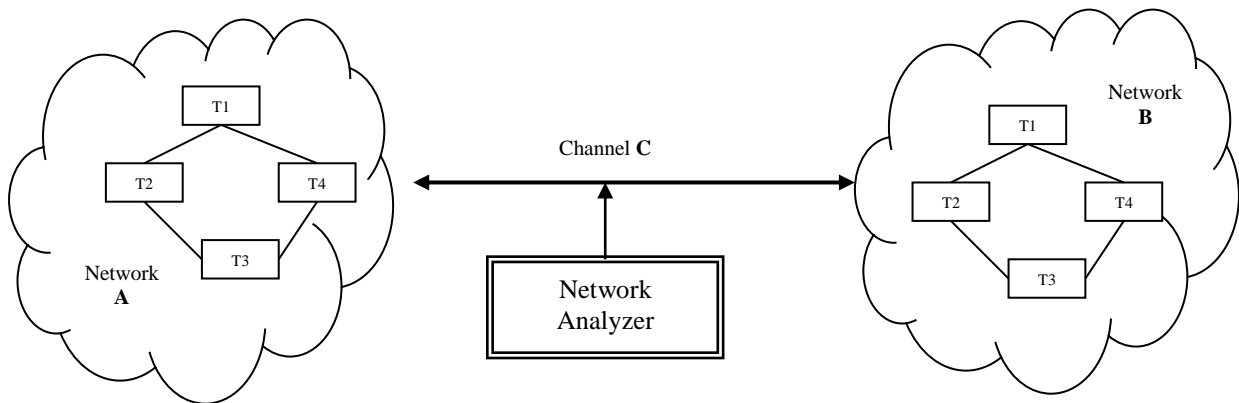
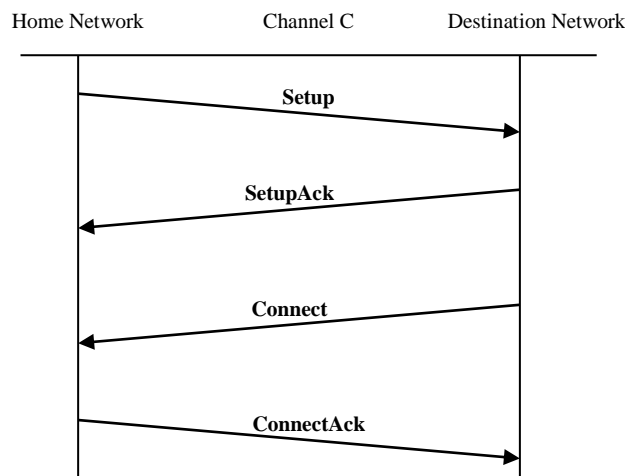


UNITS 7 AND 8: SEMANTIC ANALYSIS and ERROR HANDLING

There are two terminal networks (A and B) connected by a half-duplex C transmission channel (transmitted in both directions, but not at the same time). It also has a network analyzer that reads the messages that circulate through the channel



When one terminal of a network wants to establish a communication with a terminal of the other network, four messages are exchanged through channel C described by the following protocol:



From the home network a **Setup message** is sent to the destination network indicating that it wants to establish a communication with a specific machine of that network. The destination network must respond with a **SetupAck** message as acknowledgment of the received **Setup** message. Once the destination network terminal accepts the call request, it must send a **Connect message** to the terminal of the source network. The source network must respond with a **ConnectAck** as an acknowledgment of the received Connect message.

Each message consists of a series of Information Elements (**EI**). Each **EI**, in turn, consists of two fields. The first field always has a size of 1 byte and its value corresponds to the identifier of that **EI**. The second field will be the **EI** argument and its size depends on the **EI**.

The messages with the **EI**s of which they are composed are described below. These **EI** must appear in the order shown in the following table. The **EI** of a message may be mandatory or optional. Mandatory **EI** must always be included in the message. The optional **EI**s may or may not appear.

MESSAGE	Information Elements	Mandatory / Optional
Setup	EISetup	Mandatory
	EIAddress	Mandatory
	EIOrigin	Optional
	EIDestination	Mandatory
SetupAck	EISetupAck	Mandatory
	EIDirection	Mandatory
Connect	EIConnect	Mandatory
	EIDirection	Mandatory
ConnectAck	EIConnectAck	Mandatory
	EIDirection	Mandatory

The following table describes the defined **EI** with the description of the fields of which it is composed. For each field its size in bytes and its value are indicated. The first field always has a fixed value since it is the identifier of the **EI** and the second field has a variable value that corresponds to the argument of described **EI**:

		FIELDS					
		Field 1 (Identifier EI)			Campo 2 (Identifier EI)		
		Name	Size	Value	Name	Size	Value
EI	EISetup	IDSetup	1 byte	'A'	CREf	1 byte	[A-Z,a-z,0-9] Any alphabetic or numeric character
	EISetupAck	IDSetupAck	1 byte	'B'	CREf	1 byte	Same that Setup
	EIConnect	IDConnect	1 byte	'C'	CREf	1 byte	Same that SetupAck
	EIConnectAck	IDConnectAck	1 byte	'D'	CREf	1 byte	Same that Connect
	EIDirection	IDDirection	1 byte	'E'	Direction	1 byte	A→B = '0' B→A = '1'
	EIOrigin	IDOrigin	1 byte	'F'	Origin	3 bytes	[0-9] ³ . Three digits
	EIDestination	IDDestination	1 byte	'G'	Destination	3 bytes	[0-9] ³ . Three digits

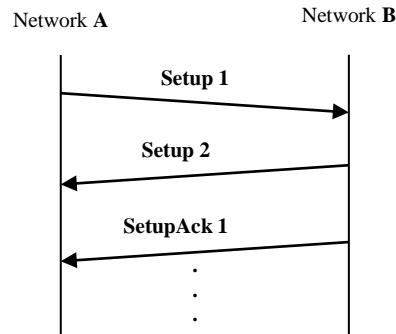
Since messages of several different communications can be circulated through the same channel, it is possible that a message of another communication appears between the four messages corresponding to the same communication. In order to identify the communication to which a message belongs, the CREf field of **EISetup**, **EISetupAck**, **EIConnect** and **EIConnectAck** is used. Therefore, the four messages of which a communication is composed must contain the same value in their CREf argument of said Information Elements.

On the other hand, the **EI** EIDirection indicates whether the message has been transmitted in the A→B or B→A direction. Its possible values are '0' or '1' respectively.

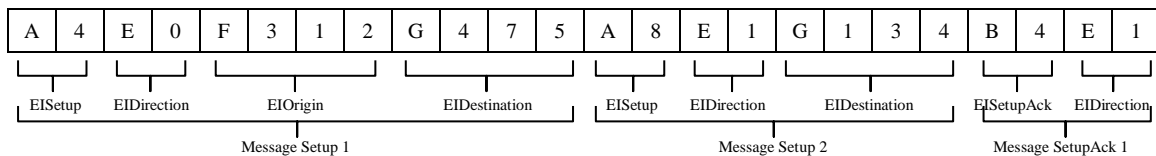
EIOrigin and EIDestination contain in their argument the address of the origin and destination terminals respectively.

Example:

Imagine that a **Setup** is sent from A to B, but before receiving the **SetupAck** from B, a **Setup** from B to A corresponding to another communication has been sent



The network analyzer would collect the following data:



It is required:

1. Describe formally the automaton that the network analyzer uses as a lexical analyzer, whose tokens are the Elements of Information (**EI**).
2. Define a grammar that generates messages that can be read on channel C. Is it LL(1)? Otherwise, make the necessary modifications to make it.
3. Can the grammar of the second exercise be used to perform an LR(1) analysis? If not, modify it to make it so. Generate the first 8 states of the LR(1) analyzer for that grammar.
4. Describe the semantic routines that should be added to the grammar, as well as the additional data structures (table of symbols, stacks, etc.) necessary to control whether the four messages corresponding to a communication have been received and in their correct order. In the case of correctly reading all messages of a communication, they will be printed on the analyzer screen. If an erroneous message is received, an error will also be printed with the Failed Message. Optionally, when an erroneous message is read, additional information can be given, for example, that a **Connect** has been read without having read the **SetupAck**. In this case you can print the **Setup** and the **Connect** and notify the incident.