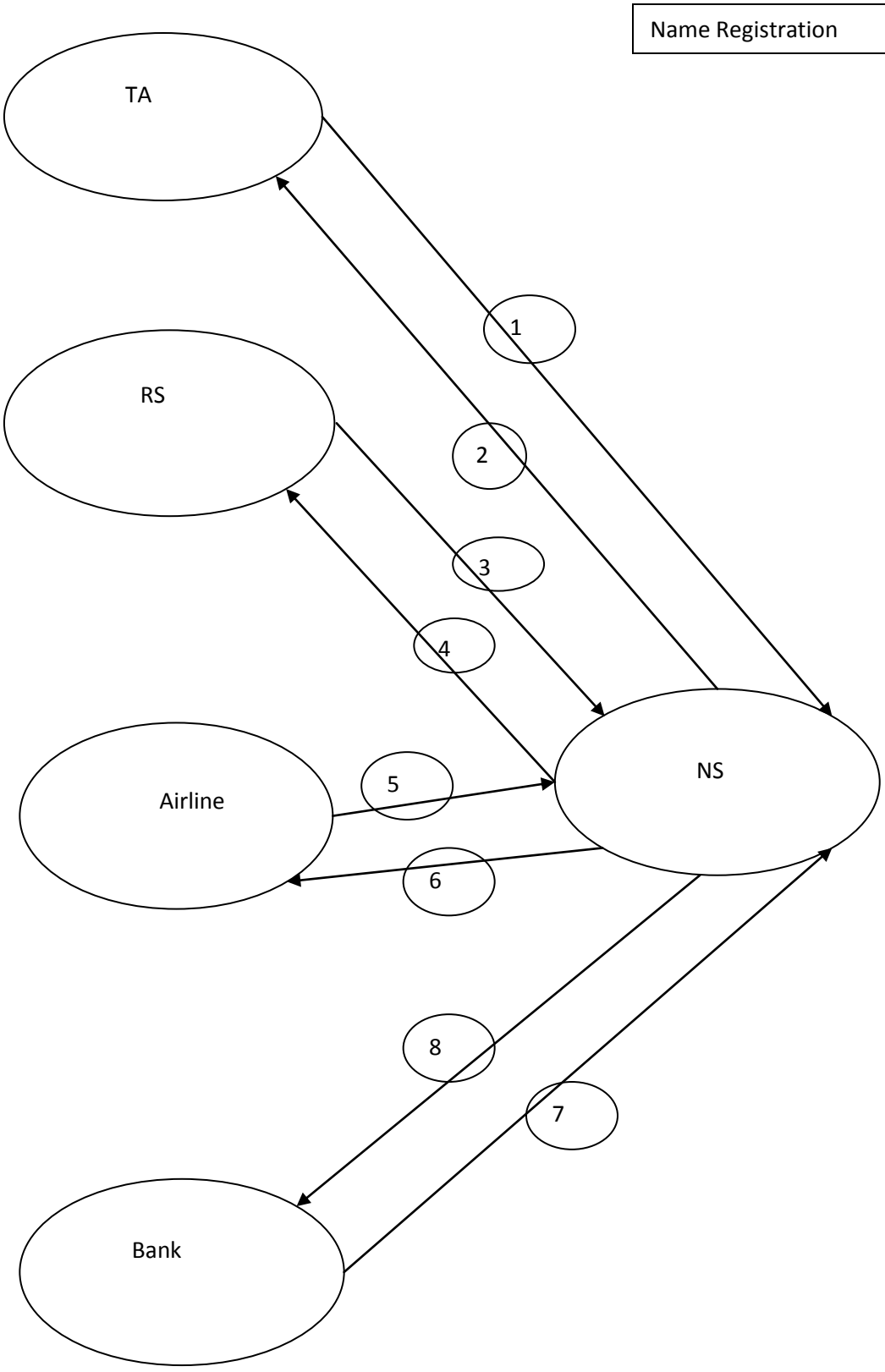
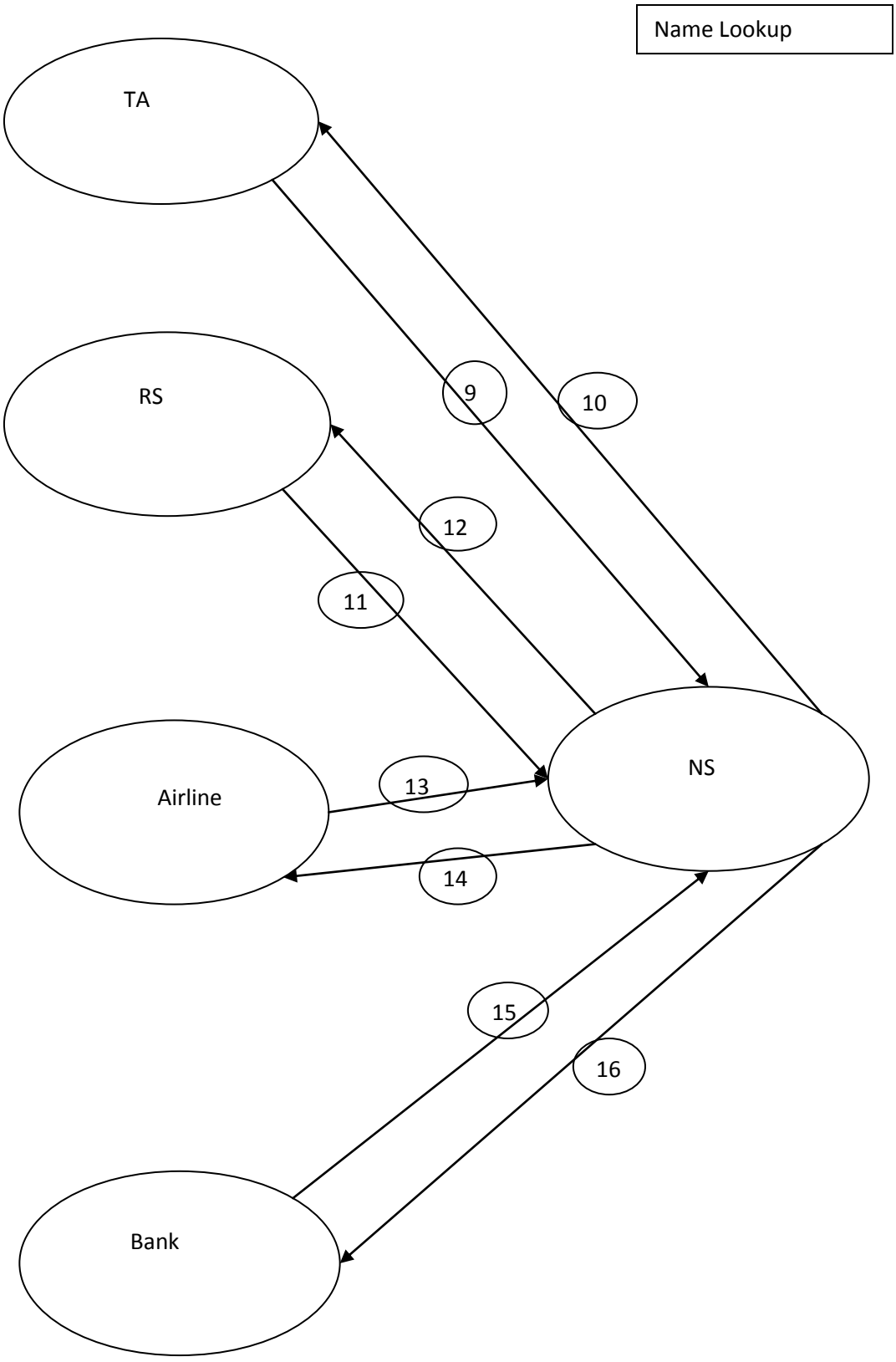
1. Figures

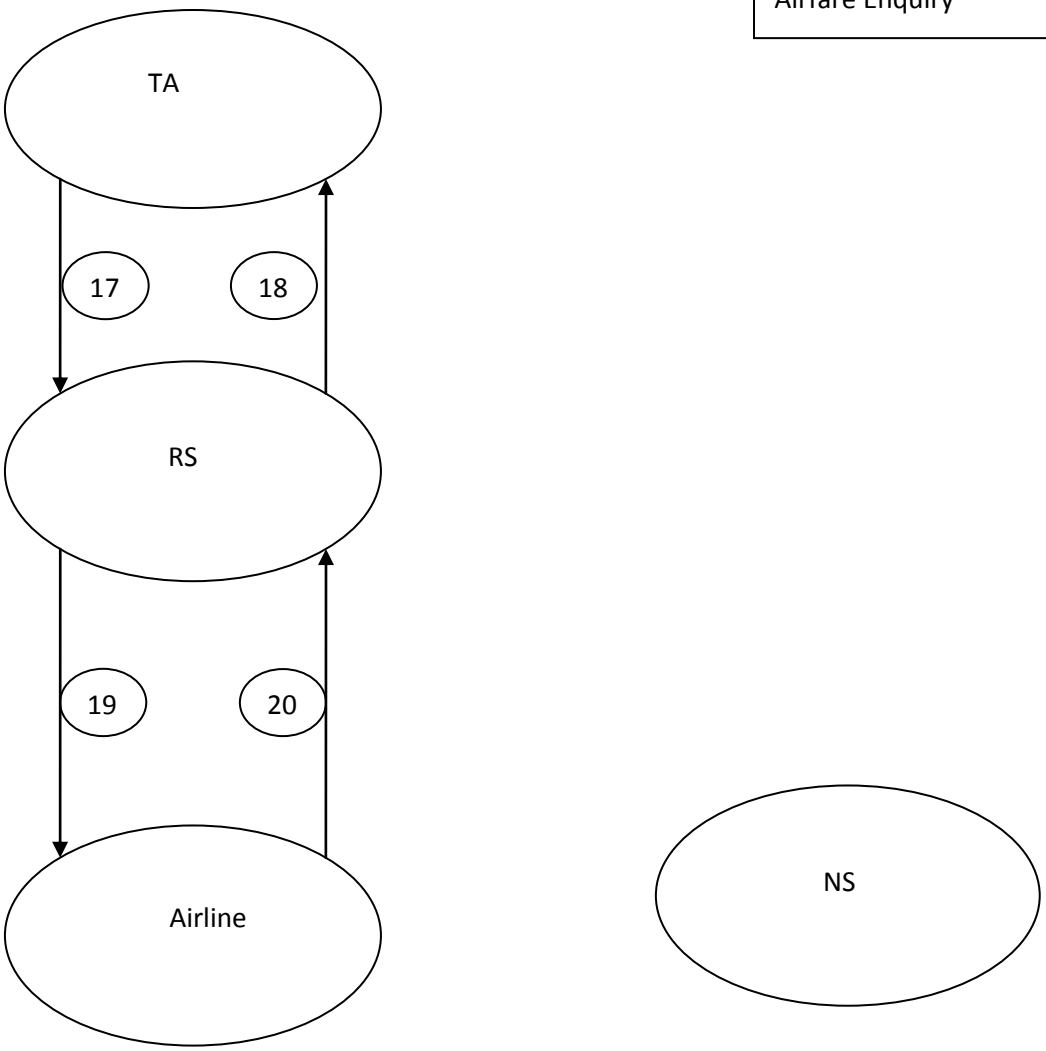
The communication between different processes is annotated as shown in the following pages below. There are 22 arrows representing the communication between different processes to perform different operations. Since there are too many communications, I am representing it in four different figures representing four different operations listed below:

(1) Name Registration
(2) Name Lookup
(3) Airfare Enquiry
(4) Air ticket purchase

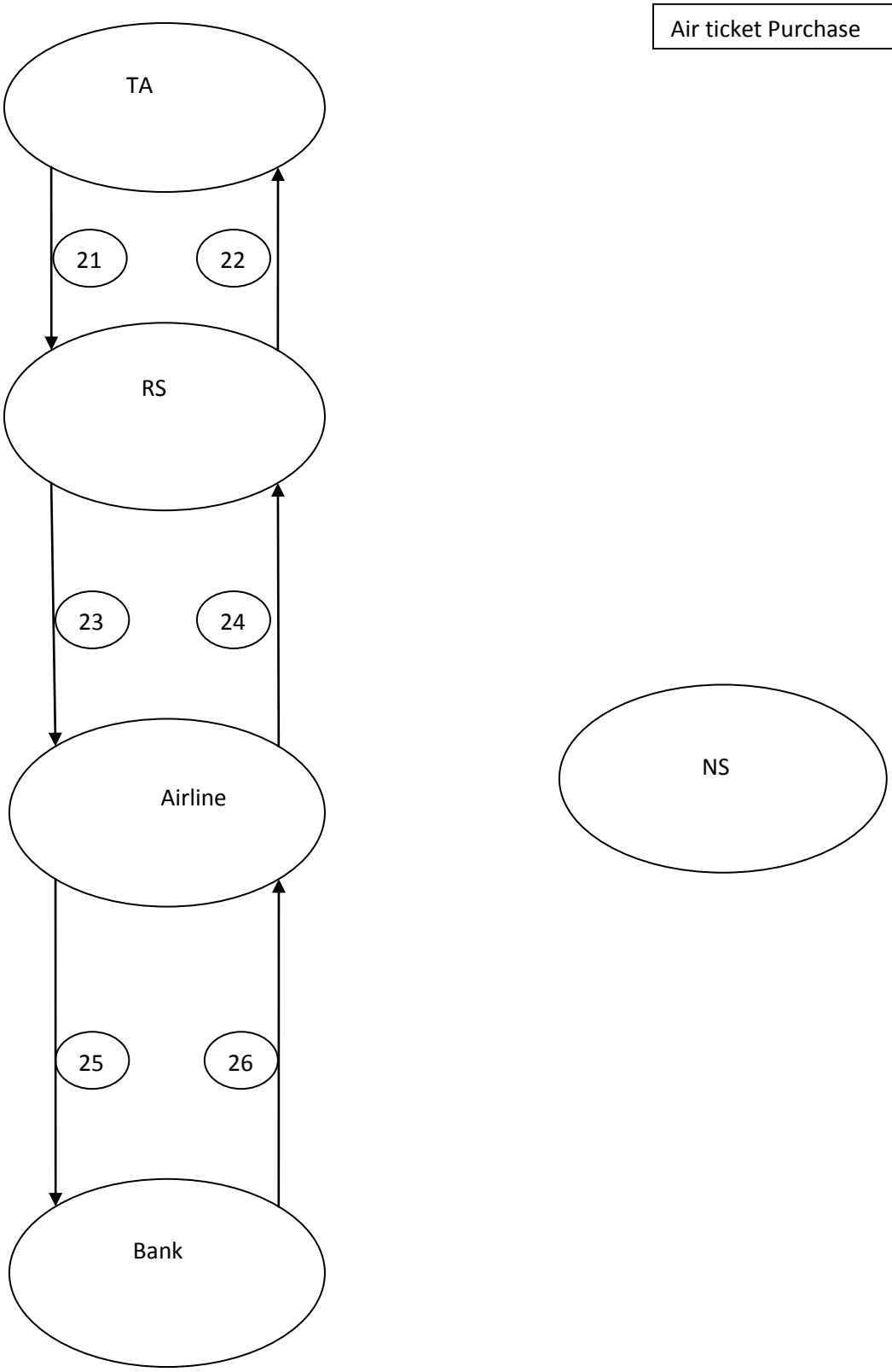

Name Registration
TA
RS
Airline
NS
Bank
1
2
3
4
5
6
7
8


Name Lookup
TA
RS
Airline
NS
Bank
9
10
11
12
13
14
15
16

Airfare Enquiry



Air ticket Purchase



2. Table Showing Communication Primitives

Arrow Number	Sending Process	Send Primitive	Receiving Process	Receive Primitive	Message Format Names used in point 4
1	TA	RPC Call	NS	RPC Accept	A
2	NS	RPC Reply	TA	RPC Call	B
3	RS	RPC Call	NS	RPC Accept	A
4	NS	RPC Reply	RS	RPC Call	B
5	Airline	RPC Call	NS	RPC Accept	A
6	NS	RPC Reply	Airline	RPC Call	B
7	Bank	RPC Call	NS	RPC Accept	A
8	NS	RPC Reply	Bank	RPC Call	B
9	TA	RPC Call	NA	RPC Accept	C
10	NS	RPC Reply	TA	RPC Call	D
11	RS	RPC Call	NS	RPC Server Accept	C
12	NS	RPC Reply	RS	RPC Call	D
13	Airline	RPC Call	NS	RPC Server Accept	C
14	NS	RPC Reply	Airline	RPC Call	D
15	Bank	RPC Call	NS	RPC Server Accept	C
16	NS	RPC Reply	Bank	RPC Call	D
17	TA	RPC Call	RS	Non-Blocking Receive	E
18	RS	Non-Blocking Send	TA	RPC Call	F
19	RS	Non-Blocking Send	Airline	Non-Blocking Receive	E
20	Airline	Non-Blocking Send	RS	Non-Blocking Receive	F
21	TA	RPC Call	RS	Non-Blocking Receive	G
22	RS	Non-Blocking Send	TA	RPC Call	H
23	RS	Non-Blocking Send	Airline	Non-Blocking Receive	G
24	Airline	Non-Blocking Send	RS	Non-Blocking Receive	H
25	Airline	Non-Blocking Send	Bank	Blocking Receive	G
26	Bank	Blocking Send	Airline	Non-Blocking Receive	H

3. Rationale for choosing Communication Primitives

- Every process registers its IP and Port number as Remote Procedure Call. It waits for the response to know whether it has registered successfully. Hence RPC is suitable primitive
- Look-up request from any process to Name Server is also executed as RPC call because the processes cannot proceed further until they get the address and port number. On the other hand Name Server accepts connections to register IP:Port and serve the lookup request for IP:Port

- Travel Agent should not be doing any work after the request is sent to Reservation Service. Hence a RPC Call
- Reservation service can receive requests from any travel agent and also responses from Airlines that needs to be forwarded to appropriate process. Hence a non-blocking receive and send primitives are used.
- Airline receives requests from Reservation Service either for enquiry or Purchase. If purchase request is received it needs to send request to Bank process. Hence a non-blocking send and non-blocking receive are used.
- Bank receives only a purchase request, hence it is waits blocking as it has nothing else to do apart from serving purchase requests. When it receives purchase requests, prints the confirmation number and replies back the sender with confirmation number.

4. Message Formats

Format A

Number of Names	Message Type	IP	Port
Unsigned Integer	Fixed Length String	Variable Length String	Unsigned Integer
4 bytes	8 bytes	7 to 15 bytes	4 bytes

Number of Names -> 3
Message Type -> “REGISTER”
IP address -> 1.1.1.1 or 255.255.255.255
Port address -> Greater than 1024 and less than 65535 as String
Overall size of message is 23 to 31 bytes

Format B

Number of Names	Registration Response
Unsigned Integer	Fixed Size String
4 bytes	6 bytes

Number of Names -> 2
Registration Process -> “SUCCESS” or “FAILURE”
Overall Size of message is 10 bytes

Format C

Number of Names	Message Type	Process Name
Unsigned Integer	Fixed Size String	Variable Size String
4 bytes	6 bytes	1 to 15 bytes

Number of Names -> 2
Request Type -> “Lookup”
Process Name -> Assuming String is between 1 to 15 bytes long
Overall size of message is 11 to 25 bytes

Format D

Number of Names	**IP**	**Port**
Unsigned Integer	Variable Length String	Unsigned Integer
4 bytes	7 to 15 bytes	4 bytes

Number of Names -> 2
IP address can be between 1.1.1.1 and 255.255.255.255
Port address > 1024 and < 65535 as String
Overall size of message is 15 to 23 bytes

Format E

Number of Names	**Message Type**	**Agent**	**Origin Airport**	**Dest Airport**	**Preferred Airline**
Unsigned Integer	Fixed Length String	Fixed Length String	Fixed Length String	Fixed Length String	Fixed Length String
4 bytes	7 bytes	6 bytes	3 bytes	3 bytes	2 bytes

Number of Names -> 5
Message Type -> “Enquiry”
Agent can be “automt” or “manual”
Origin Airport -> BNE, SYD or MEL
Destination Airport -> BNE, SYD or MEL
Preferred Airline -> QF, VA or XX (Representing Both)
Overall size of message is 25 bytes

Format F

Number of Names	**Message Type**	**Agent**	**Flight Number**	**Arrival Time**	**Departure Time**	**Price**	**Flight Number**	**… … …**	**Price**
Unsigned Integer	Fixed Length String	Fixed Length String	Unsigned Integer	Fixed Length String	Fixed Length String	Floating Point	Unsigned Integer	….	Floating Point
4 bytes	15 bytes	6 bytes	4 bytes	5 bytes	5 bytes	4 bytes	4 bytes	….	4 bytes

Number of Names -> Can be between 2 and (2 + Number of flights times 4 fields)
Message Type -> “EnquiryResponse”
Agent can be “automt” or “manual”
Flight Number can be between 1 and 1000
Arrival Time can be between “00:00” and “23:00”
Departure Time can be between “00:00” and “23:00”
Price can be a float point value with 2 digits in decimal
Overall size can be between 10 bytes and (10 + 18 × number of flights)

Format G

Number of Names	**Message Type**	**Agent**	**Airline**	**Fight Number**	**Origin Airport**	**Destination Airport**	**Name**	**Credit Card**
Unsigned Integer	Fixed Length String	Fixed Length String	Fixed Length String	Unsigned Integer	Fixed Length String	Fixed Length String	Variable Length String	Fixed Length String
4 bytes	8 bytes	6 bytes	2 bytes	4 bytes	3 bytes	3 bytes	5 to 10 bytes	16 bytes

Number of Names -> 8
Message Type -> “Purchase”
Agent can be “automt” or “manual”
Airline can be either “QF” or “VA”
Flight Number can be between 1 and 1000
Origin Airport can be BNE, SYD or MEL
Destination Airport can be BNE, SYD or MEL
Name is a simple string
Credit card is 16 bytes in length
Total size of message is 51 to 56 bytes

Format H

Number of Names	**Message Type**	**Agent**	**Confirmation Number**	**Message**
Unsigned Integer	Fixed Length String	Fixed Length String	Unsigned Integer	Variable Sized String
4 bytes	16 bytes	6 bytes	4 bytes	10 to 30 bytes

Number of Names -> 4
Message Type -> PurchaseResponse

Agent can be “automt” or “manual”
Unsigned Integer can be between 0 and 10000. 0 indicates that booking was unsuccessful.
Message gives reason for error or returns a string that booking is confirmed
Overall size of message is 40 to 70 bytes

5. Assumptions

The following are the assumptions made while designing the inter-process communication:

a. All processes are started before the travel agent makes any enquiry.
b. The processes request name server every time when they have to communicate with other processes. This way if any process has restarted then its new address is re-fetched.
c. With regards to Message Format F, if no flight details are found only first two fields will be sent back from Airline to Reservation Service to Travel Agent. The Travel Agent on reading there are only 2 fields infers that there is no flights are found
d. In a few message formats like A,C, E,F, G and H a field type named, “Message Type” is used to identify the type of message and the action the process should take on receiving such a message.
e. Assuming the length of name of person is between 5 to 10 bytes in length