

CS8601 –MOBILE COMPUTING

UNIT 4

MOBILE TRANSPORT AND APPLICATION LAYER

4.5. Wireless transaction Protocol (WTP)

WTP is on top of either WDP or, if security is required, WTLS. WTP has been designed to run on very thin clients, such as mobile phones.

Advantages of WTP:

- Improved reliability over datagram services
- Improved efficiency over connection-oriented services
- Support for transaction-oriented services such as web browsing.

Three classes of WTP transaction service:

- Class 0 provides unreliable message transfer without any result message.
- Classes 1 provides reliable message transfer without any result message.
- Class 2 provides reliable message transfer with one reliable result message.

WTP achieves reliability using:

- Duplicate removal
- Retransmission
- Acknowledgements
- Unique transaction identifiers.

No class requires any connection set-up or tear-down phase. This avoids unnecessary overhead on the communication link. Allows for

- Asynchronous transactions
- Abort of transactions
- Concatenation of messages
- Report success or failure of reliable messages.

The three service primitives offered by WTP are:

- TR-Invoke - to initiate a new transaction
- TR-Result - to send back the result of a previously initiated transaction
- TR-Abort - to abort an existing transaction.

Types of WTP PDU:

- ✓ Invoke PDU – used to convey a request from an initiator to a responder
- ✓ ACK PDU – used to acknowledge an Invoke or Result PDU
- ✓ Result PDU – used to convey response of the server to the client
- ✓ Abort PDU – used to abort a transaction
- ✓ Segmented invoke PDU and segmented result PDU used for segmentation and reassembly
- ✓ Negative acknowledgment PDU – used to indicate that some packets did not arrive

WTP Class 0 : Unreliable Message Transfer without result message

- In this class the responder does not ACK & initiator does not perform any retransmission.
- The transaction is stateless and cannot be aborted.
- Requested with TR-Invoke.req primitive.
- Parameters are: (SA, SP, DA, DP, A, UD, C=0, H) SA - source address

SP - source port

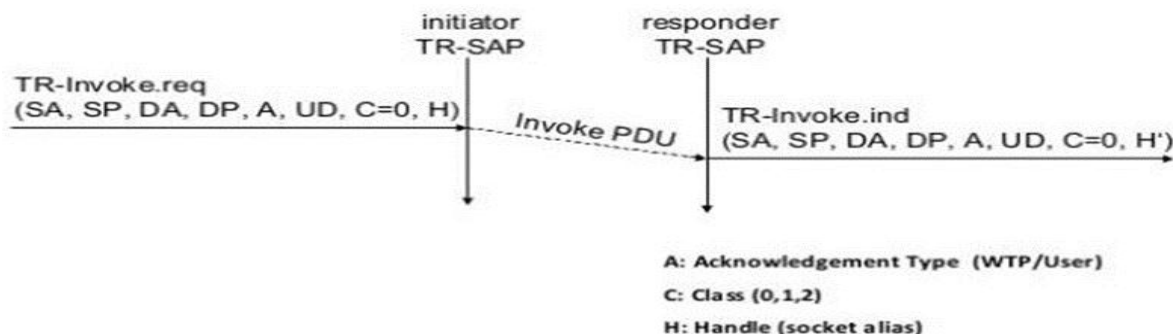
DA - destination address DP - destination port

A - acknowledgement flag, if the responder WTP should generate an ACK or if a user acknowledgement is used.

UD - User data

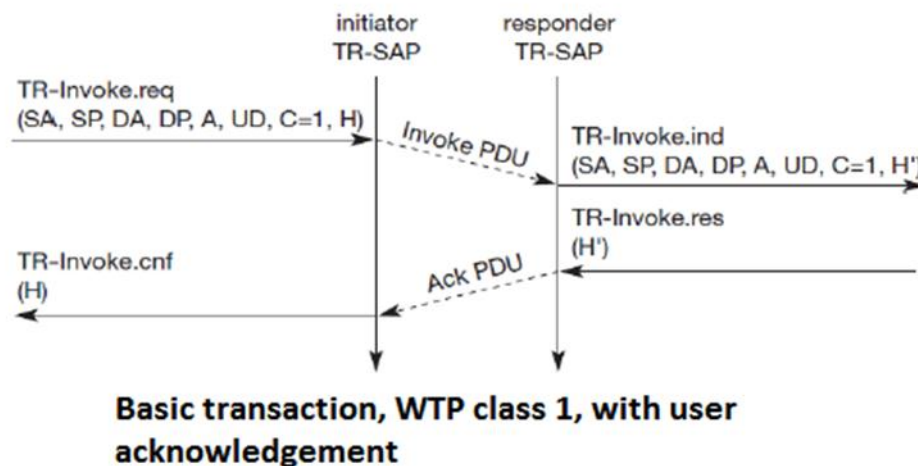
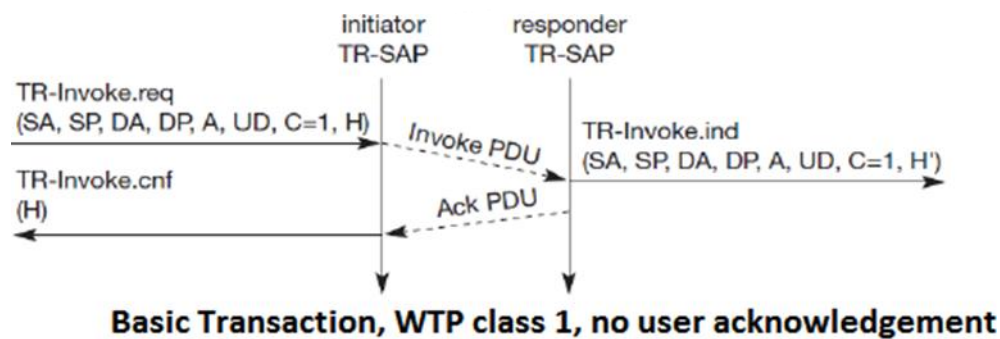
C - class type which is 0 for this class.

H - handle simple index to uniquely identify the transaction



WTP Class 1 : Reliable Message Transfer without result message

- Sender send a TR-Invoke.req
- Parameters are: (SA, SP, DA, A, UD, C=1, H)
- C is class type which is 1 for this class.
- Responder signals the incoming TR-Invoke.ind & ACK automatically
- Sender on receipt of ACK will close the connection.
- Responder maintains the connection for sometime in case it receives the duplicate TR-Invoke.req indicating the loss of ACK.



WTP Class 2 : Reliable Message Transfer with one result message

- Reliable request/respond transaction.
- Depending on user requirements, many different scenarios are possible for initiator/responder interaction

WTP class 2 transaction, No user Ack & No hold on:

1. Initiator requests the service using TR-Invoke.req and the WTP entity sends the invoke PDU to the responder.

2. Responder request with the TR-Invoke.ind.
3. The responder sent back the result PDU to the initiator using TR- Result.req.
4. The initiator indicate the successful transmission of the invoke message and the result with the two service primitives:
 - ✓ TR-Invoke.cnf
 - ✓ TR-Result.ind.
5. A user respond with TR-Result.res.
6. An acknowledgement PDU is then generated which finally triggers the TR-Result.cnf primitive on the responder.

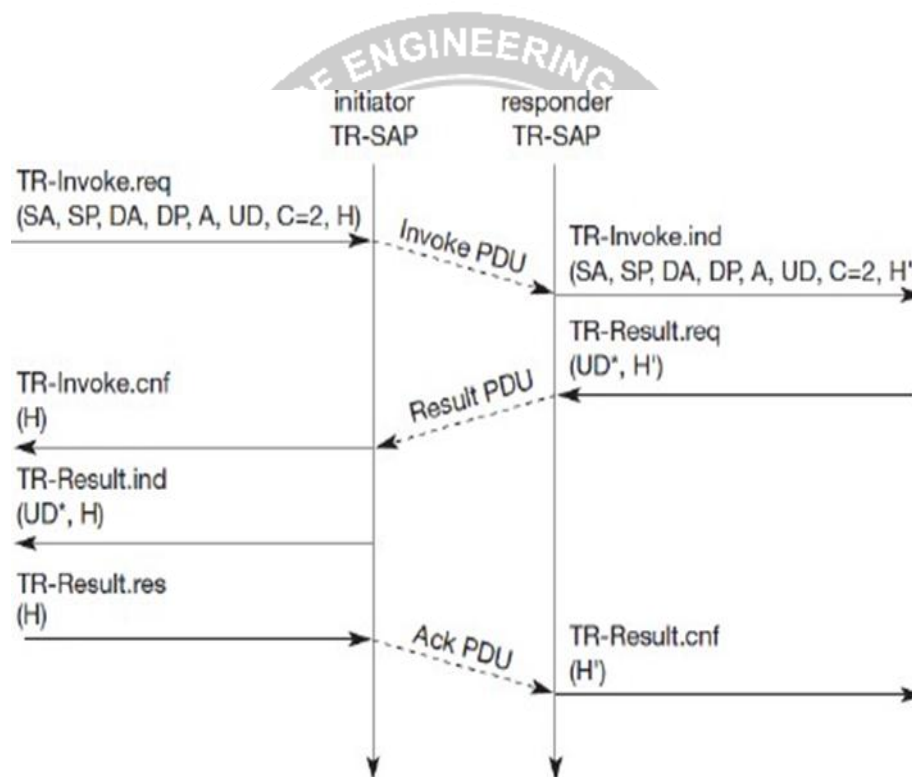
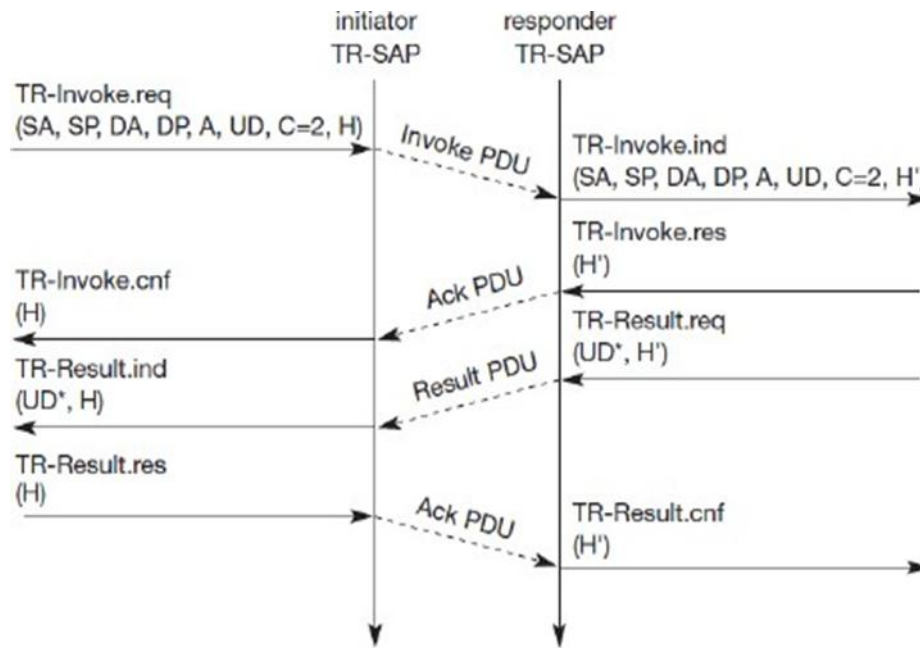


Fig. Basic transaction of class 2 without-user acknowledgement

WTP class 2 transaction, user Ack:

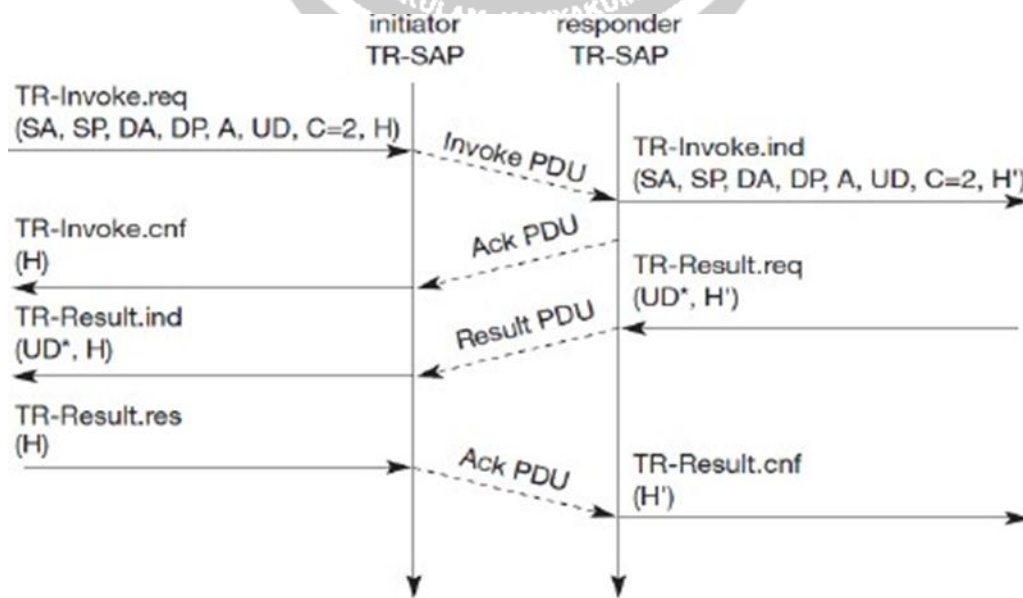
1. The responder explicitly responds to the Invoke PDU using the TR- Invoke.res.
2. Then the initiator triggers the TR-Invoke.cnf via an Ack PDU.



Basic transaction, WTP class2, with user Ack

WTP class 2 transaction, hold on & no user Ack:

If the calculation of the result takes some time, the responder can put the initiator on "hold on" to prevent a retransmission.



WTP class 2 transaction with "hold on", no user Ack