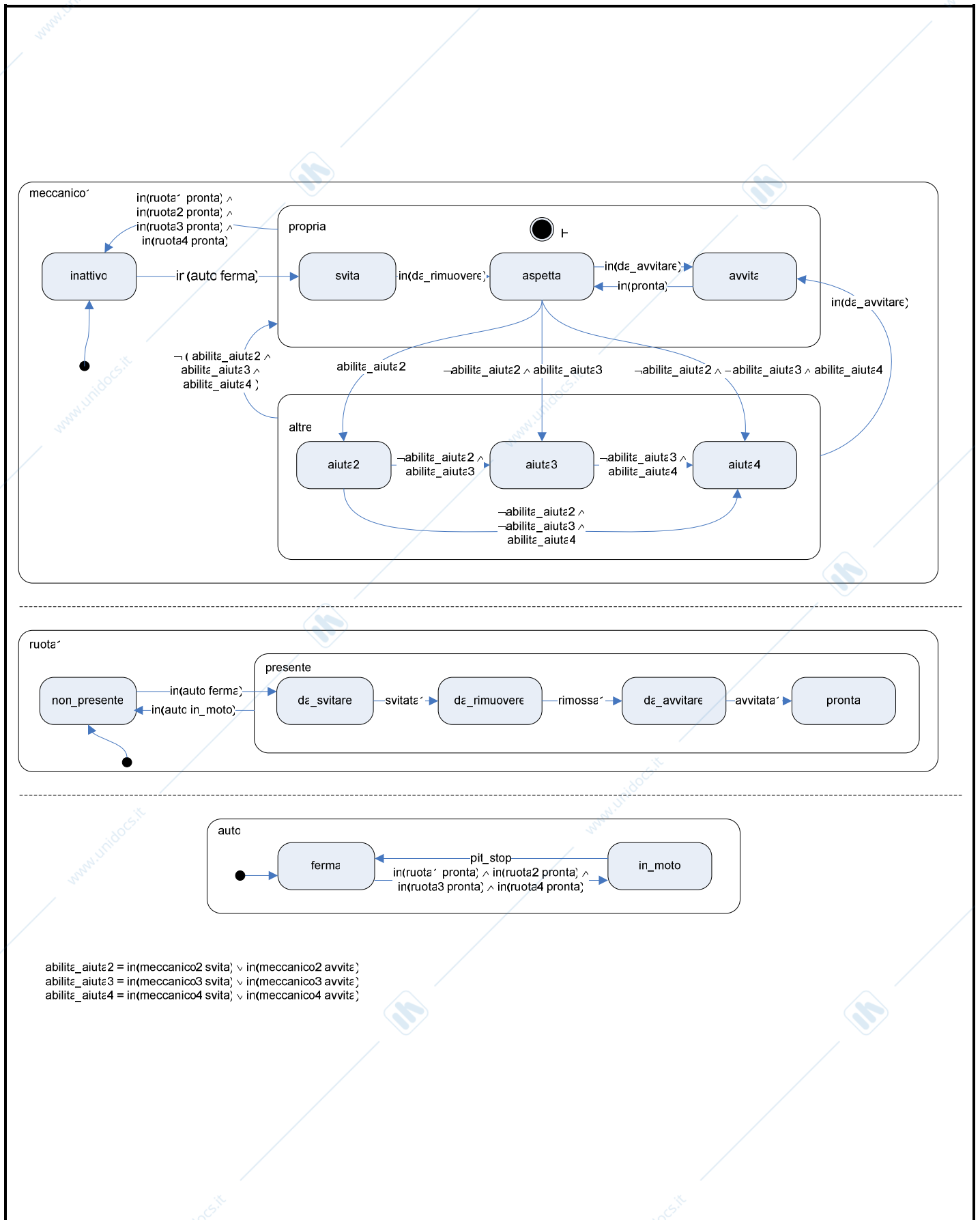


| | | | |
|---|---|----------------|-----------|
|  | Politecnico di Milano Facoltà di Ingegneria dell'Automazione INFORMATICA INDUSTRIALE Appello 06 marzo 2009 | COGNOME E NOME | |
| | RIGA | COLONNA | MATRICOLA |

- Il presente plico pinzato, composto di quattro fogli (fronte/retro), deve essere debitamente compilato con cognome, nome, numero di matricola, posizione durante lo scritto, e deve essere firmato.
- I compiti non compilati, non firmati o con fogli mancanti non saranno considerati validi e quindi non saranno corretti.
- Sarà valutato solo quanto scritto su questi fogli.
- Non è consentito consultare testi né appunti.
- Sul tavolo non devono essere presenti telefoni cellulari, né astucci, né custodie di altro tipo.
- Le risposte devono essere scritte negli appositi riquadri, qualsiasi testo esterno a tali riquadri non verrà preso in considerazione.
- Se lo spazio per la soluzione degli esercizi non fosse sufficiente, si può scrivere sull'ultimo foglio.

FIRMA

Esercizio 1 (10 punti). Si vuole modellare tramite il formalismo degli StateCharts il coordinamento di un gruppo di meccanici durante il pit-stop di un'auto da corsa. Un'auto può essere *ferma* o *in_movimento* (l'evento *pit_stop* attiva la transizione da *in_movimento* a *ferma*, la transizione inversa è attivata quando tutte le gomme sono state cambiate). Una volta che una macchina si ferma, per ogni gomma si attiva un meccanico (*meccanicoi*) che prima svita e poi avvita la gomma. Una volta svitata la gomma, un ulteriore meccanico (il cui comportamento non è da modellare) si occupa di rimuovere la gomma che è stata svitata e montare la gomma nuova. Ogni gomma può trovarsi in: *da_svitare*, *da_rimuovere*, *da_avvitare*, *pronta*. Le transizioni sono attivate dai seguenti eventi: *svitata*, *rimossa*, *avvitata*. La particolarità del problema è che ogni meccanico addetto a svitare/avvitare una specifica gomma può aiutare i meccanici addetti a svitare/avvitare le altre gomme. (L'assunzione è che meccanici diversi possono impiegare tempi diversi per avvitare/svitare e rimuovere le gomme.) Nello specifico: un meccanico può trovarsi in *svita*, *aspetta*, e *avvita*. In *aspetta* il meccanico sta aspettando che la propria gomma venga rimossa oppure ha finito di avvitare la propria gomma e sta aspettando che tutte le altre gomme siano state cambiate. In questo stato il *meccanicoi* può aiutare il *meccanicoj* che in quell'istante sta svitando o avvitando la propria gomma *j* (indicare gli stati come: *aiutaj*, *aiutak*, *aiutal*, dove *j*, *k*, e *l* sono indici). Inoltre, se il *meccanicoi* sta aiutando un altro meccanico e la gomma *i* è stata rimossa, allora il *meccanicoi* va subito ad avvitare quest'ultima gomma, indipendentemente da cosa stia facendo. La macchina ripartirà quando tutte le gomme sono state cambiate.



Esercizio 2 (9 punti). Siano dati i seguenti processi aperiodici J_1 - J_5 riportati in tabella

| | J_1 | J_2 | J_3 | J_4 | J_5 |
|-------|-------|-------|-------|-------|-------|
| a_i | 0 | 2 | 3 | 1 | 5 |
| C_i | 2 | 1 | 2 | 3 | 2 |
| d_i | 11 | 6 | 10 | 7 | 10 |

Si supponga che non sia possibile fare preemption e si voglia schedulare il problema descritto sopra. Si dica perché non è utilizzabile EDF. Si scheduli il problema con Spring, senza forward checking e senza nodi di backtracking con euristica $h_i = d_i$ e successivamente con $h_i = d_i - C_i$. Si riportino gli alberi prodotti e si dica quale delle due schedulazioni minimizza L_{max} .

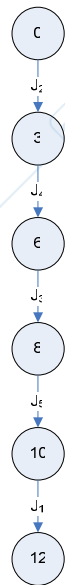
Si si consideri poi il problema di schedulazione costituito dai primi tre processi aperiodici dati sopra e i seguenti due processi periodici.

| | τ_1 | τ_2 |
|----------|----------|----------|
| ϕ_i | 0 | 1 |
| C_i | 1 | 2 |
| T_i | 3 | 7 |

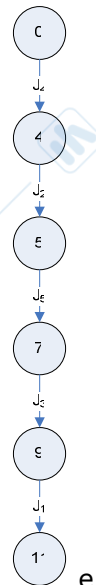
Si produca la schedulazione mediante EDF.

Infine, si consideri il problema di schedulazione costituito dai primi due processi aperiodici descritti sopra e i due processi periodici. Si assuma che i processi aperiodici siano soft real-time. Si dica perché EDF non è più l'algoritmo appropriato per questo problema. Si scheduli il problema con EDF+TBS* (si riporti solo la soluzione grafica segnalando la deadline iniziale e finale assegnate ad ogni processo aperiodico).

Riporto le schedulazioni prodotte con Spring.

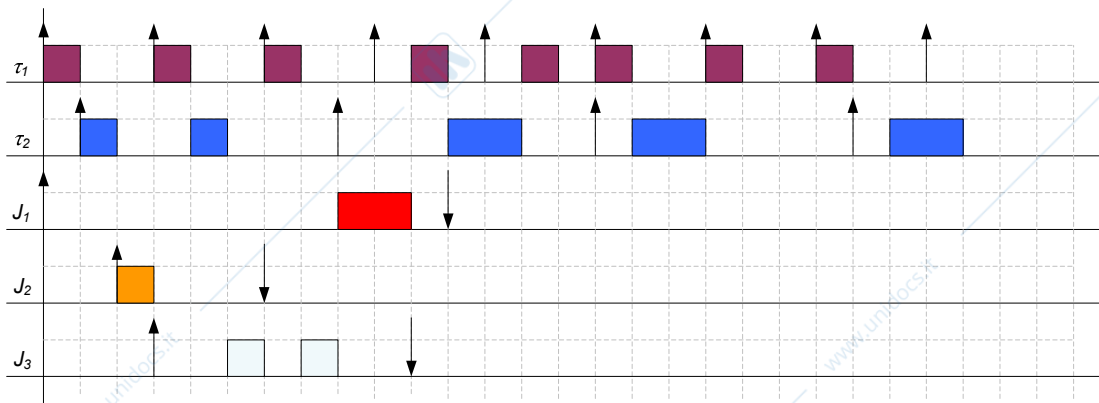


La prima schedulazione è: $L_{max} = 1$

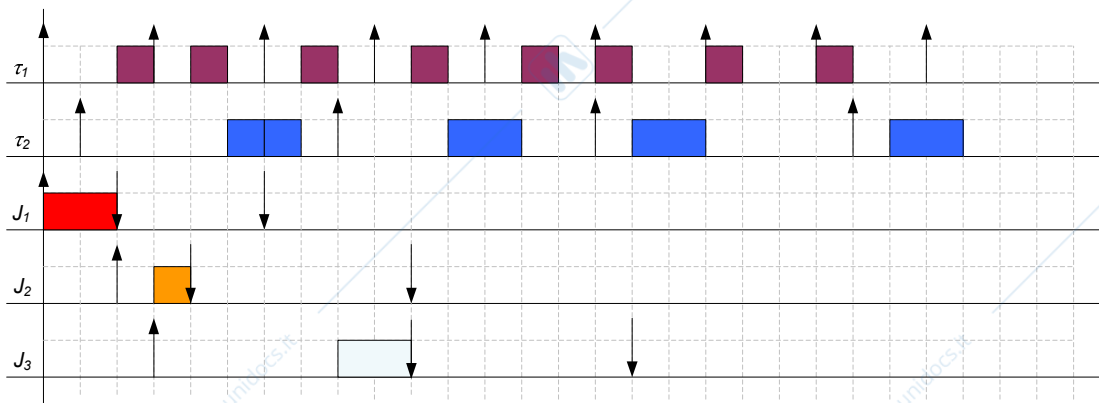


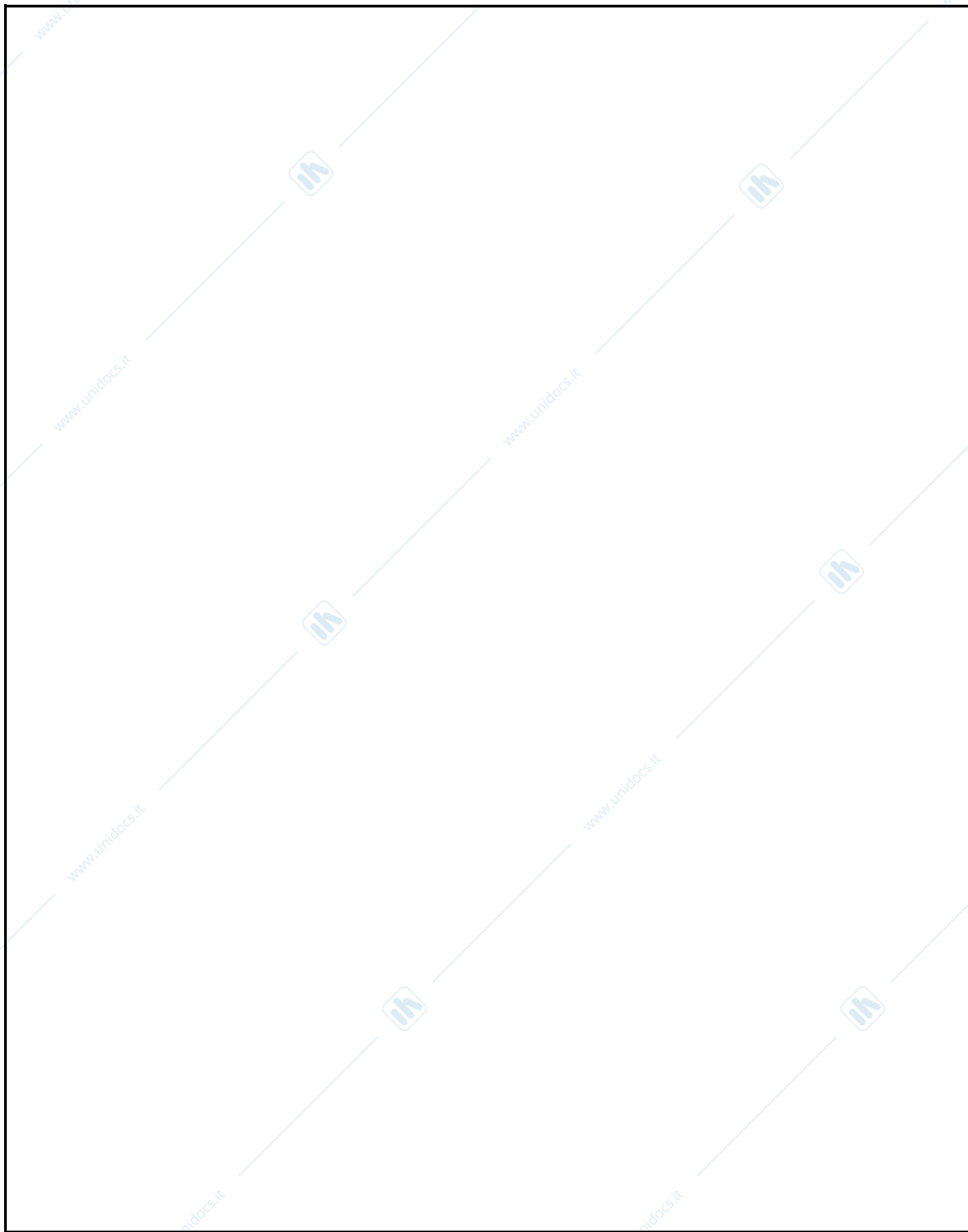
La seconda schedulazione è: $L_{max} = 0$.

Riporto la schedulazione prodotta da EDF.



Riporto la schedulazione prodotta da TBS*.





Esercizio 3 (6 punti). A riguardo del problema della schedulazione real-time di processi periodici, si risponda alle seguenti domande.

- Si dica cos'è Rate Monotonic?
- Si dica se è ottimo e a che tipo di ottimalità si riferisce?
- Riportare lo scheletro della dimostrazione di ottimalità di Rate Monotonic (segnalando ipotesi e tesi).

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Esercizio 4 (6 punti). Con riferimento alle reti di calcolatori, si risponda alle seguenti domande:

- Di cosa si occupa il livello di trasporto all'interno della pila ISO/OSI?
- Come viene gestita la comunicazione dai protocolli UDP e TCP?
- Come viene ricostruito un pacchetto dai protocolli UDP e TCP?

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari