMARY

- Compatible with FC-PH - **just another topology**
- Simple protocol
- Low-cost connectivity for 126 active Nodes
- Looks like point-point
- Supports:
  - Fabric topologies
  - Fairness
  - Replicate - Broadcast and Selective Multicast
  - Isochronous and multiple L_Port transfers

# THE PRESENT AND THE FUTURE

- Is a standard ever finished?
- FC-EP has considerable new function
- FC-EP mostly extends cross-point switch functions
- New  physical variants (2Gbaud, 4Gbaud, 8Gbaud?)

# FIBRE CHANNEL LEVELS THE FUTURE



# FIBRE CHANNEL ENHANCED PHYSICAL (FC-PH-2 and FC-PH-3)

- Next-generation FC-PH
- New project beginning work in ANSI X3T11
- Extends functionality of existing Fibre Channel definition
- Builds on existing FC-PH - no changes to FC-2, FC-1 or FC-0
- Provides additional functions in consistent manner for use by all protocols:
  - Striping - Transmission of single unit of information in parallel across multiple Links (for greater bandwidth)
  - Hunt Groups - Allow frames for single unit of information to be delivered to one of a set of n N_Ports (for greater efficiency)
  - Multicast - Allow Data frames to be delivered to multiple N_Ports
  - Provision of guaranteed bandwidth and latency services:
    - Supports multimedia and various types of real-time traffic
    - Extension of current Fabric capabilities only

# FC DOCUMENT STATUS

**Projected dates as of 8/2/93 to forward documents:**

| | | |
|---|---|---|
| ▪ FC-AL | Arbitrated Loop | Oct93 |
| ▪ FC-EP | Enhanced Physical | Aug94 |
| ▪ FC-FG | Fabric Generic | Dec93 |
| ▪ FC-FP | HIPPI Framing Protocol | Aug93 |
| ▪ FC-IG | Implementation Guide | Oct93 |
| ▪ FC-LE | Link Encapsulation | Aug93 |
| ▪ FC-PH | Physical (2nd review) | Aug93 |
| ▪ FC-SB | Mapping to Single-Byte Command Code Sets | Jun93 |
| ▪ FC-SW | Cross Point Switch Fabric | Dec93 |
| ▪ IPI-3 | IPI-3 Mapping on FC-PH | Dec95 |
| ▪ FC-ATM | ATM over FC | proposed project |
| ▪ SCSI-GPP | SCSI-3 Generic Packetized Protocol | in X3T9.2 (SCSI) |
| ▪ SCSI-FCP | SCSI Packetized Protocol | in X3T9.2 (SCSI) |
| ▪ HIPPI-FC | FC over HIPPI | Dec93 |
| ▪ FC-SG | Services Generic | proposed project |

# OPPORTUNITIES

- A new era in performance:
  - FC's full-duplex operation can cut number of connections in half
  - Multiplexing permits more device and media sharing
- Distance opens up new application opportunities:
  - Direct device connection instead of via File Server
  - Locate processors near people but protect and isolate data
  - Improved security, unmatched system layout convenience
- True mass-market for fiber-optics to drive cost down:
  - Millions SCSI peripherals in 1990
  - SCSI was a $7 billion market in 1992
  - IPI grew to 1 million units/year by 1992
  - High-performance units growing
  - Network attached peripherals are the future

# SUMMARY

- Fiber Channel is a comprehensive physical interface for multiple existing and new protocols
    - May be the only interface a system needs!!
- Fiber-optic technology is available to meet FC requirements in a cost-effective manner
- Copper technology meets FC requirements and is the cheapest (today)
- Fiber Channel definition is stable enough to begin design NOW
- FC-PH has been forwarded for Second Public Review
- "**Serial is Serious**"

# FOR MORE INFORMATION

- **THE REAL THING** - FC-PH is X3.230-1994
  - Available from Global Engineering at 1-800-854-7179 for $65
- Previous revision - FC-PH Revision 4.0 (397 pages). IS X3.230-199x (X3t9.3/93-090), in Jun93 mailing beginning page 593.
- Previous revision - FC-PH Revision 3.0. Is X3T9.3/92-092, in Aug92 X3T9.3 mailing beginning page 665