Efficient Multi-Modal Route Planning

Dirk Kienle

Master Thesis

June 14th, 2012

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Introduction

UNI FREIBURG

- Computing shortest paths involving different modes of transportation

Introduction

- Computing shortest paths involving different modes of transportation
  - combine road and transit network
    - walk & use public transportation (bus, subway)

Introduction

- Computing shortest paths involving different modes of transportation
  - combine road and transit network
    - walk & use public transportation (bus, subway)
  - input: source & target address, date, departure time

Introduction

- Computing shortest paths involving different modes of transportation
  - combine road and transit network
    - walk & use public transportation (bus, subway)
  - input: source & target address, date, departure time
  - output: optimal paths that show us what roads to use & which vehicles to take

Introduction

- Computing shortest paths involving different modes of transportation
  - combine road and transit network
    - walk & use public transportation (bus, subway)
  - input: source & target address, date, departure time
  - output: optimal paths that show us what roads to use & which vehicles to take
  - more than one solution (bi-criteria optimization)
    - minimize travel time & # transfers + [walking between stations]

Overview

UNI FREIBURG

- Constructing Transit & Road Network
- Combining Road & Transit Network
- Bi-criteria Optimization
- Experiments
- Conclusion

Constructing the Transit Network

UNI FREIBURG

- Data: GTFS (General Transit Feed Specification)

Constructing the Transit Network

UNI FREIBURG

- Data: GTFS (General Transit Feed Specification)
  - stops/stations (Freiburg Hbf)

Constructing the Transit Network

UNI FREIBURG

- Data: GTFS (General Transit Feed Specification)
  - stops/stations (Freiburg Hbf)
  - trips (bus 11: Messe Freiburg @ 17:27 to Freiburg Hbf @ 17:37)

Constructing the Transit Network

UNI FREIBURG

- Data: GTFS (General Transit Feed Specification)
  - stops/stations (Freiburg Hbf)
  - trips (bus 11: Messe Freiburg @ 17:27 to Freiburg Hbf @ 17:37)
  - routes (all journeys of bus 11)

Constructing the Transit Network

UNI FREIBURG

- Data: GTFS (General Transit Feed Specification)
  - stops/stations (Freiburg Hbf)
  - trips (bus 11: Messe Freiburg @ 17:27 to Freiburg Hbf @ 17:37)
  - routes (all journeys of bus 11)
  - service days (weekdays, weekends)

Constructing the Transit Network

UNI FREIBURG

- Data: GTFS (General Transit Feed Specification)
  - stops/stations (Freiburg Hbf)
  - trips (bus 11: Messe Freiburg @ 17:27 to Freiburg Hbf @ 17:37)
  - routes (all journeys of bus 11)
  - service days (weekdays, weekends)
- Time-Expanded vs.Time-Dependent Approach

Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3

Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

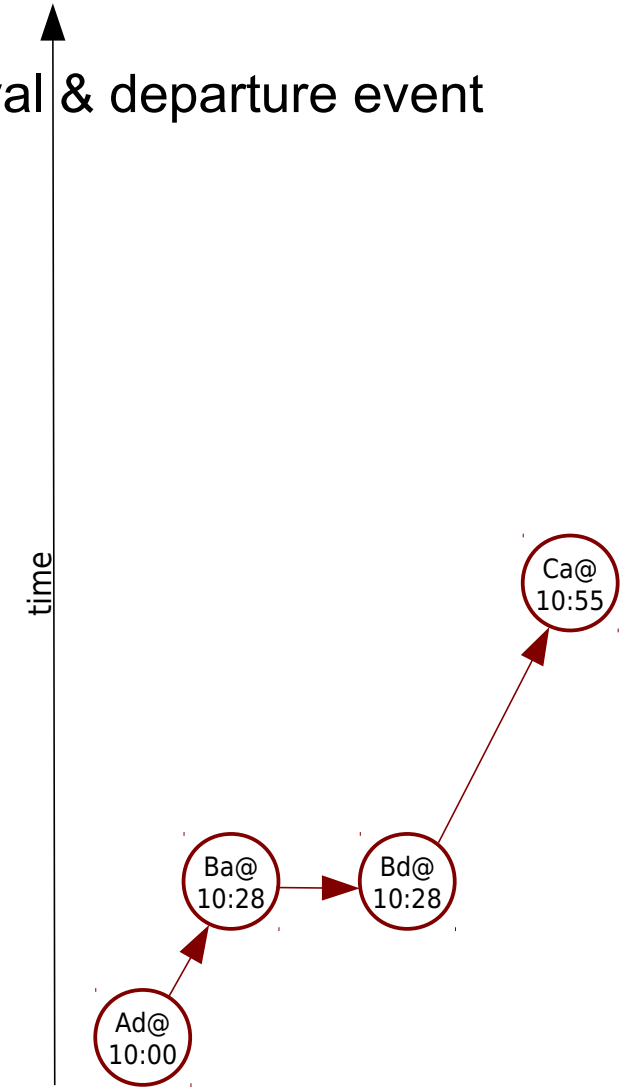
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3

Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

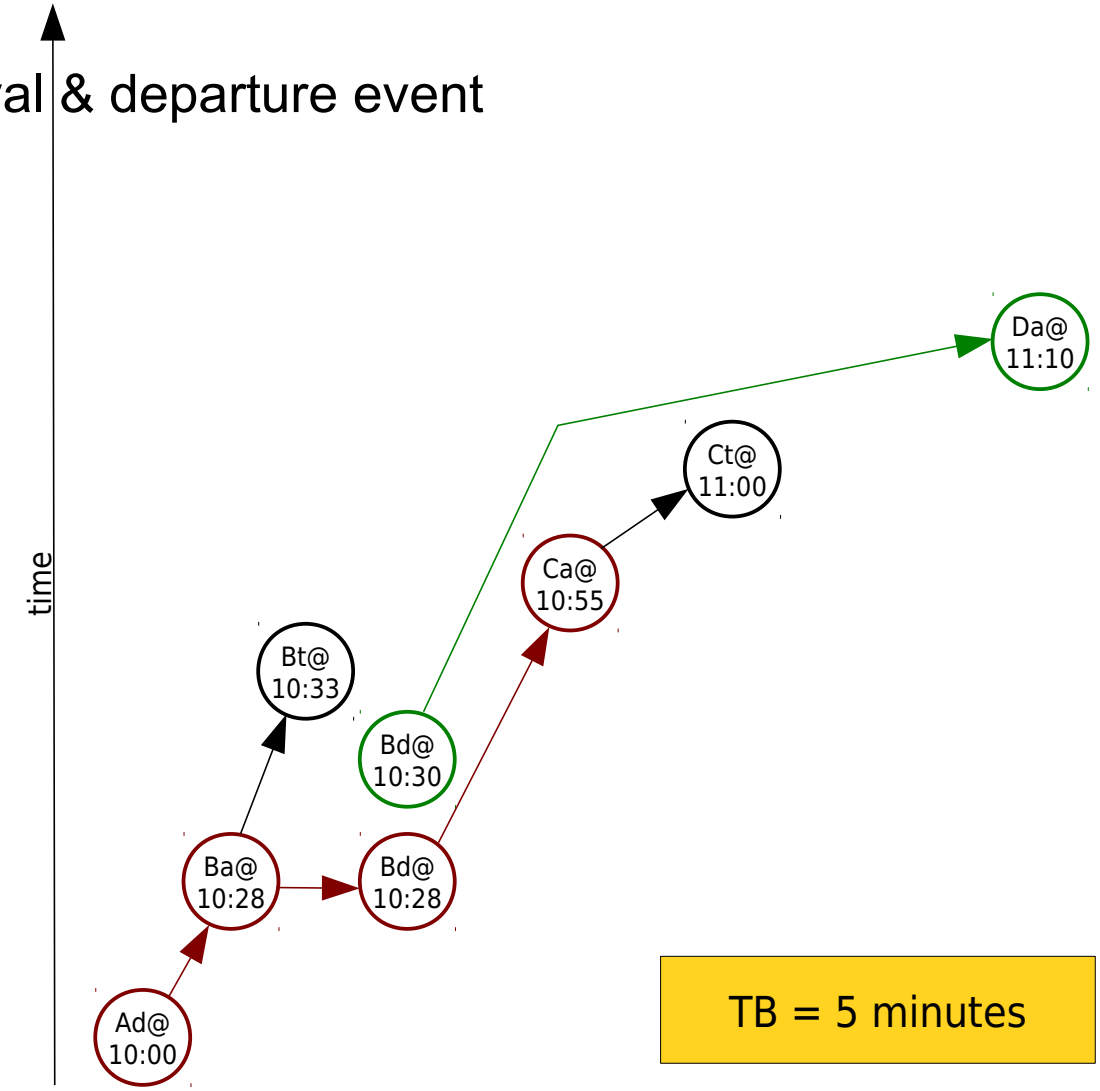
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

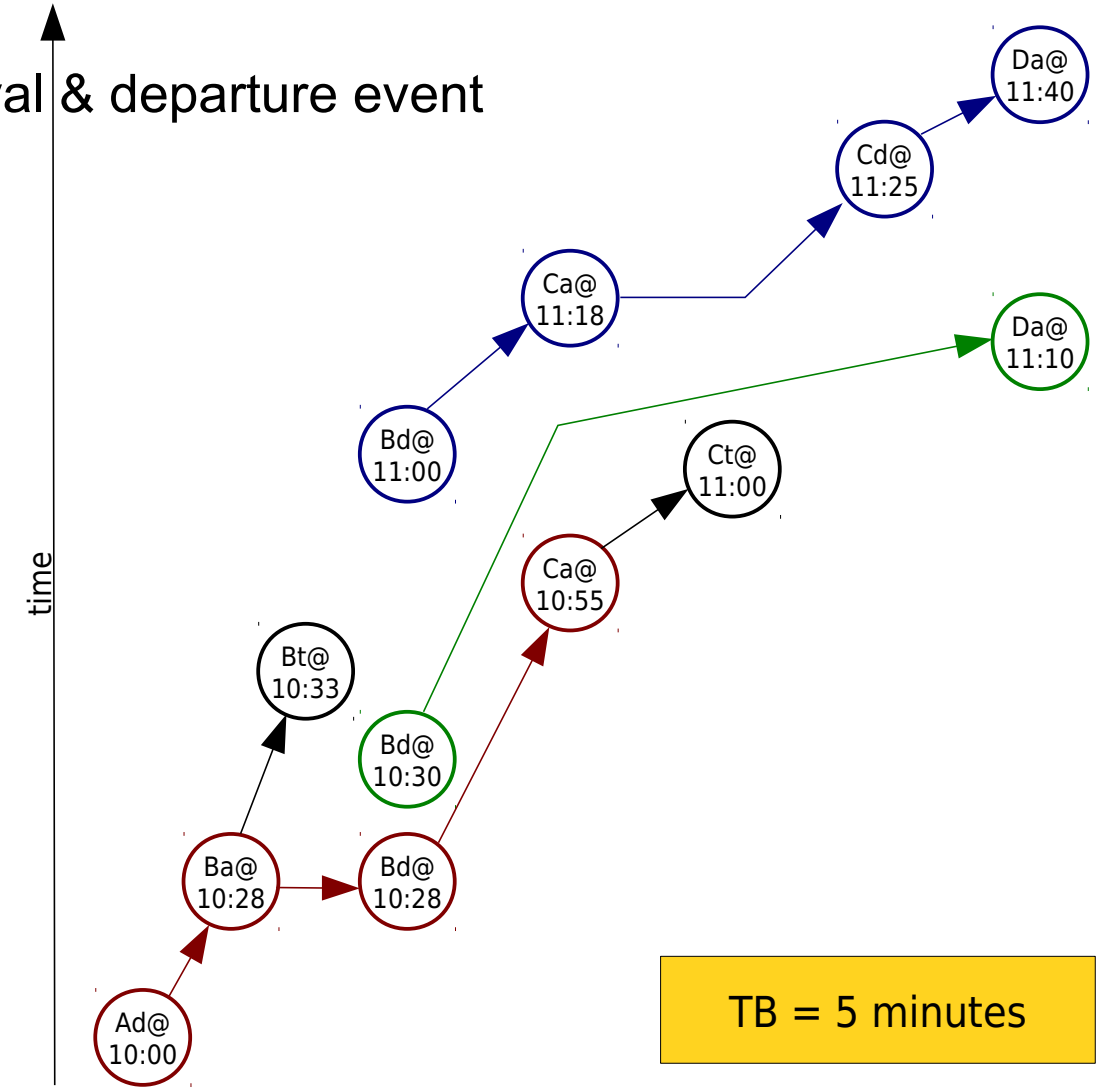
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

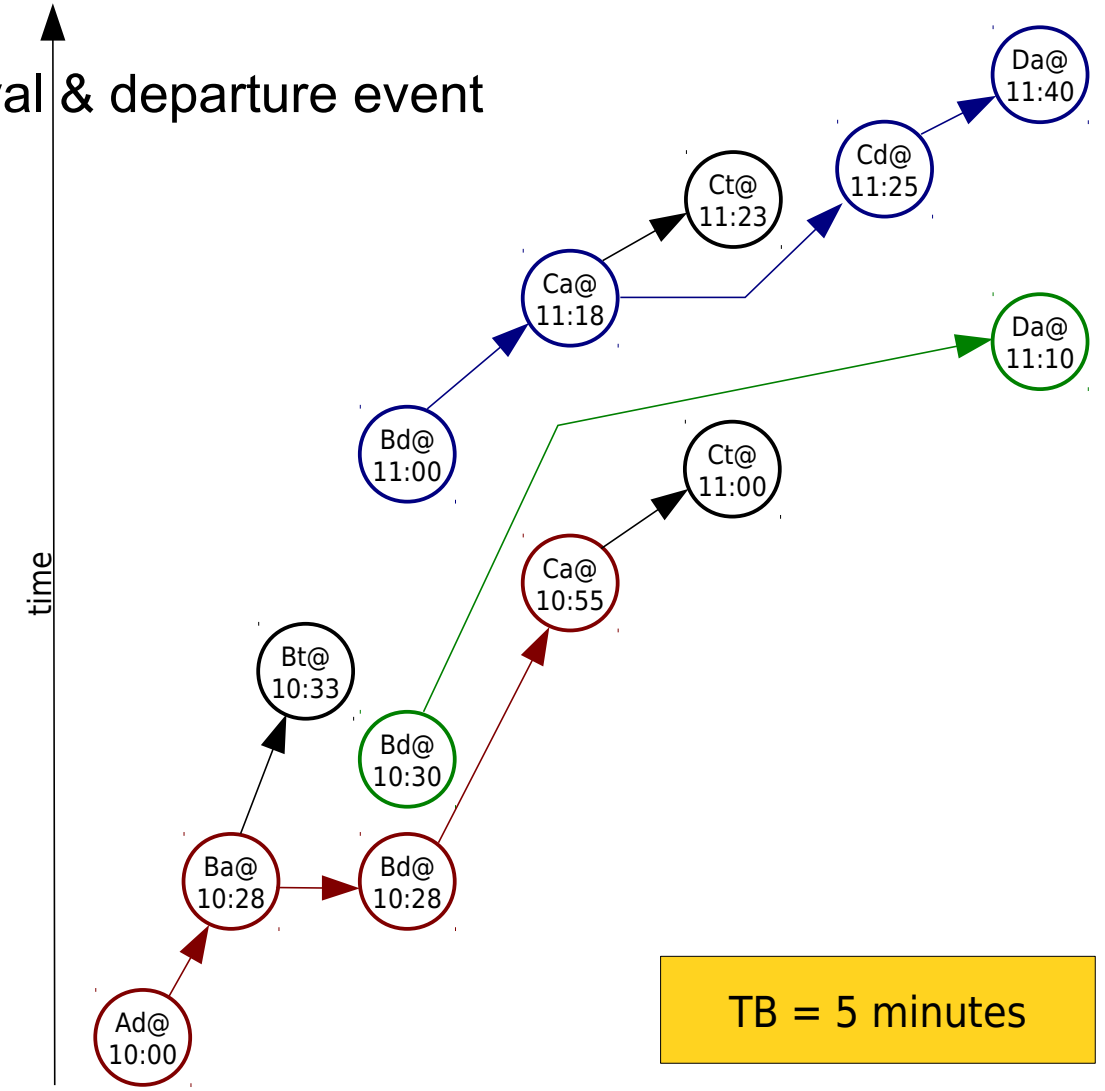
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

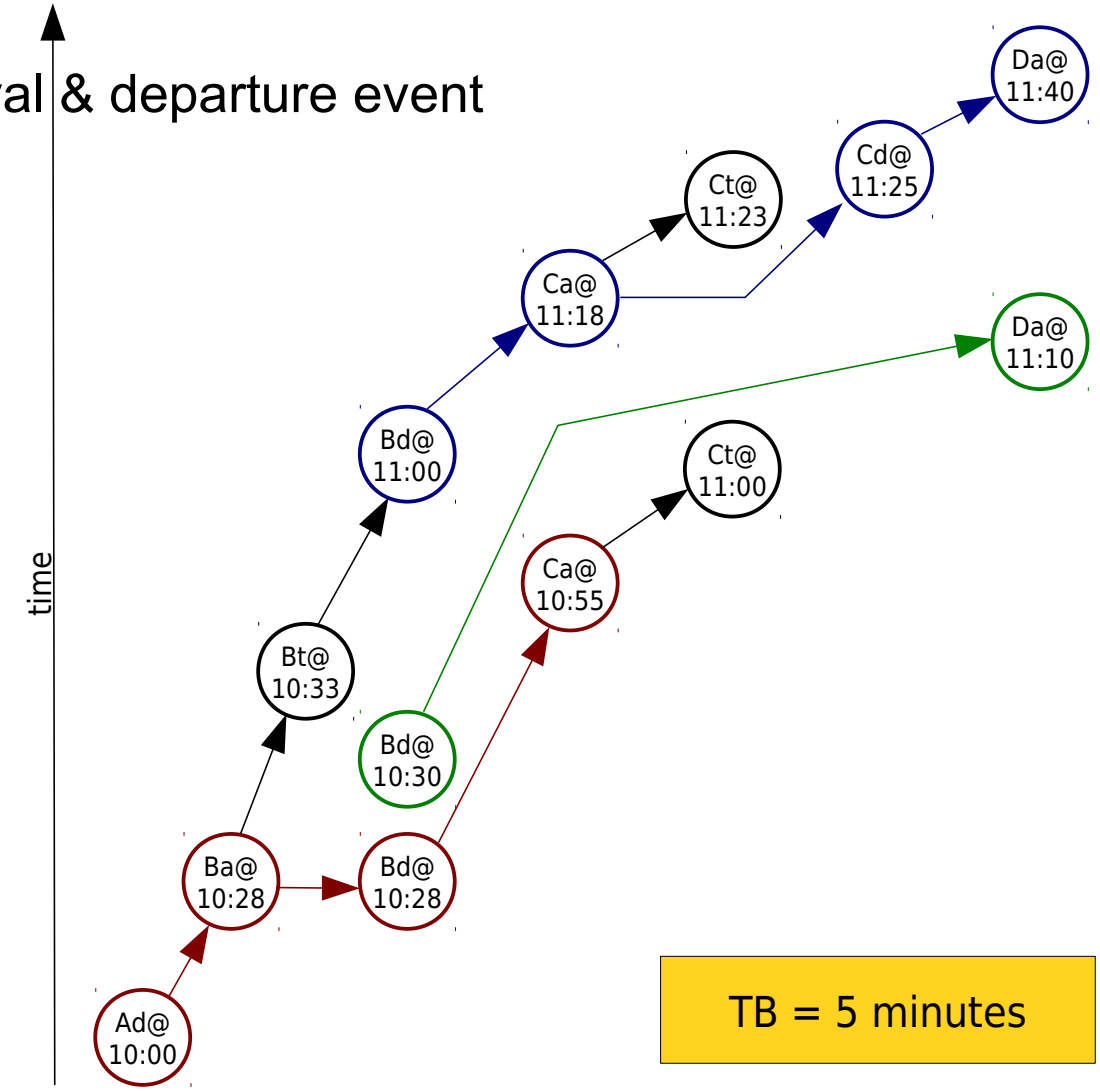
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

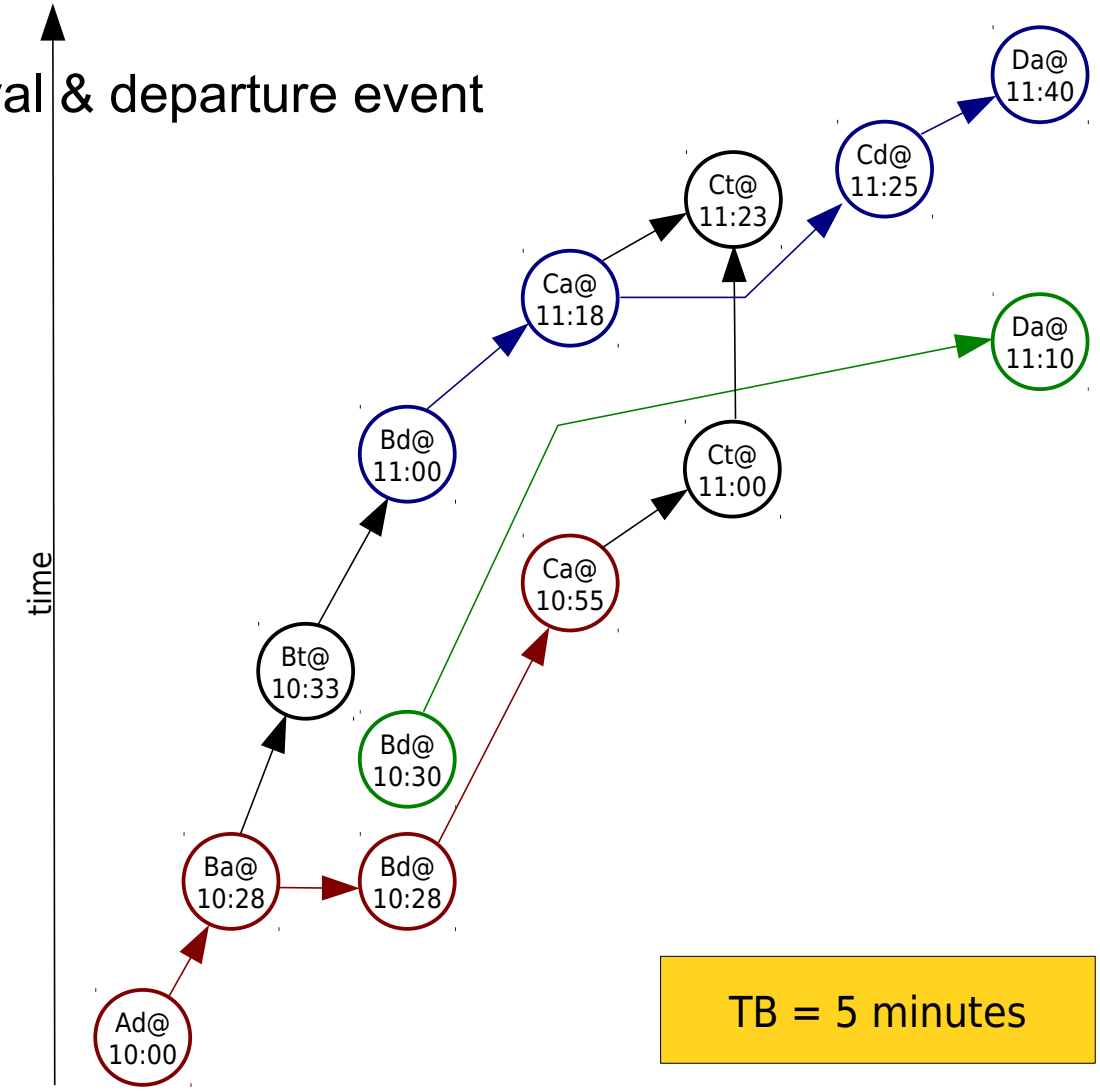
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



Time-Expanded Network

- One node per arrival & departure event

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

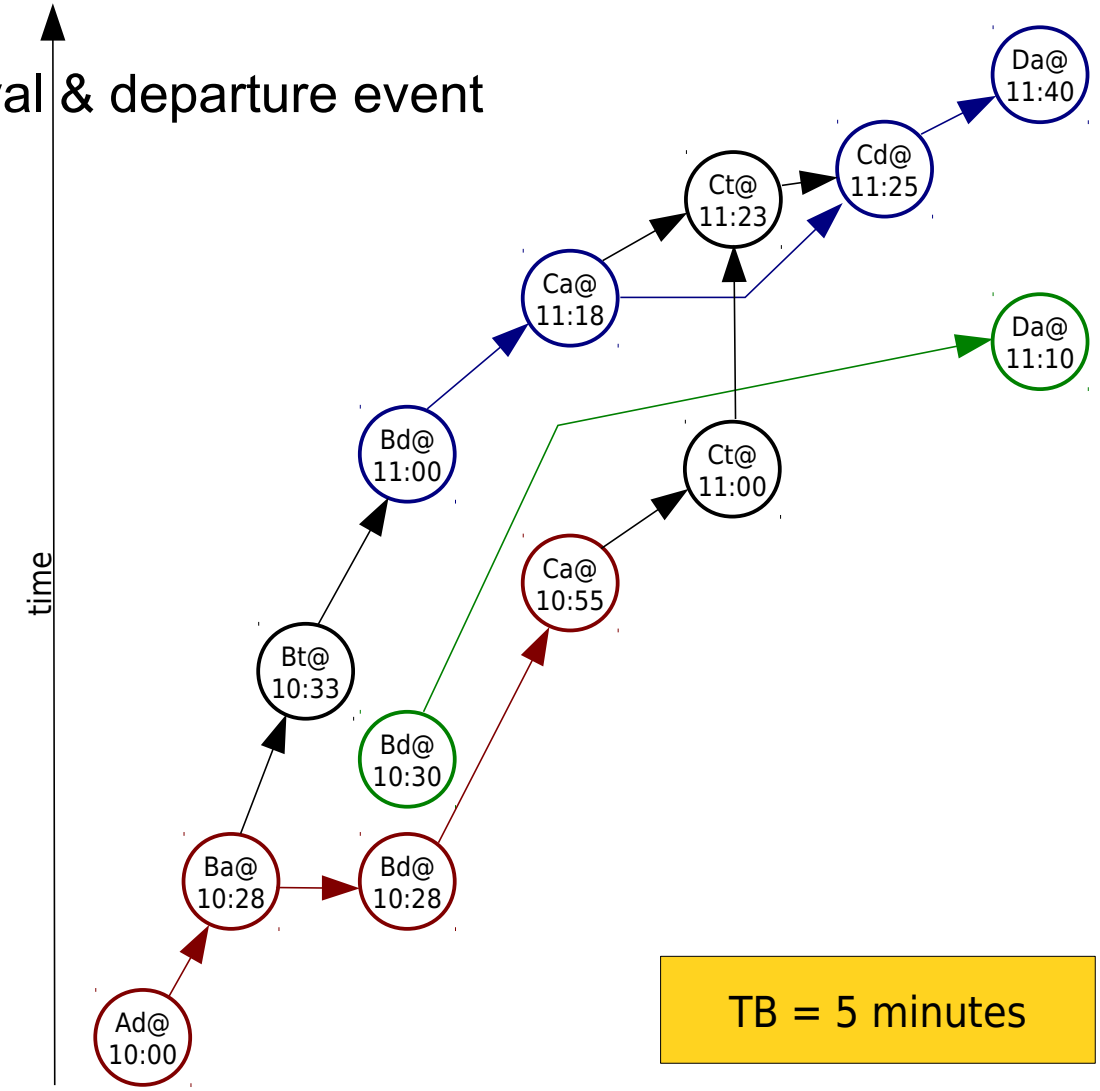
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



Time-Dependent Network

- One node per station

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3

Time-Dependent Network

- One node per station

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

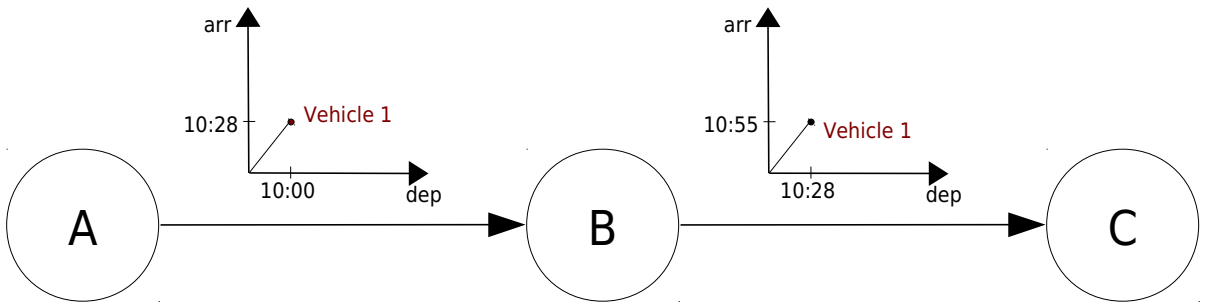
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



Time-Dependent Network

- One node per station

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

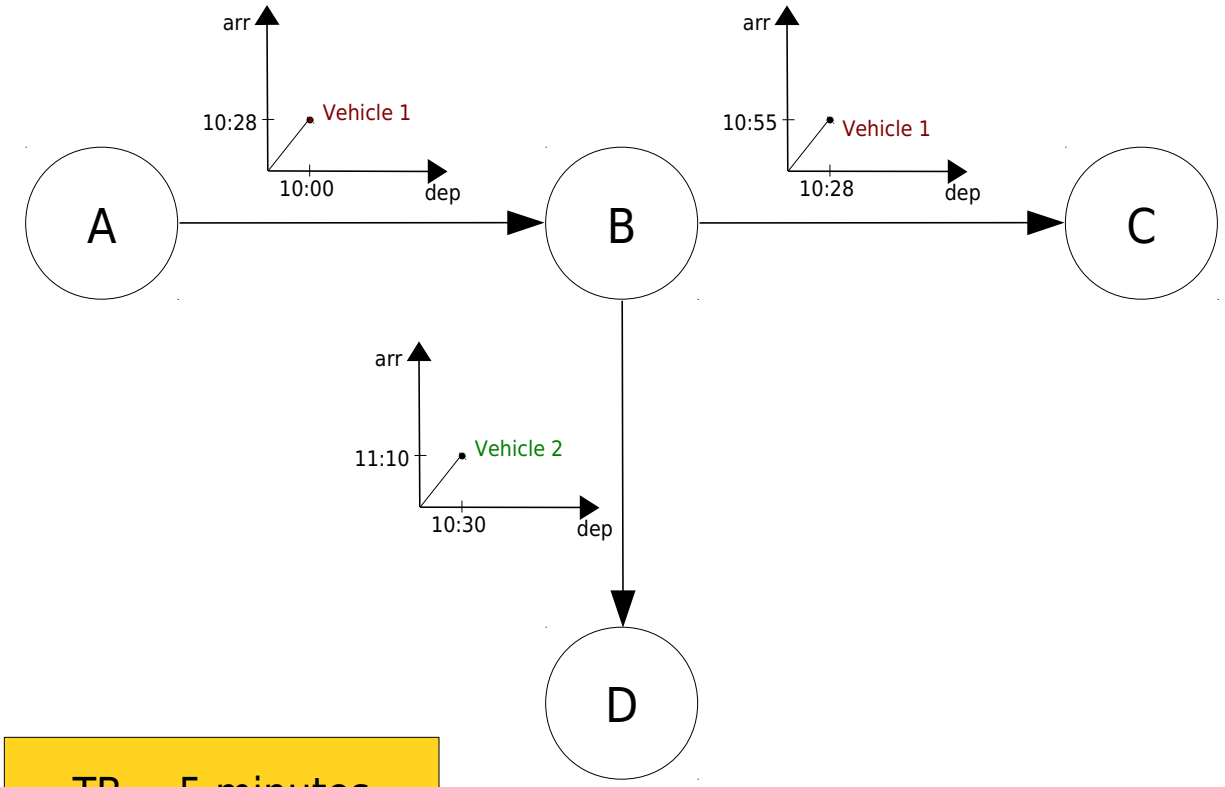
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



TB = 5 minutes

Time-Dependent Network

- One node per station

station	time
A	10:00 (dep)
B	10:28 (arr) 10:28 (dep)
C	10:55 (arr)

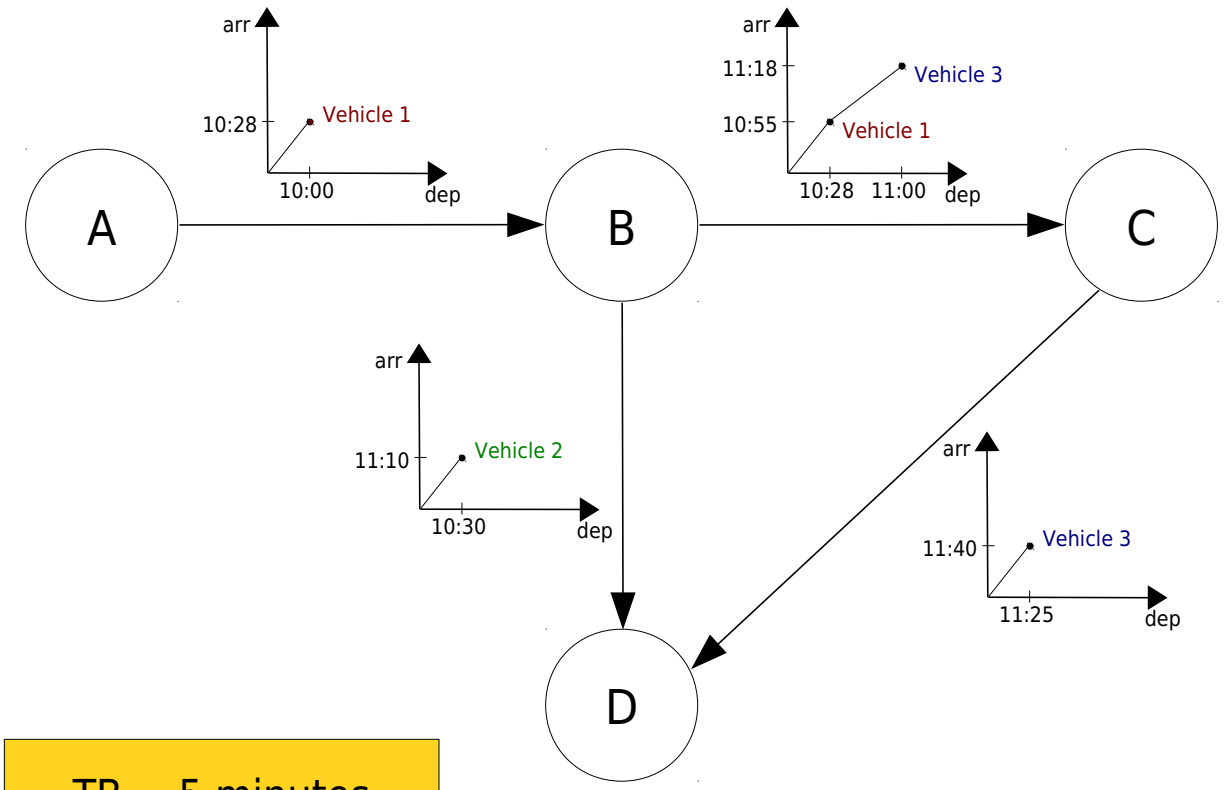
Vehicle 1

station	time
B	10:30 (dep)
D	11:10 (arr)

Vehicle 2

station	time
B	11:00 (dep)
C	11:18 (arr) 11:25 (dep)
D	11:40 (arr)

Vehicle 3



TB = 5 minutes

Constructing the Road Network

UNI FREIBURG

- Each junction is a node in the graph

Constructing the Road Network

UNI FREIBURG

- Each junction is a node in the graph
- Edges are the road segments

Constructing the Road Network

UNI FREIBURG

- Each junction is a node in the graph
- Edges are the road segments
- Cost of edge = walking time

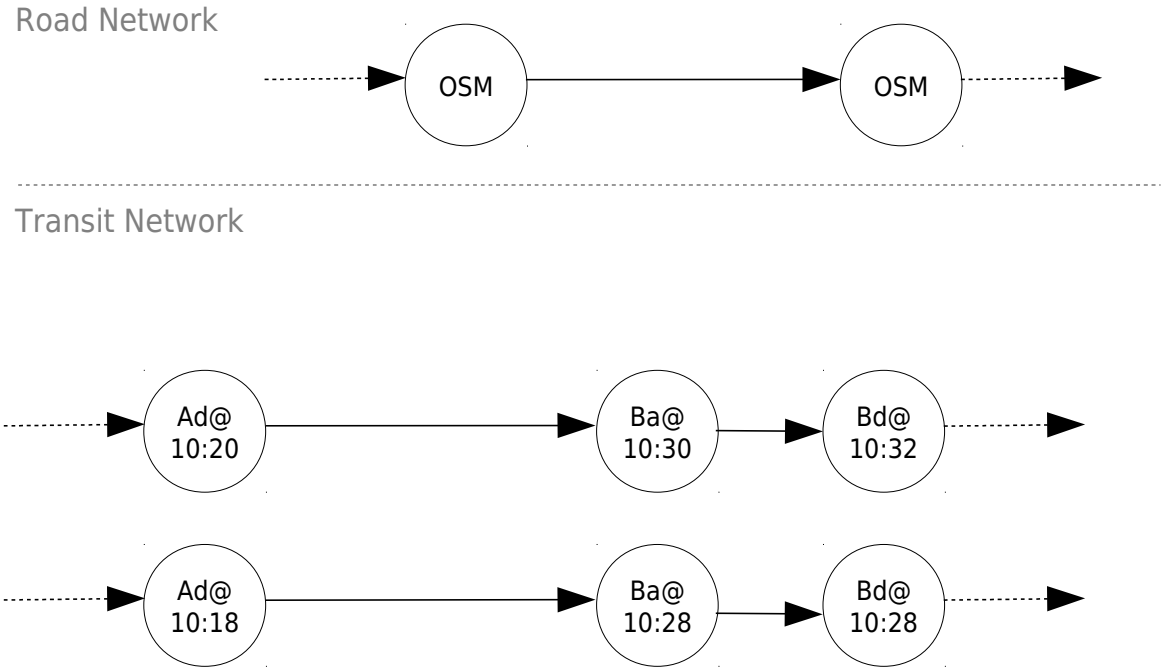
Constructing the Road Network

UNI FREIBURG

- Each junction is a node in the graph
- Edges are the road segments
- Cost of edge = walking time
- Data: OSM (OpenStreetMaps)
  - ways (street as vector of nodes)
  - nodes with latitude and longitude

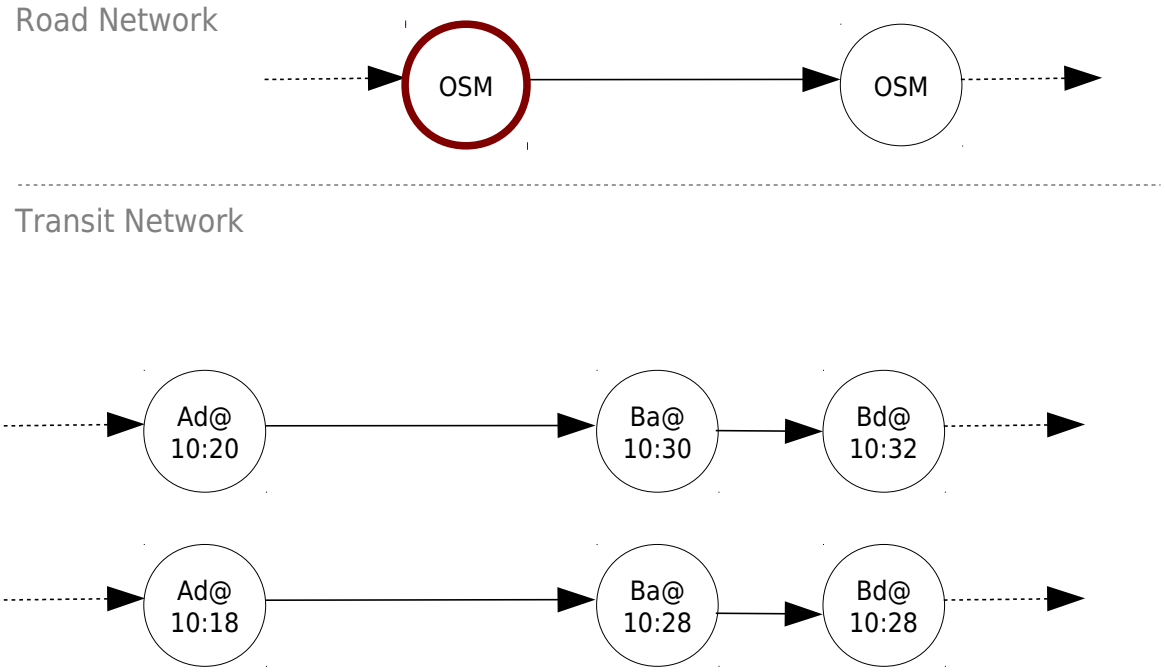
Combining Road & Transit Network

- Time-Expanded Network



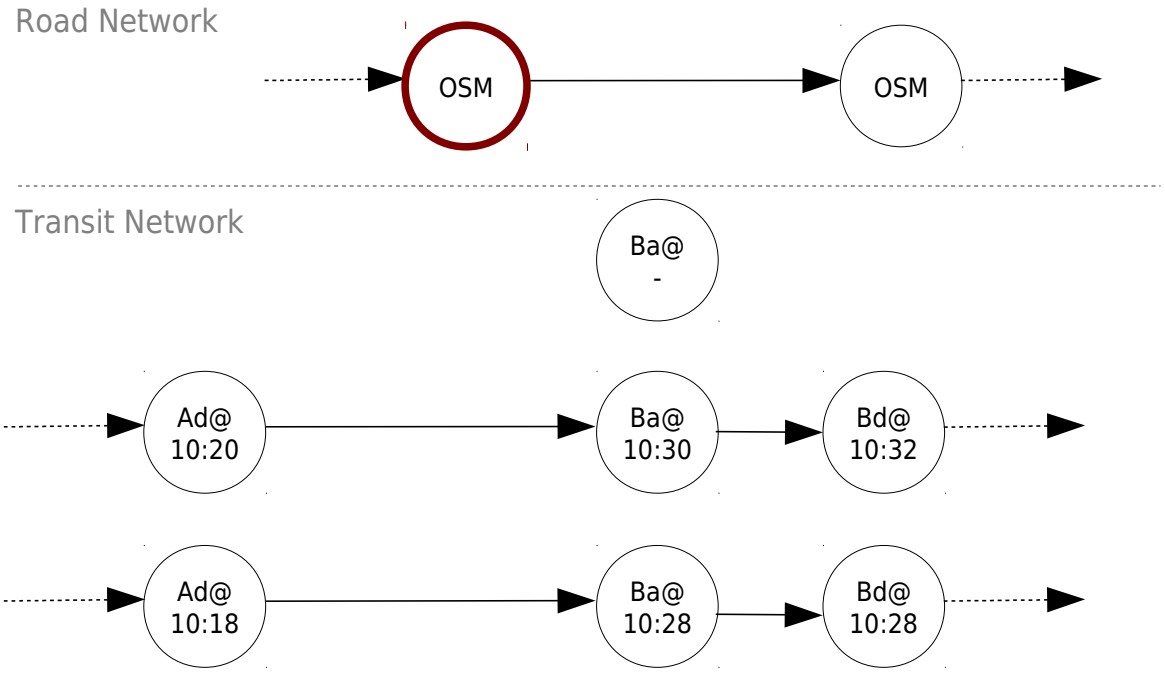
Combining Road & Transit Network

- Time-Expanded Network



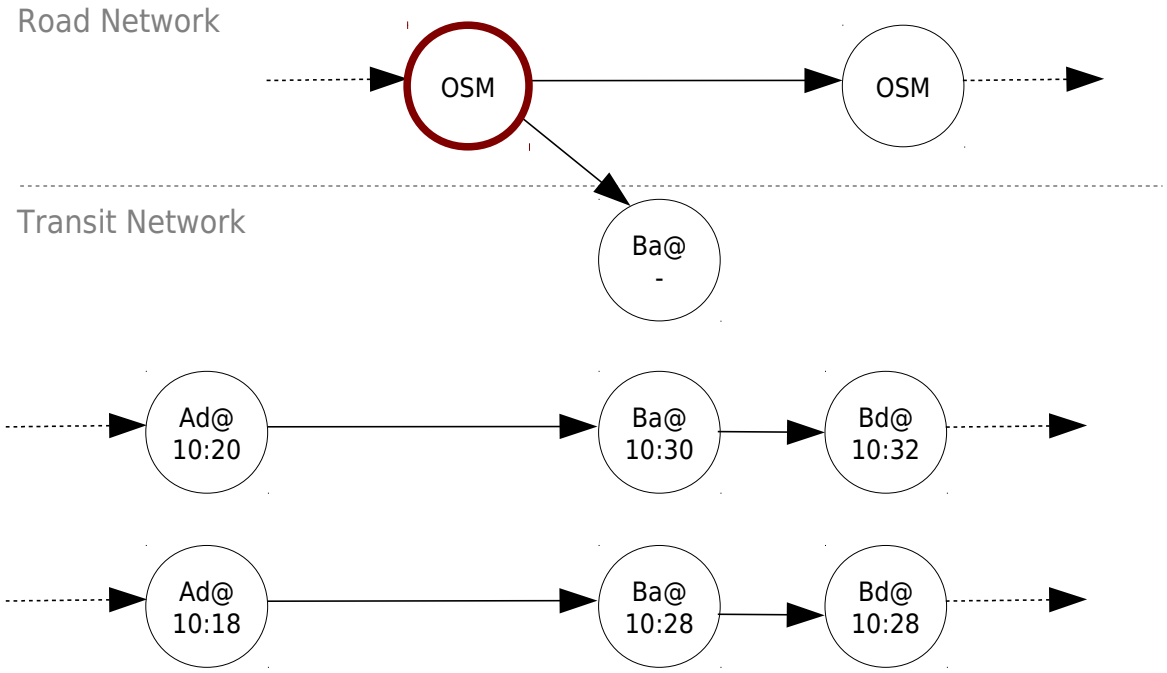
Combining Road & Transit Network

- Time-Expanded Network



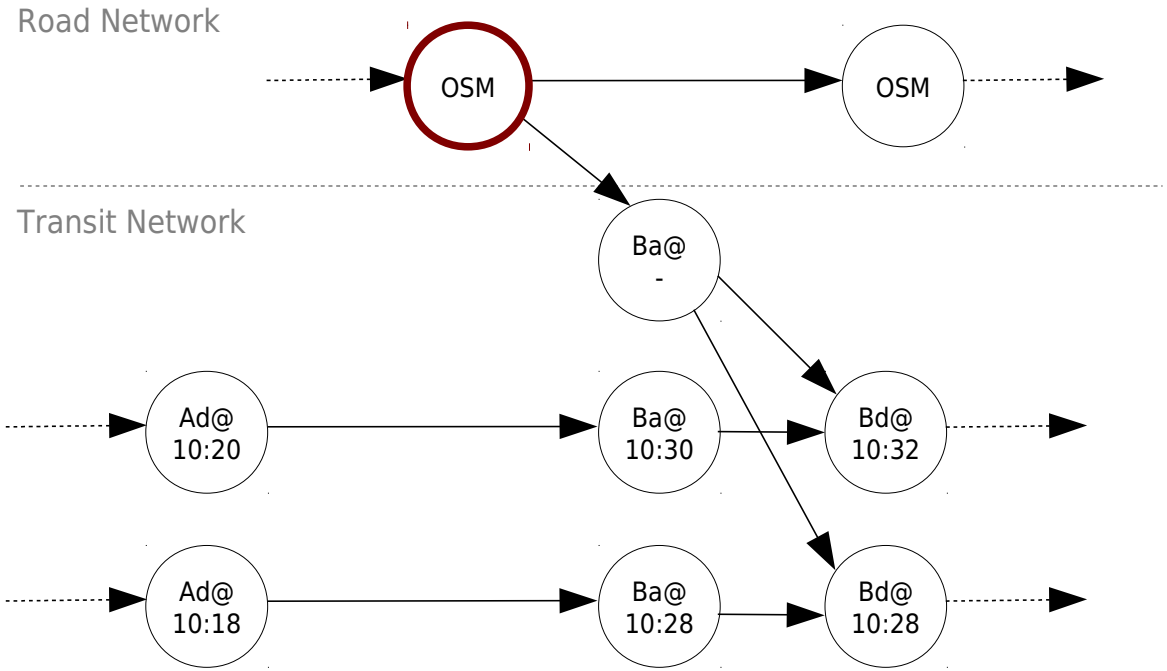
Combining Road & Transit Network

- Time-Expanded Network



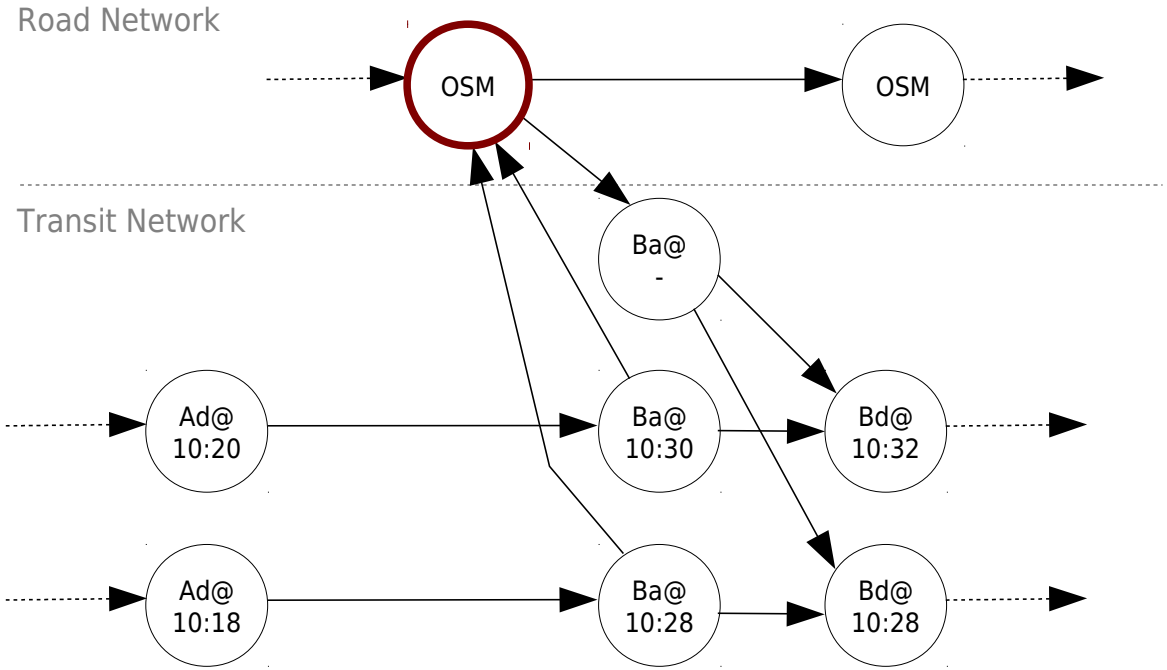
Combining Road & Transit Network

- Time-Expanded Network



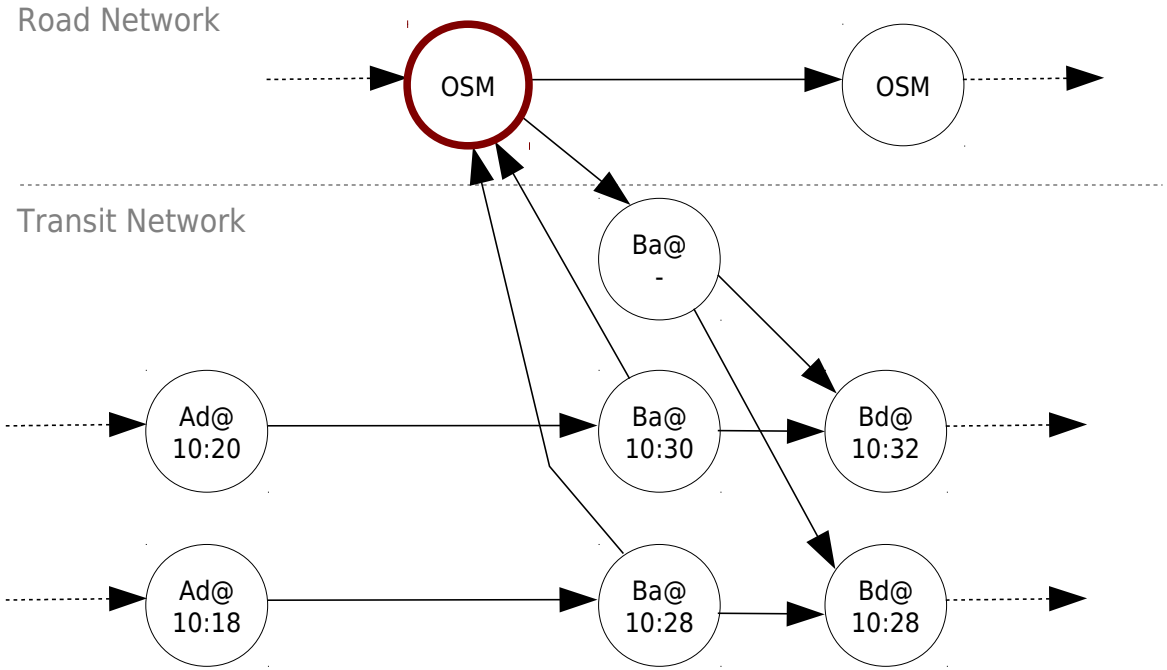
Combining Road & Transit Network

- Time-Expanded Network



Combining Road & Transit Network

- Time-Expanded Network



- Time-Dependent Network
  - adding bidirectional edge from the geographically closest OSM node to the corresponding station

Bi-criteria Optimization

UNI FREIBURG

- Two criteria to optimize

Bi-criteria Optimization

UNI FREIBURG

- Two criteria to optimize
  - labels at nodes of the form (travel time, penalty)

Bi-criteria Optimization

UNI FREIBURG

- Two criteria to optimize
  - labels at nodes of the form (travel time, penalty)
  - penalty = # transfers + [walking penalty]

Bi-criteria Optimization

UNI FREIBURG

- Two criteria to optimize
  - labels at nodes of the form (travel time, penalty)
  - penalty = # transfers + [walking penalty]
  - more than one optimal solution
    - (45,1) incomparable to (30,2)
    - but (30,1) better than (45,1)

Bi-criteria Optimization

UNI FREIBURG

- Two criteria to optimize
  - labels at nodes of the form (travel time, penalty)
  - penalty = # transfers + [walking penalty]
  - more than one optimal solution
    - (45,1) incomparable to (30,2)
    - but (30,1) better than (45,1)
  - increase penalty value when changing vehicles

Bi-criteria Optimization

UNI FREIBURG

- Two criteria to optimize
  - labels at nodes of the form (travel time, penalty)
  - penalty = # transfers + [walking penalty]
  - more than one optimal solution
    - (45,1) incomparable to (30,2)
    - but (30,1) better than (45,1)
  - increase penalty value when changing vehicles
  - increase penalty value when walking for a minute

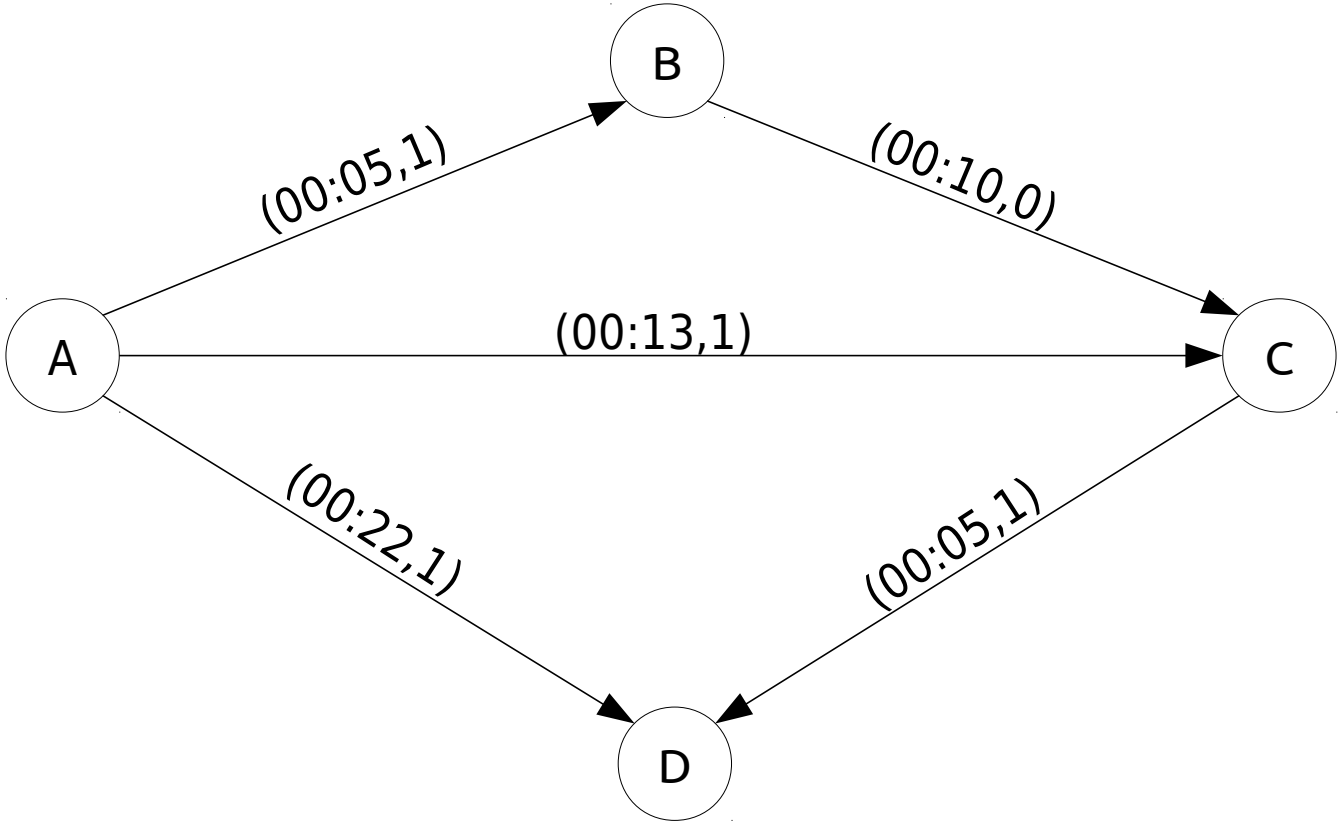
Bi-criteria Optimization

UNI FREIBURG

- Two criteria to optimize
  - labels at nodes of the form (travel time, penalty)
  - penalty = # transfers + [walking penalty]
  - more than one optimal solution
    - (45,1) incomparable to (30,2)
    - but (30,1) better than (45,1)
  - increase penalty value when changing vehicles
  - increase penalty value when walking for a minute
  - compute all optimal solutions
    - use a Multi Label Dijkstra

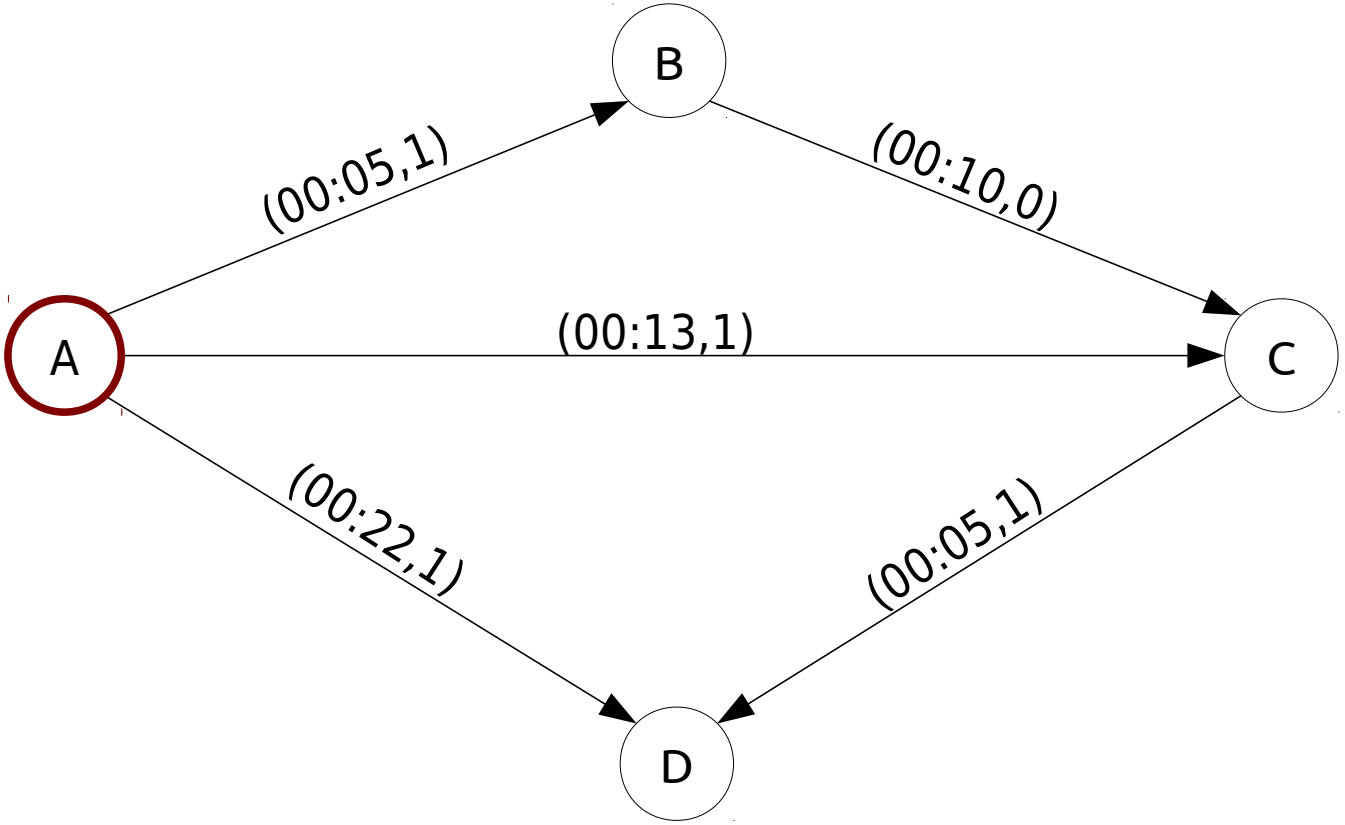
Multi Label Dijkstra

- Example execution



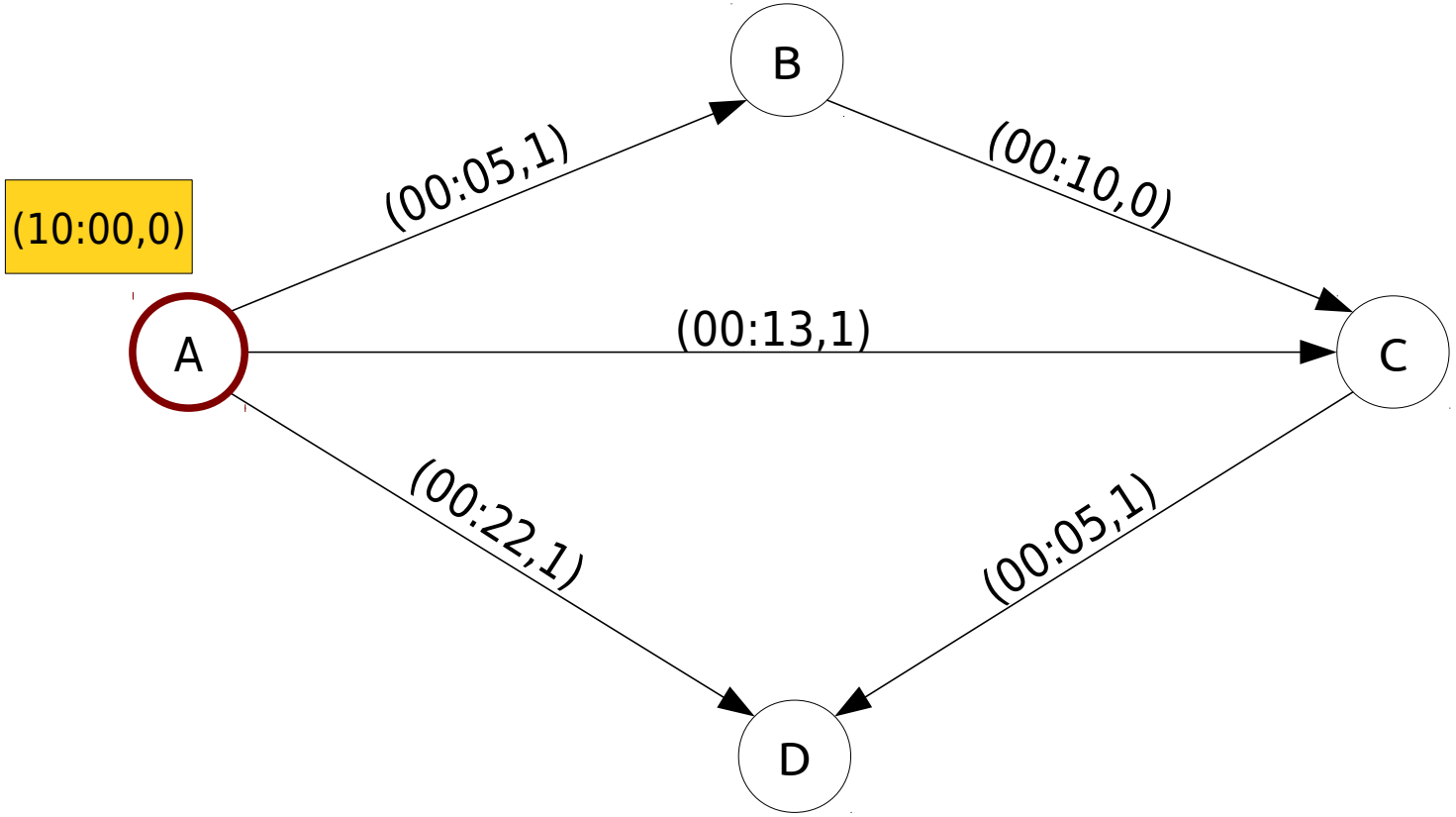
Multi Label Dijkstra

- Example execution



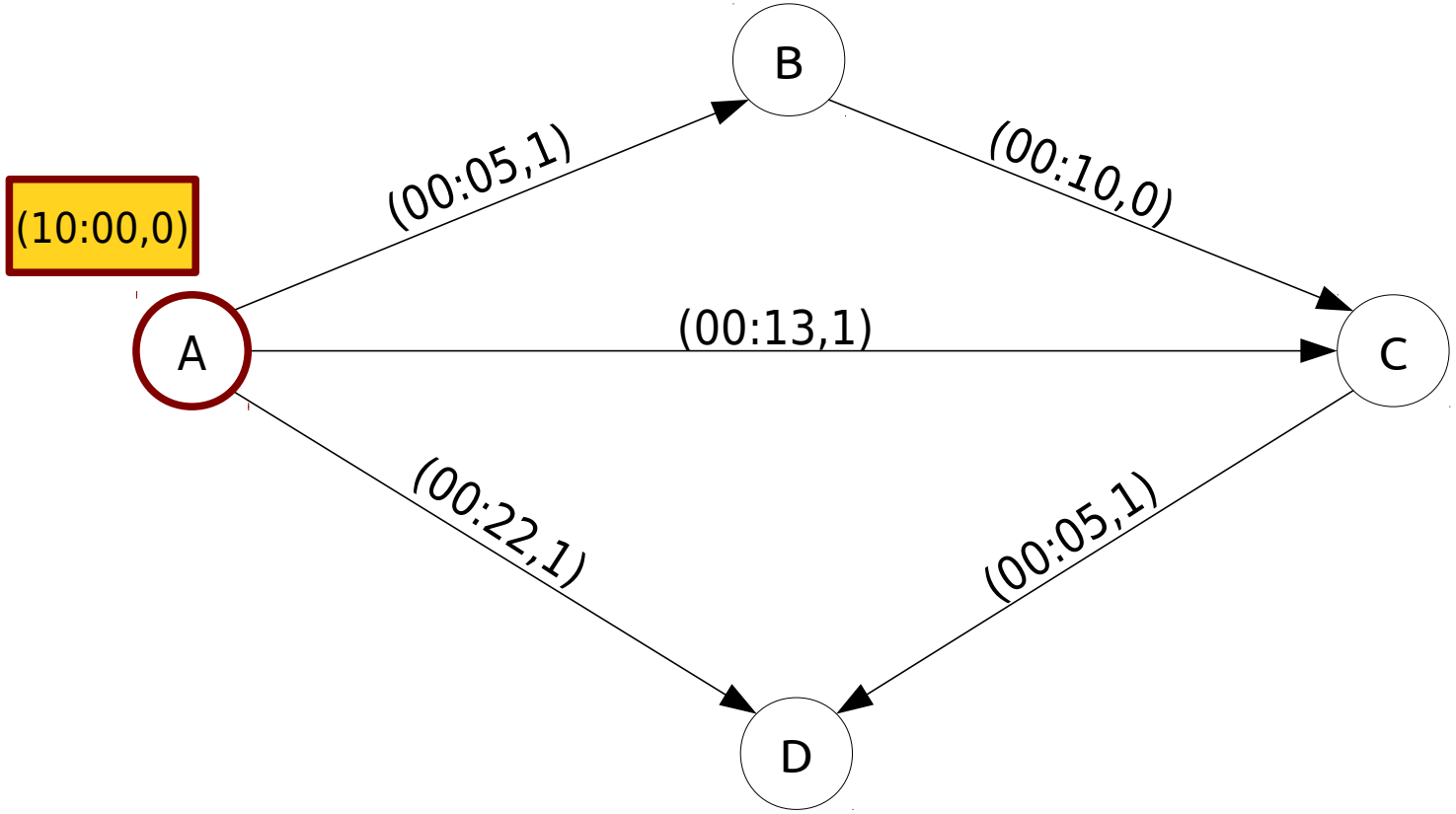
Multi Label Dijkstra

- Example execution



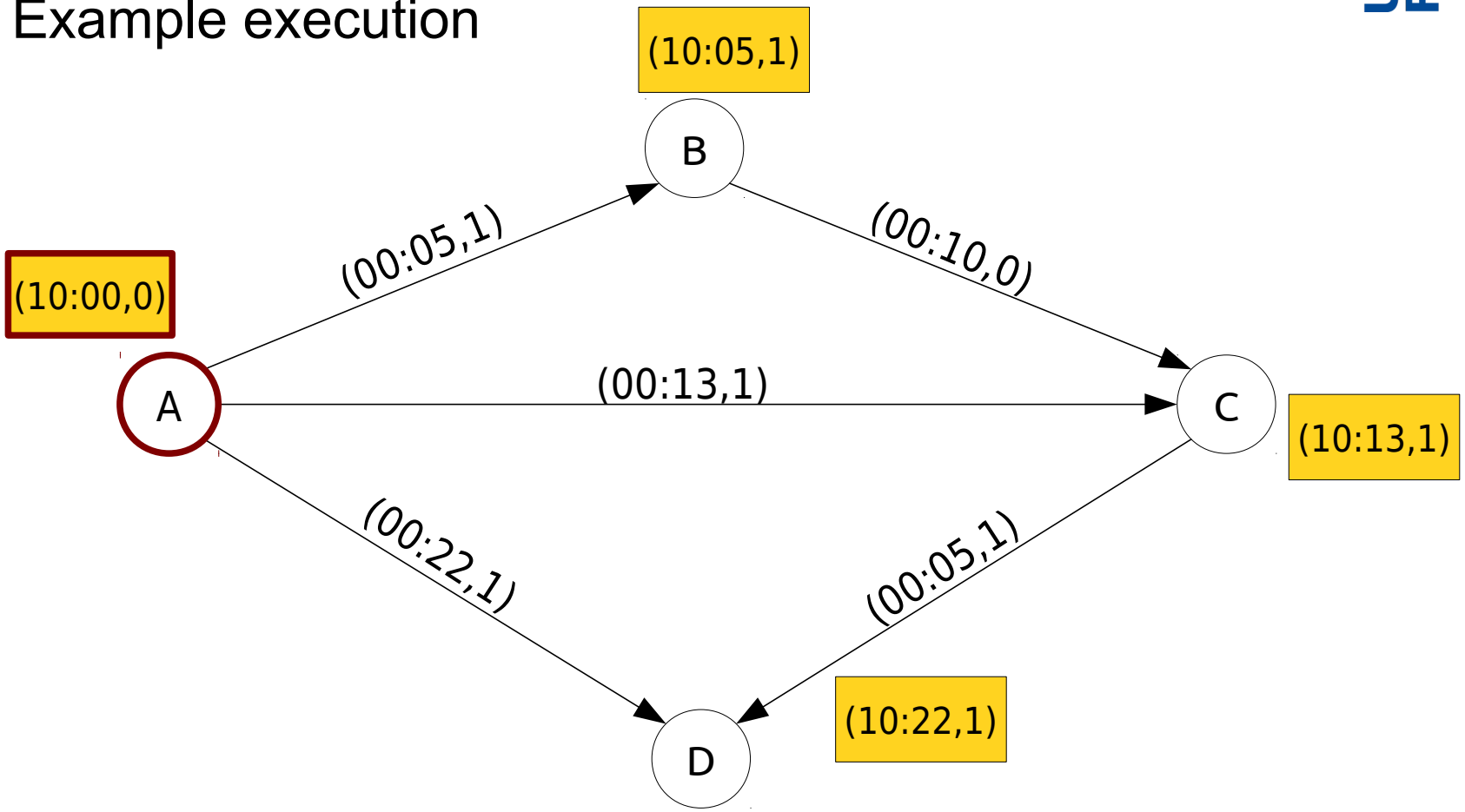
Multi Label Dijkstra

- Example execution



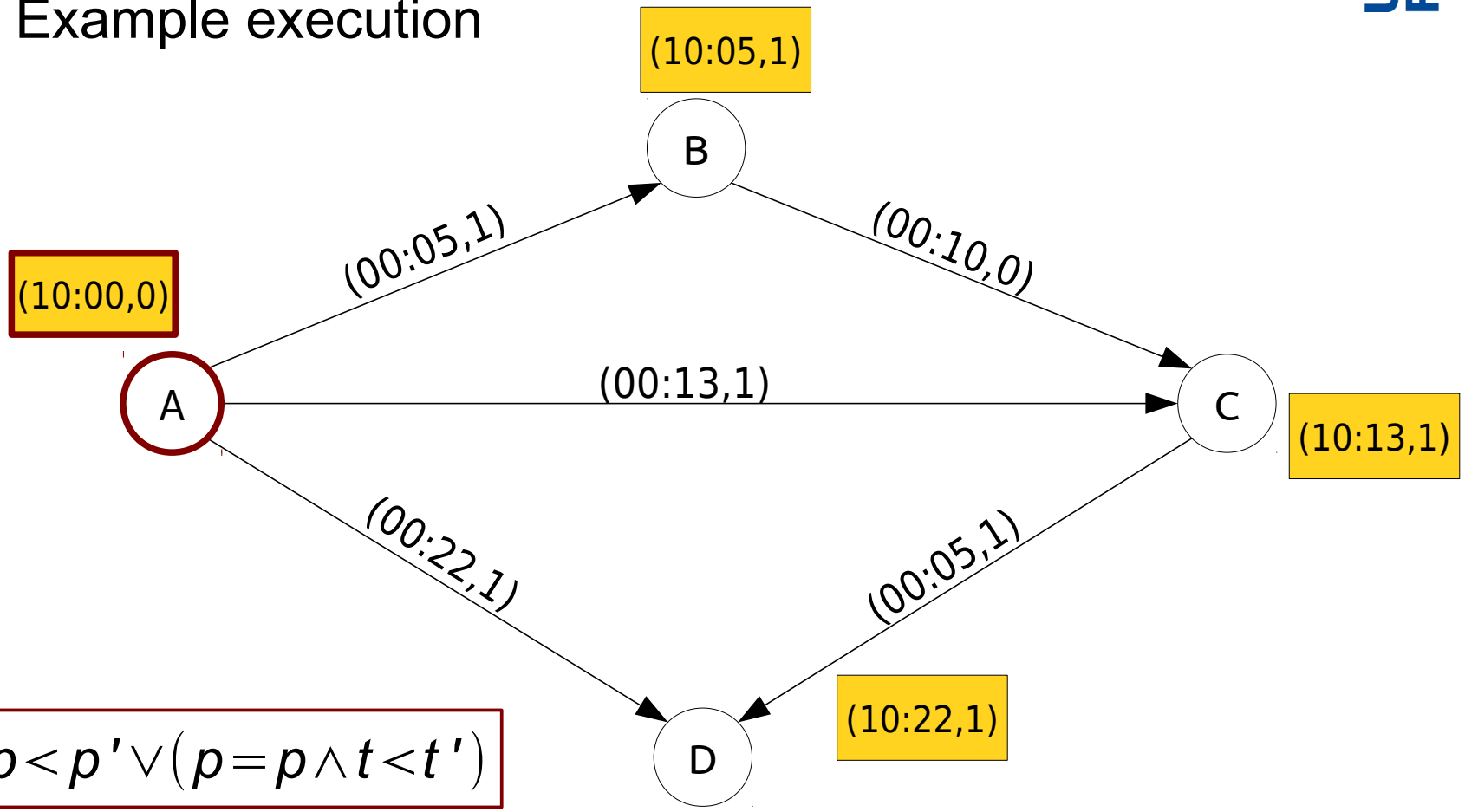
Multi Label Dijkstra

- Example execution



Multi Label Dijkstra

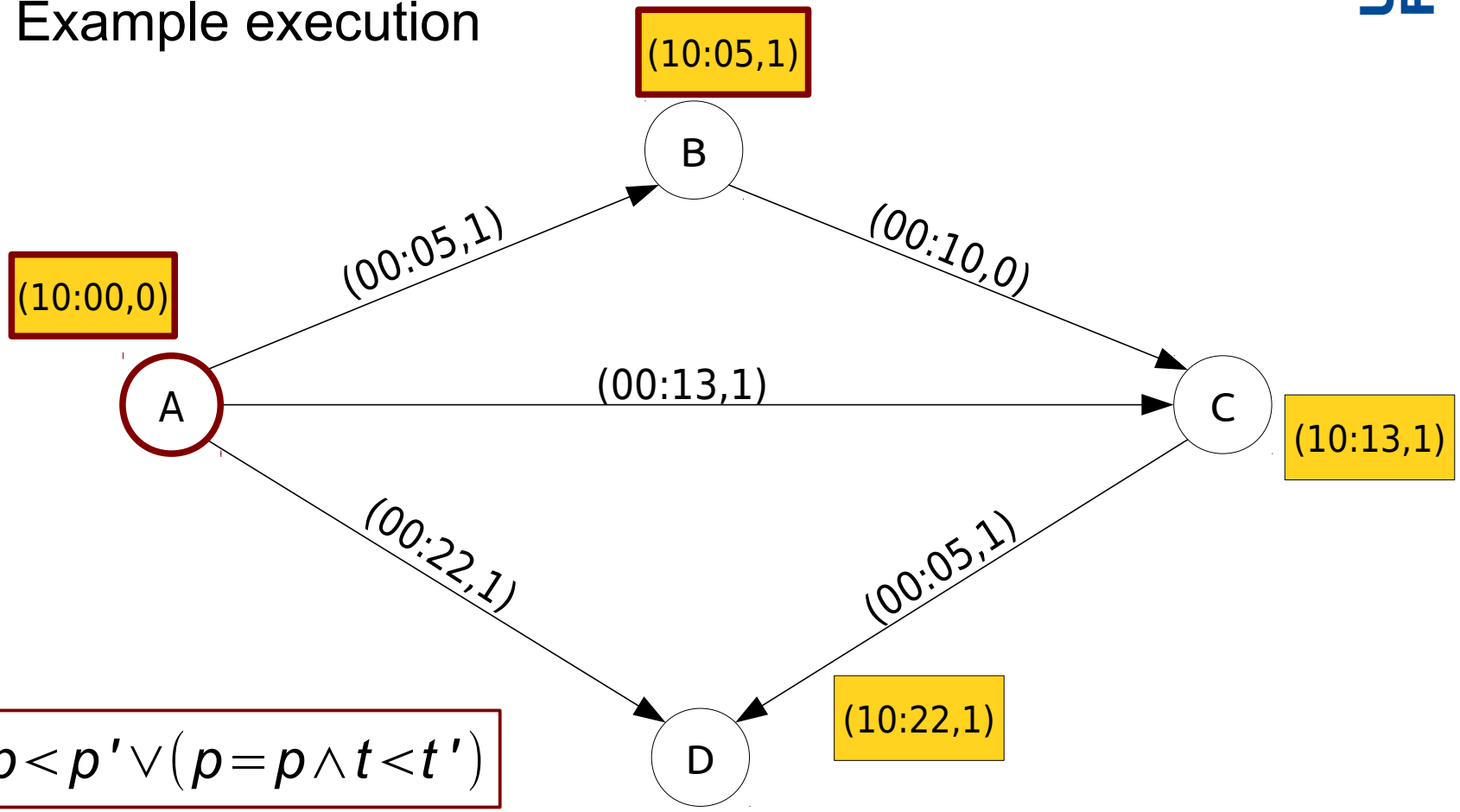
- Example execution



$$p < p' \vee \left(p = p \wedge t < t' \right)$$

Multi Label Dijkstra

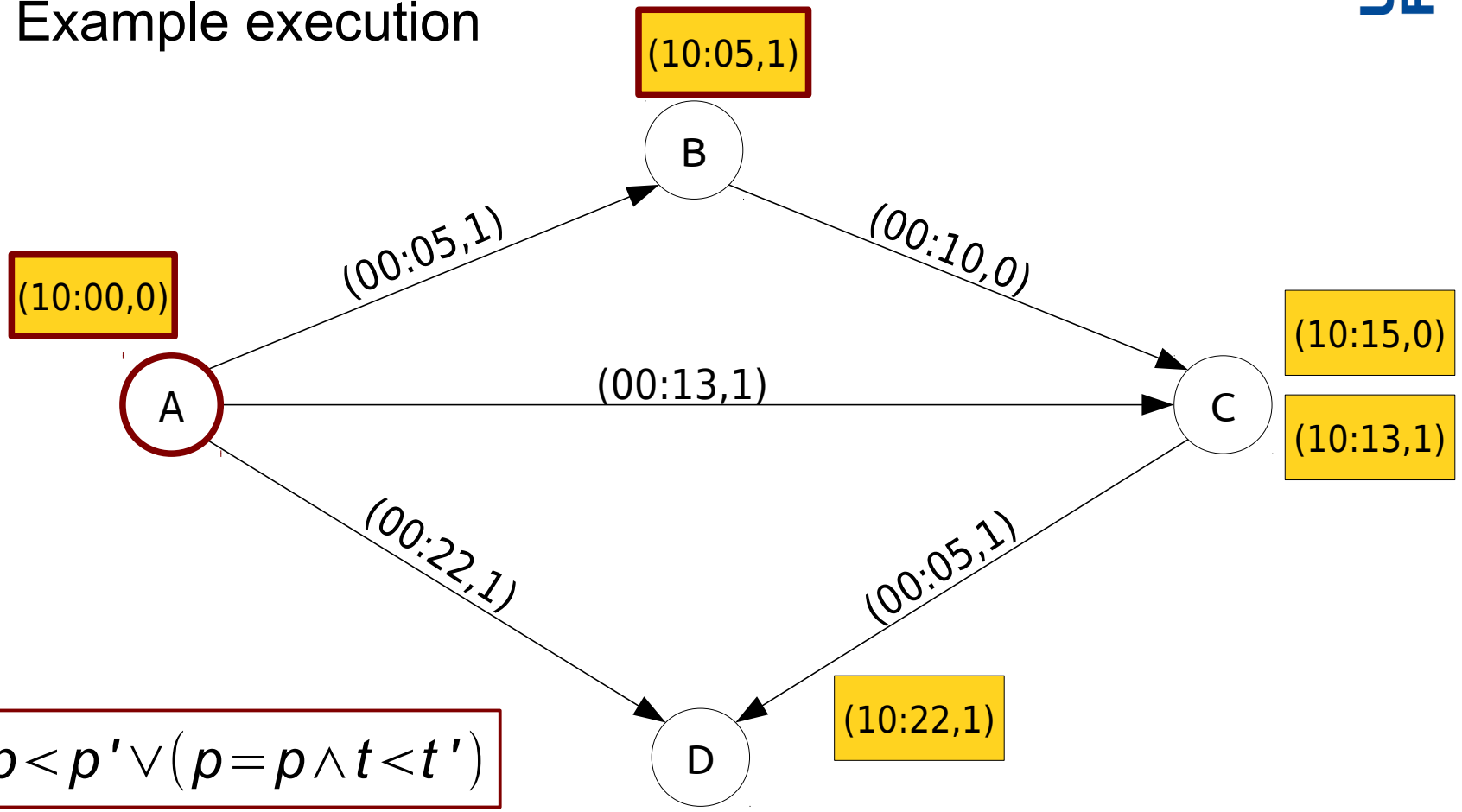
- Example execution



$$p < p' \vee \left(p = p \wedge t < t' \right)$$

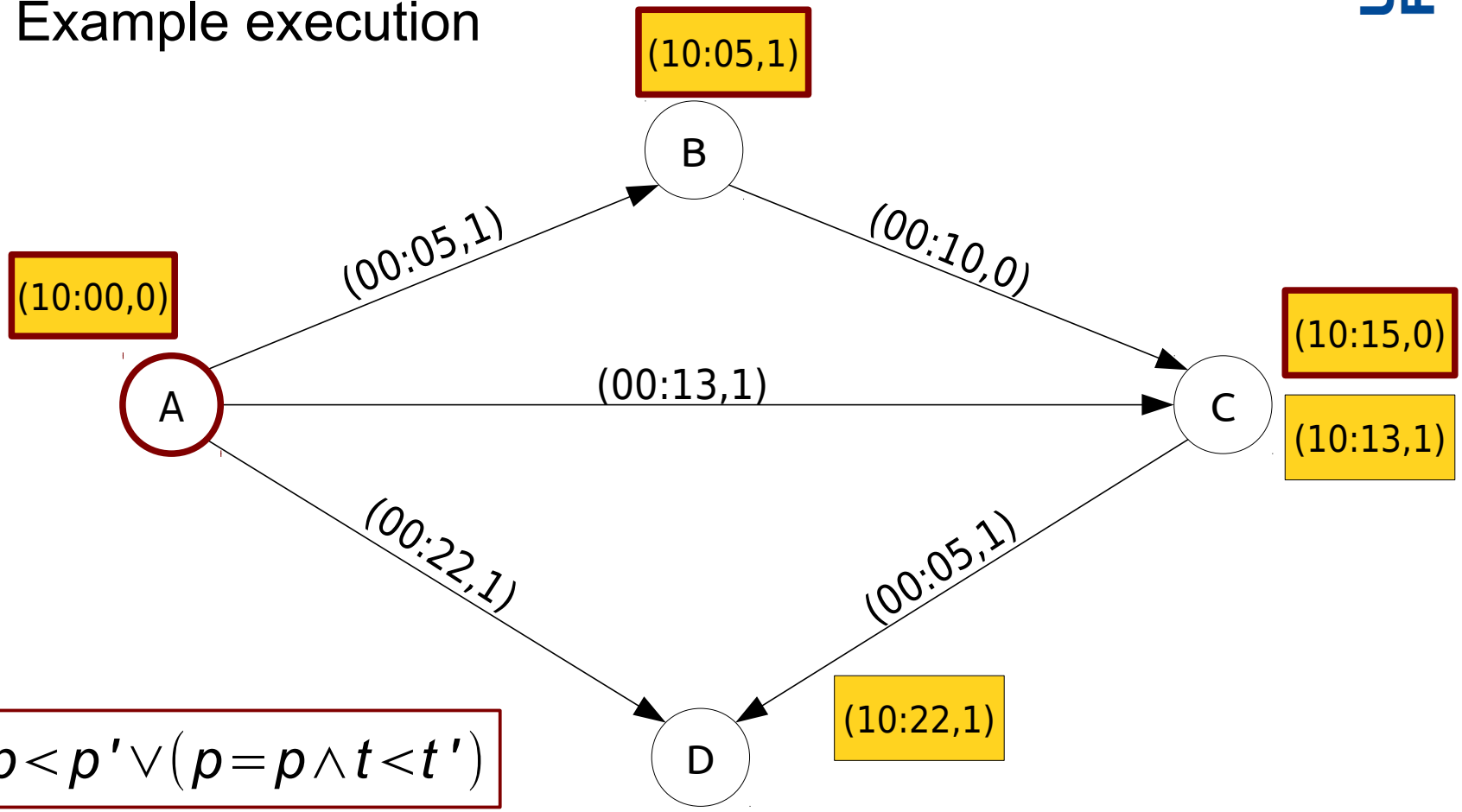
Multi Label Dijkstra

- Example execution



Multi Label Dijkstra

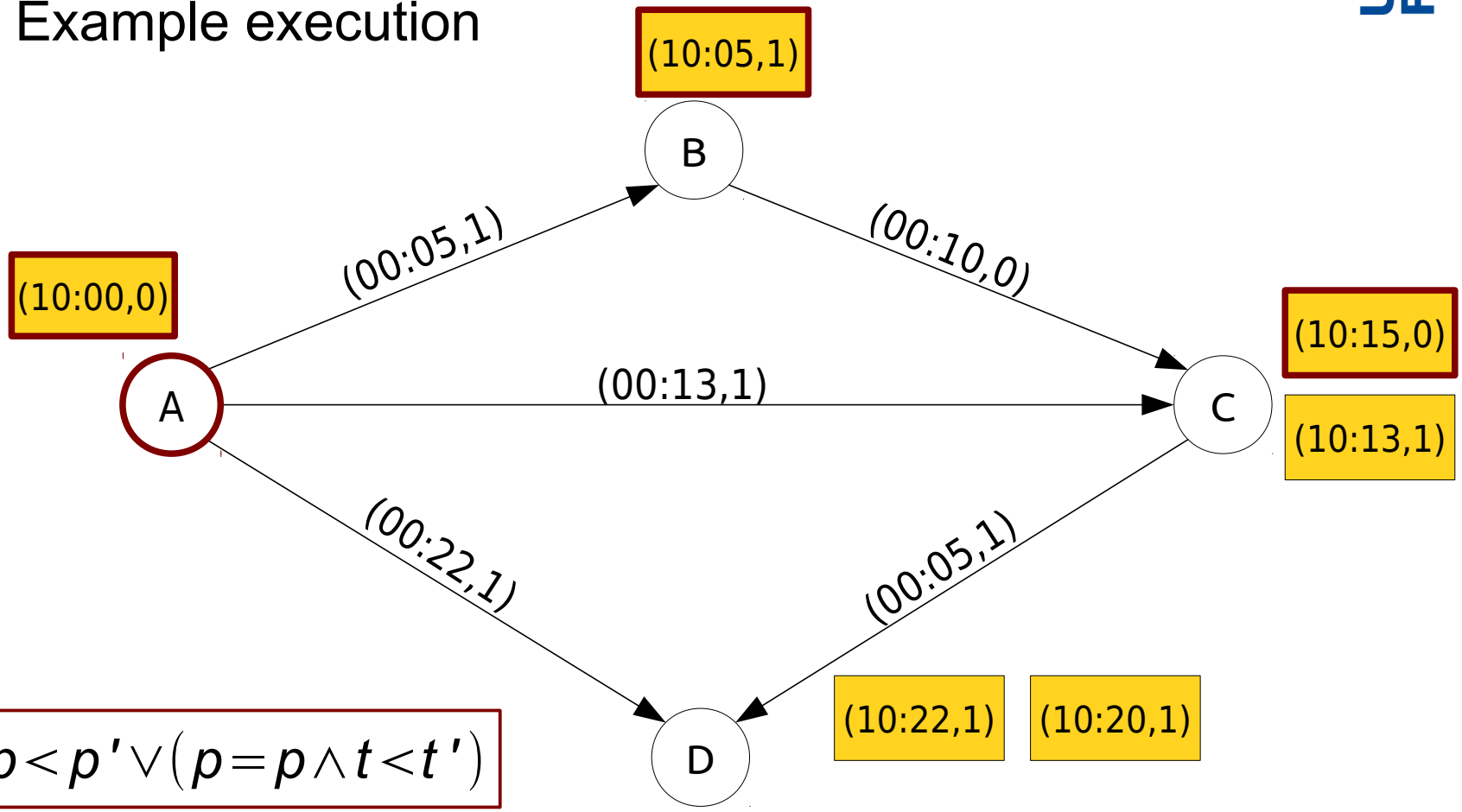
- Example execution



$$p < p' \vee \left(p = p \wedge t < t' \right)$$

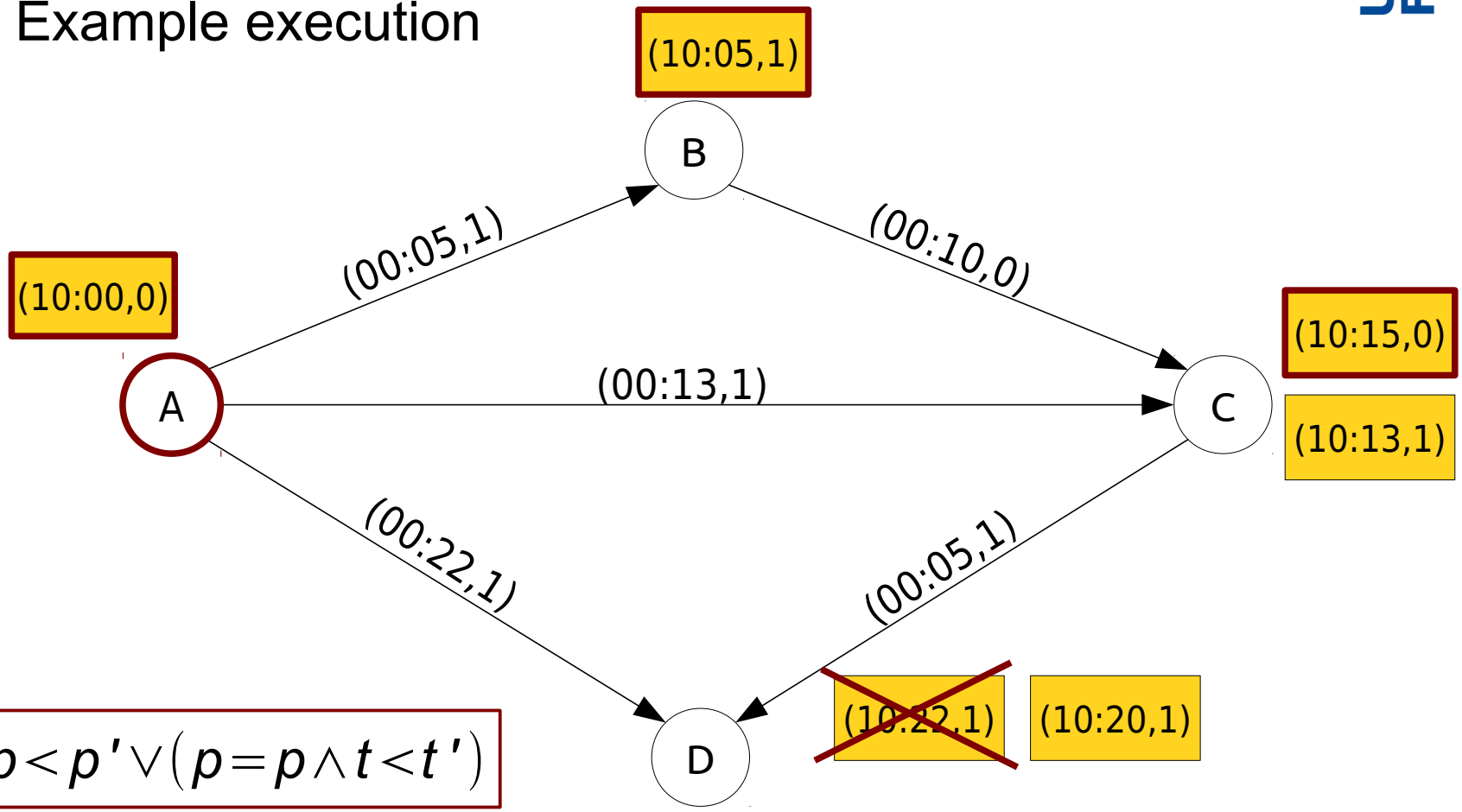
Multi Label Dijkstra

- Example execution



Multi Label Dijkstra

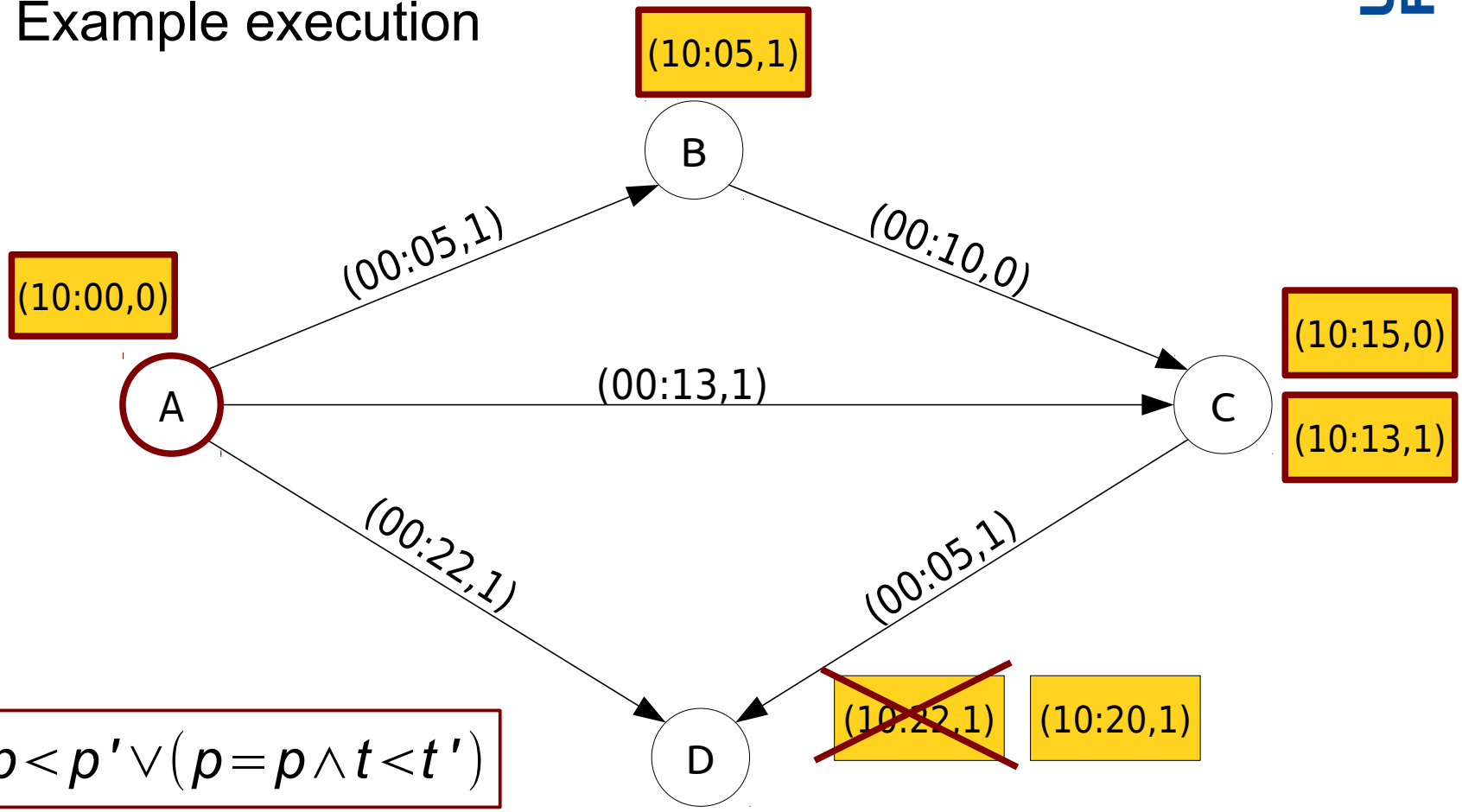
- Example execution



$$p < p' \vee \left(p = p \wedge t < t' \right)$$

Multi Label Dijkstra

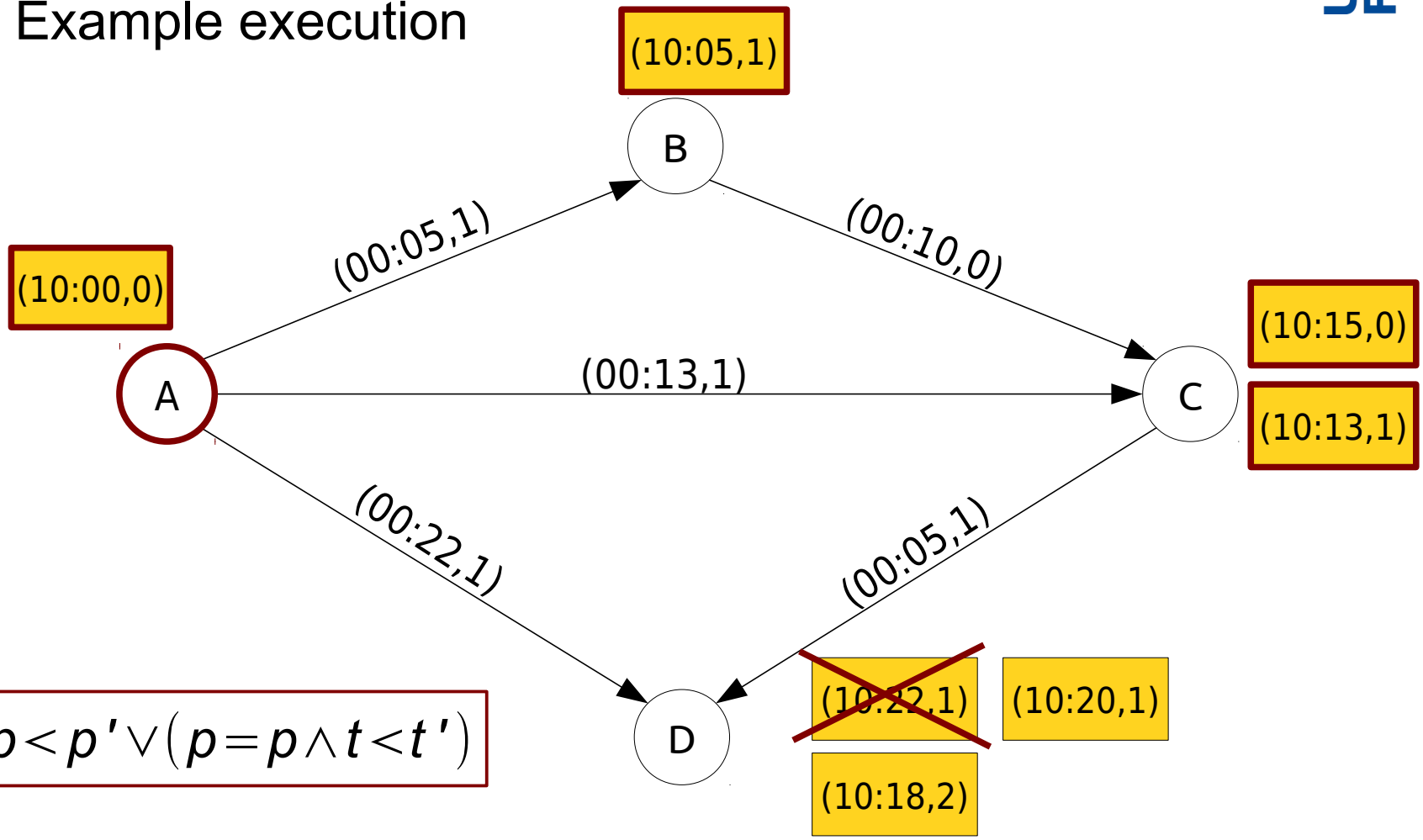
- Example execution



$$p < p' \vee \left(p = p \wedge t < t' \right)$$

Multi Label Dijkstra

- Example execution



Maintaining Label Sets

- When relaxing edge (v,w) create label

  $(t,p)_w \leftarrow (t,p)_v + (t,p)_e$

Maintaining Label Sets

UNI FREIBURG

- When relaxing edge (v,w) create label

  $(t,p)_w \leftarrow (t,p)_v + (t,p)_e$

- For all labels $(t',p')_w$ at w
  - if $(t,p)_w < (t',p')_w$ remove label $(t',p')_w$

Maintaining Label Sets

UNI FREIBURG

- When relaxing edge (v,w) create label
  $(t,p)_w \leftarrow (t,p)_v + (t,p)_e$
- For all labels $(t',p')_w$ at w
  - if $(t,p)_w < (t',p')_w$ remove label $(t',p')_w$
- If $(t,p)_w$ is not dominated or incomparable to all $(t',p')_w$
  - insert $(t,p)_w$ into label set at w
  - insert $(t,p)_w$ into priority queue

Experiments 1/2

- Comparing the Graph Size

	# nodes	# edges	space
Exp	48.3 M	112.4 M	10 679 MB
Dep	4.5 M	9.4 M	1 643 MB

  - GTFS Data: New York City Subway, New York City Bus (Bronx, Brooklyn, Manhattan, Queens, Staten Island), Long Island Bus and the Metro-North Railroad
  - OSM Data: State New York
  - Date interval: 09.04.2012 – 15.04.2012

Experiments 2/2

- Comparing Performance

	time	settled labels	labels/station
bi-criteria (# transfers only)			
Exp	62.3 s	20.7 M	-
Dep	16.7 s	2.0 M	3
bi-criteria (# transfers and minimizing walking)			
Exp	89.2 s	33.8 M	-
Dep	58.6 s	2.4 M	13

- 1000 random paths

Conclusion

UNI FREIBURG

- Shortest path problems on combined road & transit networks using the TE and TD approach

Conclusion

UNI FREIBURG

- Shortest path problems on combined road & transit networks using the TE and TD approach
- Minimize travel time & # transfers + [walking between stations]

Conclusion

UNI FREIBURG

- Shortest path problems on combined road & transit networks using the TE and TD approach
- Minimize travel time & # transfers + [walking between stations]
- TD approach about 3.7 times faster in average than the TE approach (bi-criteria)

Conclusion

UNI FREIBURG

- Shortest path problems on combined road & transit networks using the TE and TD approach
- Minimize travel time & # transfers + [walking between stations]
- TD approach about 3.7 times faster in average than the TE approach (bi-criteria)
- When minimizing walking between stations in addition, the gap is smaller (speed up 1.5)

UNI FREIBURG

Traffic Days

- Interval e. g. 09.04.2012, 00:00 – 15.04.2012, 24:00
- Have connection between stations/nodes for a day if there is a valid connection between stations
- Times stored are integers relative to the begin of the interval
  - e. g. 11.04.2012,10:20:
    - $t = 2 \cdot 24 \cdot 3600 + 10 \cdot 3600 + 20 \cdot 60$
- Actual time of day = $t \bmod (24 \cdot 3600)$
- Actual day = $\lfloor t/(24 \cdot 3600) \rfloor$

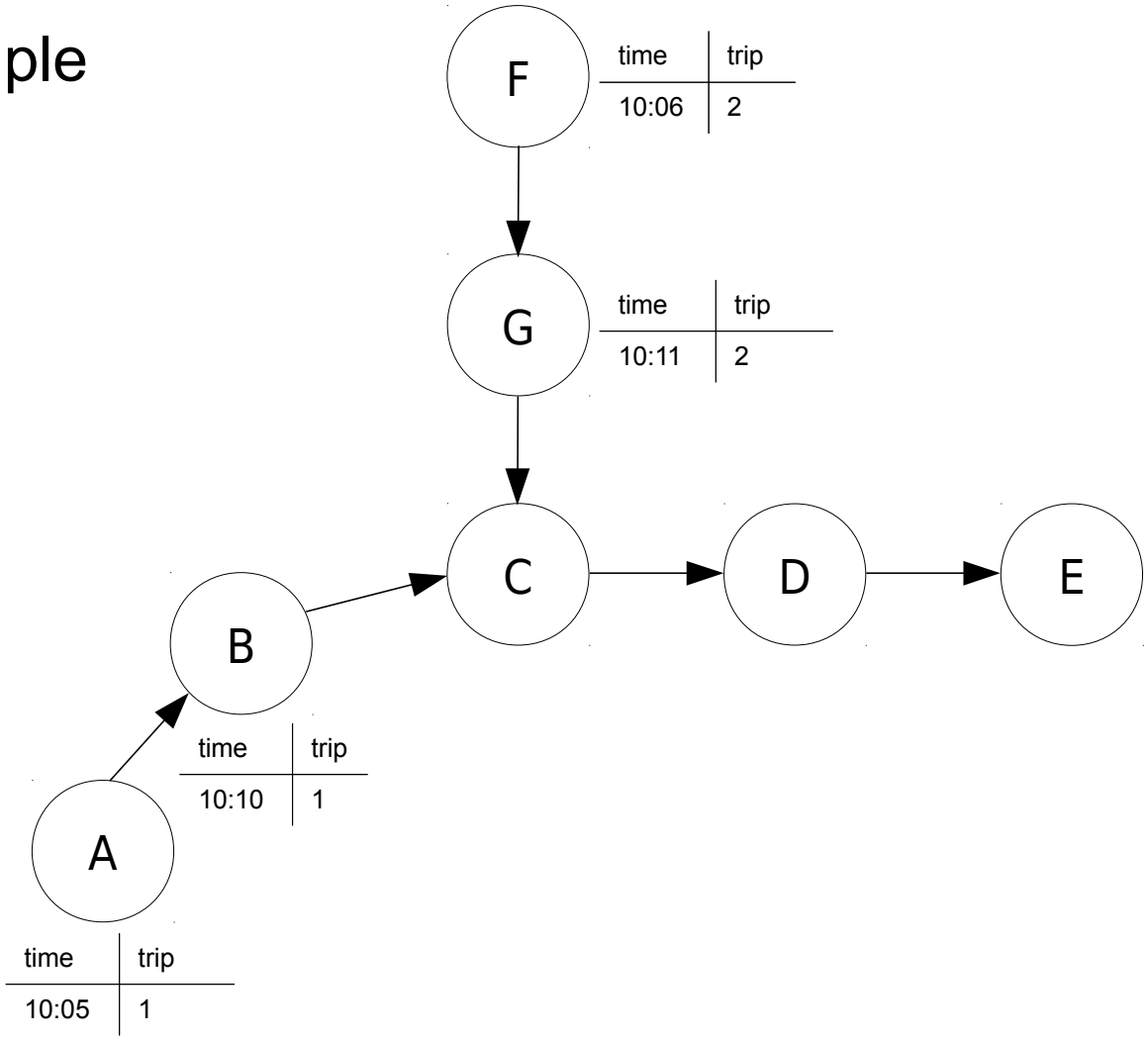
Reducing Labels per Station (TD)

UNI FREIBURG

- Problem: too many labels per station
- Solution: for each trip pair remember how many stations both trips have in common up to the final station
- Do precomputation:
  - hash map $h : (\ell,\ell') \rightarrow (\text{arr}(S, \ell), \text{arr}(S, \ell'))$
  - If for two labels (t,p), (t',p') there is an entry and $t \geq \text{arr}(S, \ell) \wedge t' \geq \text{arr}(S, \ell')$
    - compare labels in the time-expanded sense

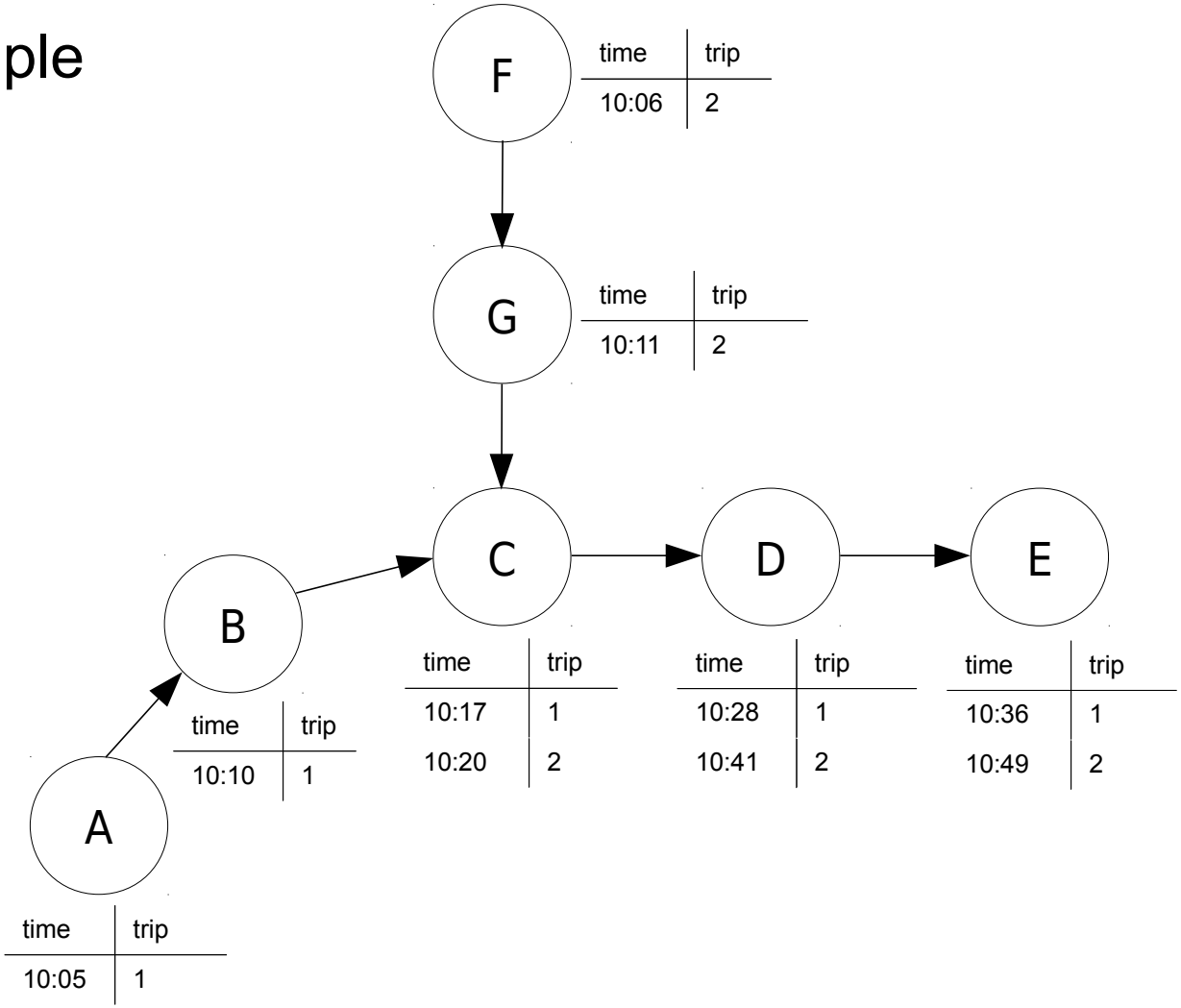
Reducing Labels per Station (TD)

- Example



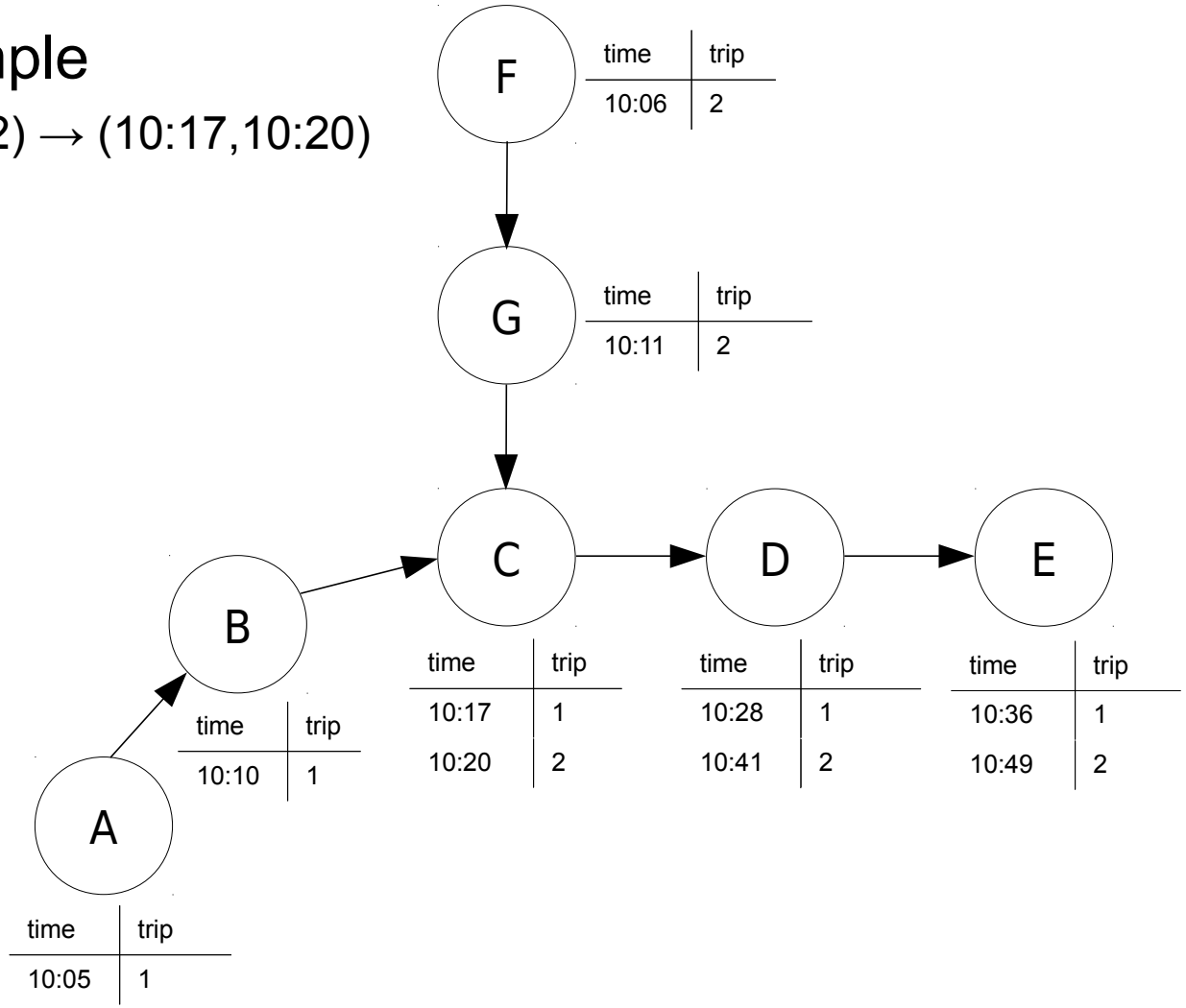
Reducing Labels per Station (TD)

- Example



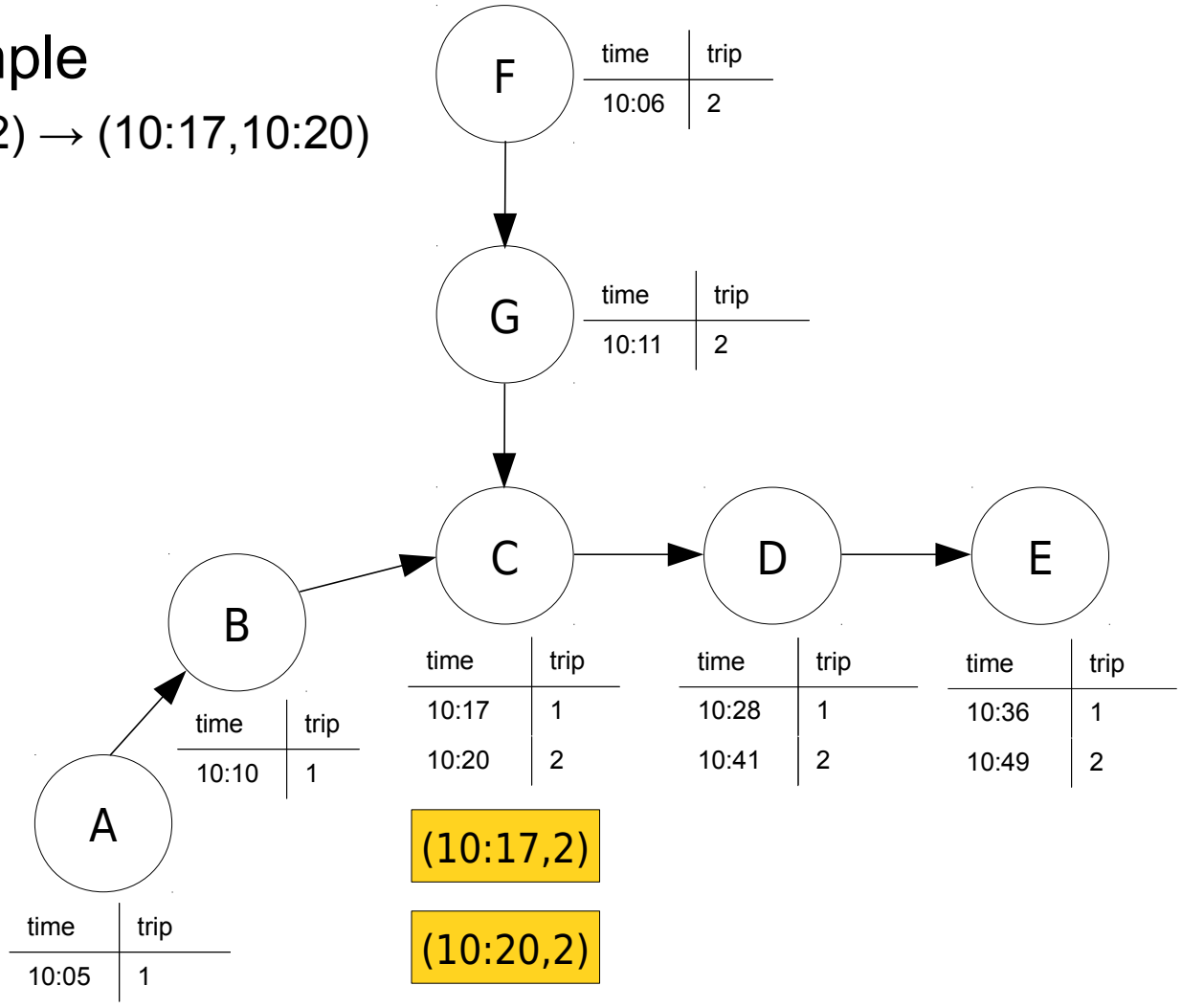
Reducing Labels per Station (TD)

- Example
  - (1,2) → (10:17,10:20)



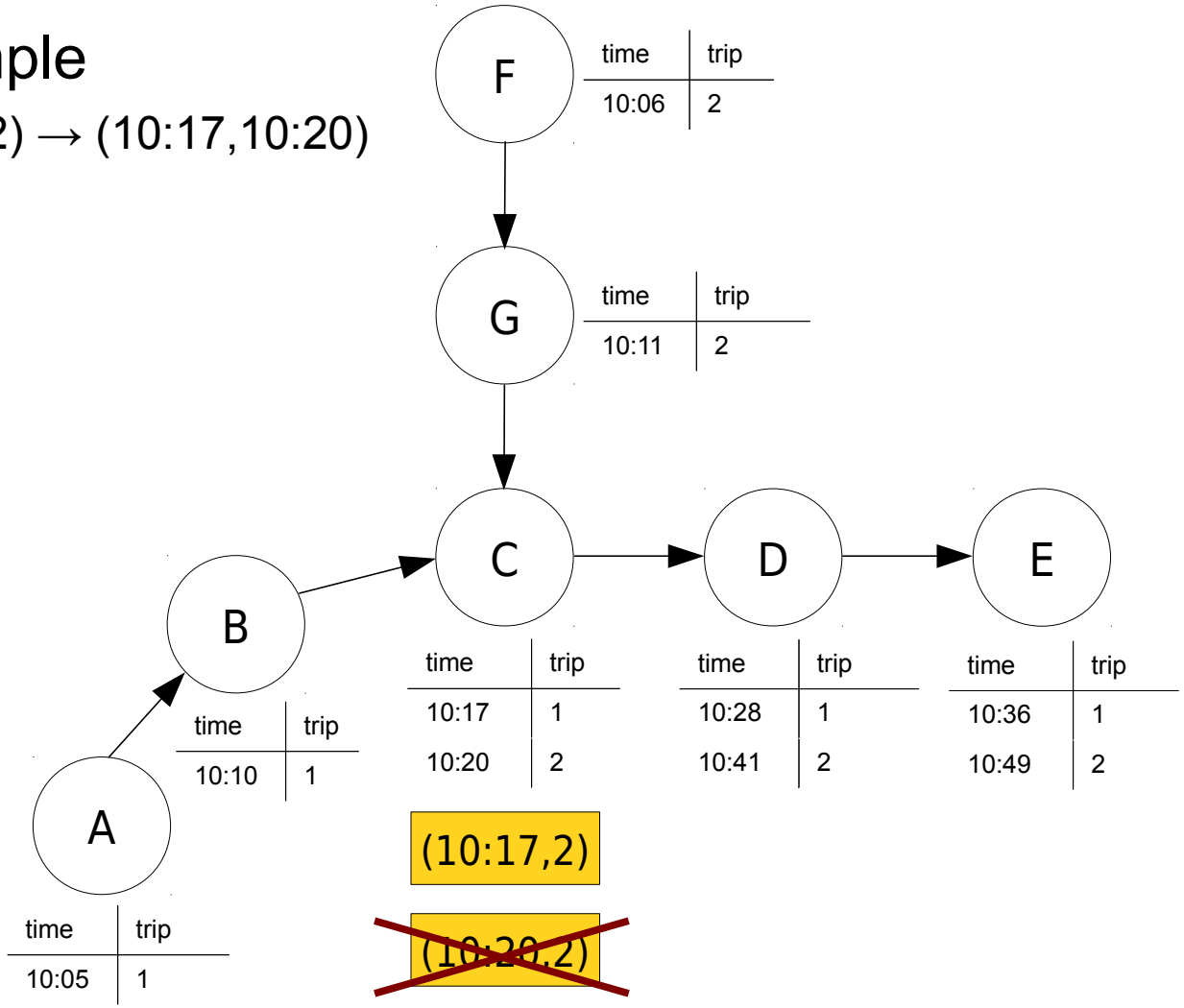
Reducing Labels per Station (TD)

- Example
  - (1,2) → (10:17,10:20)



Reducing Labels per Station (TD)

- Example
  - (1,2) → (10:17,10:20)



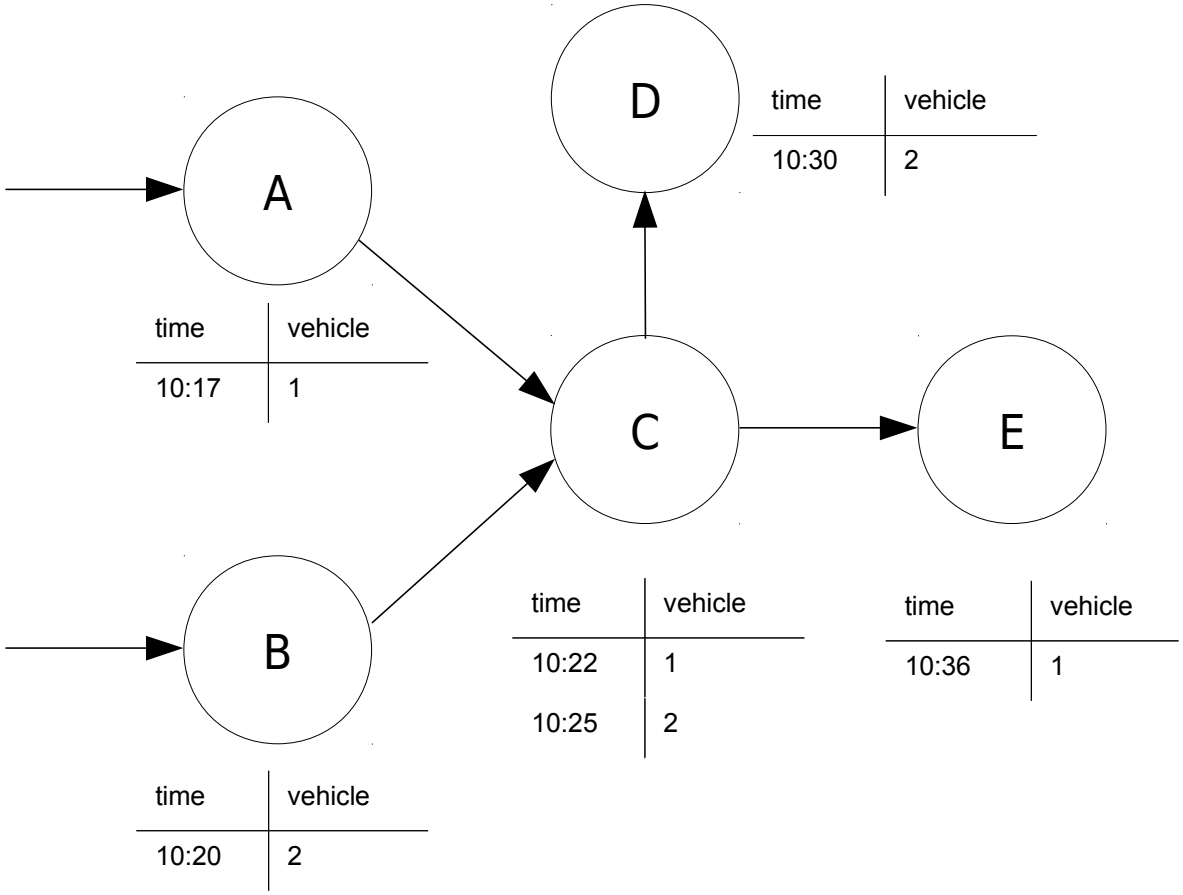
Compare Labels

UNI FREIBURG

- Time-Expanded Approach
  - $(t, p) \leq (t', p')$ if and only if $t \leq t'$ and $p \leq p'$
  - $(t, p) < (t', p')$ if and only if $(t < t' \wedge p \leq p') \vee (t \leq t' \wedge p < p')$
  - (t, p) (t', p') incomparable if neither $(t, p) \leq (t', p')$ nor $(t', p') \leq (t, p)$
- Time-Dependent Approach
  - for labels (t, p) and (t', p') with $\ell = \ell'$ use label domination in the time expanded sense
  - $(t, p) < (t', p')$ if and only if $t + TB \leq t' \wedge p < p'$
  - (t, p) (t', p') incomparable if neither $(t, p) < (t', p')$ nor $(t', p') < (t, p)$

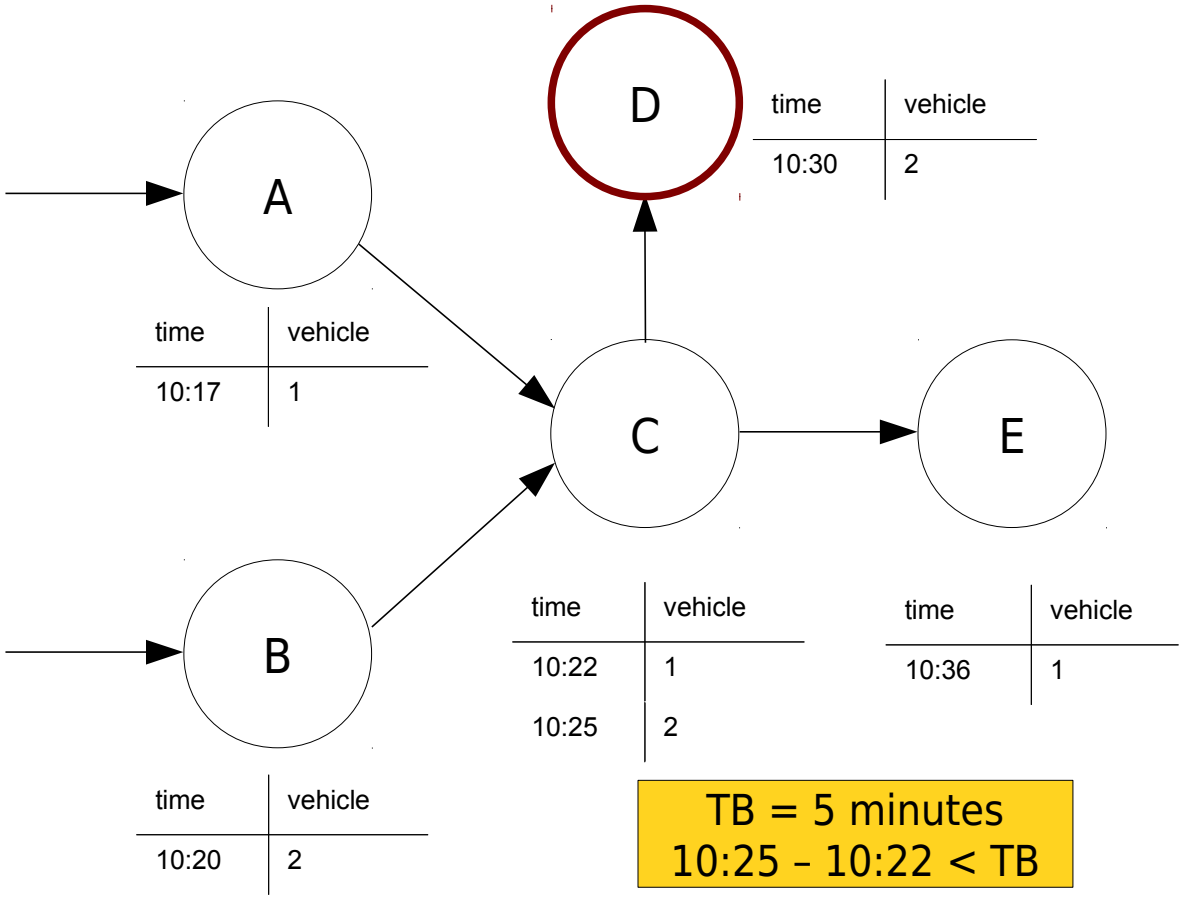
Time Dependent Problem 1

- Keeps labels with times t and t` with |t-t`|<TB



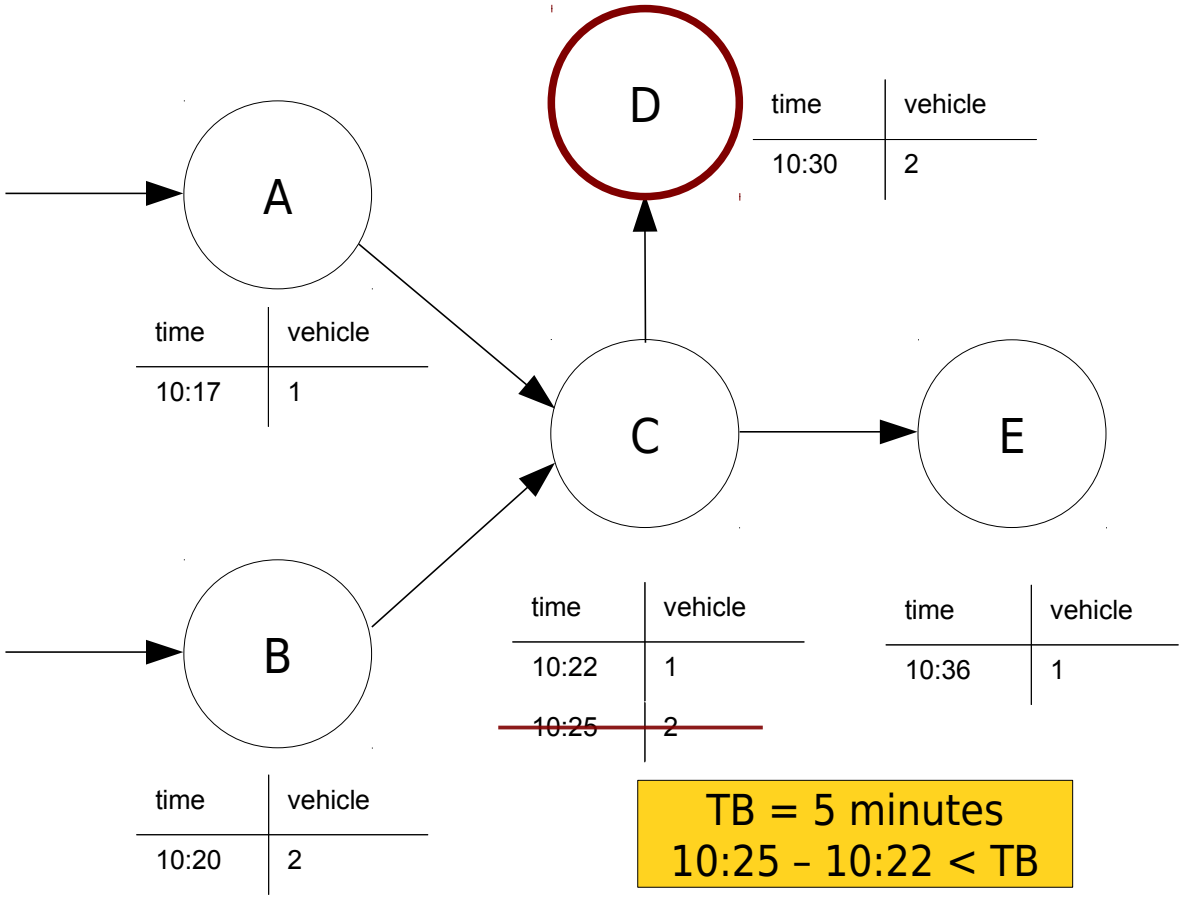
Time Dependent Problem 1

- Keeps labels with times t and t` with |t-t`|<TB



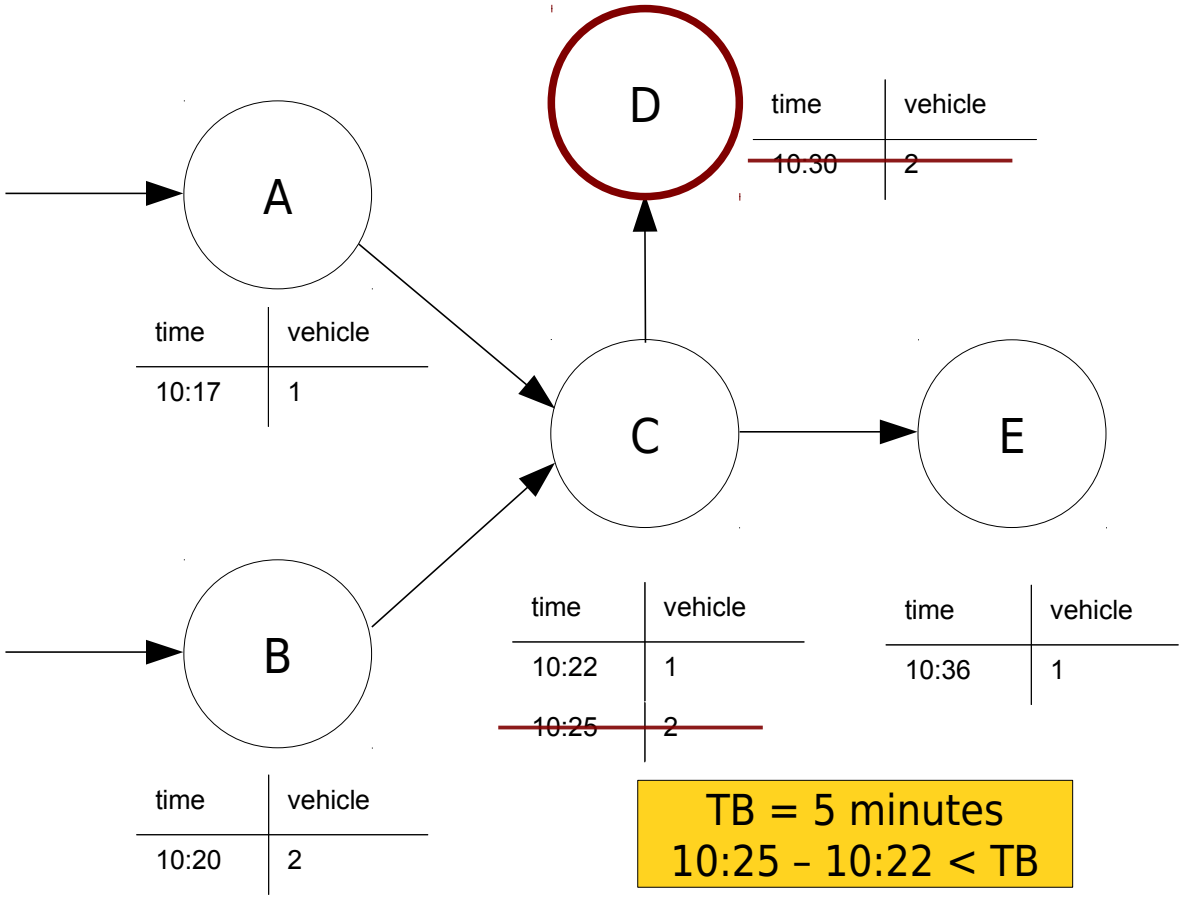
Time Dependent Problem 1

- Keeps labels with times t and t` with |t-t`|<TB



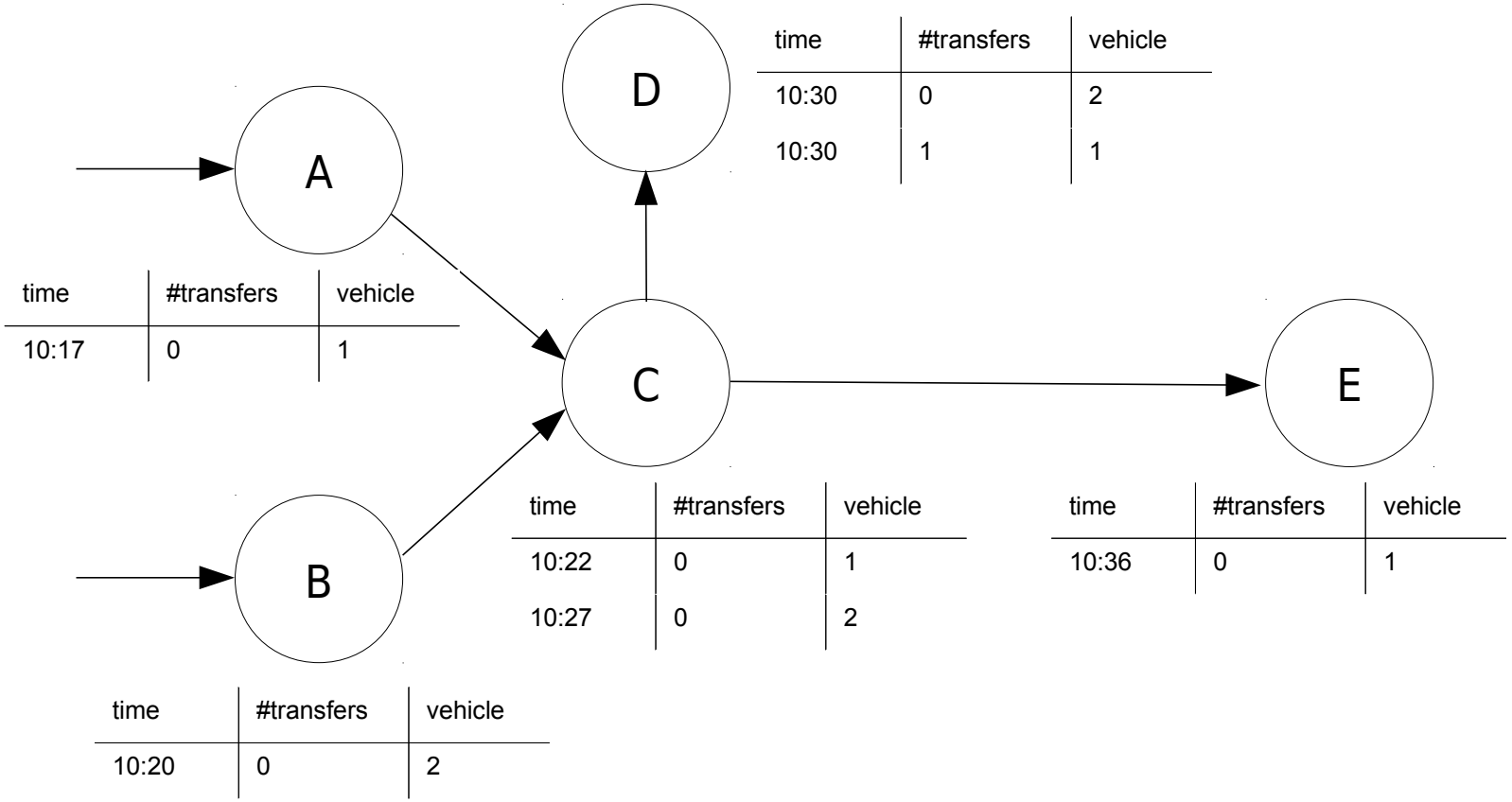
Time Dependent Problem 1

- Keeps labels with times t and t` with |t-t`|<TB



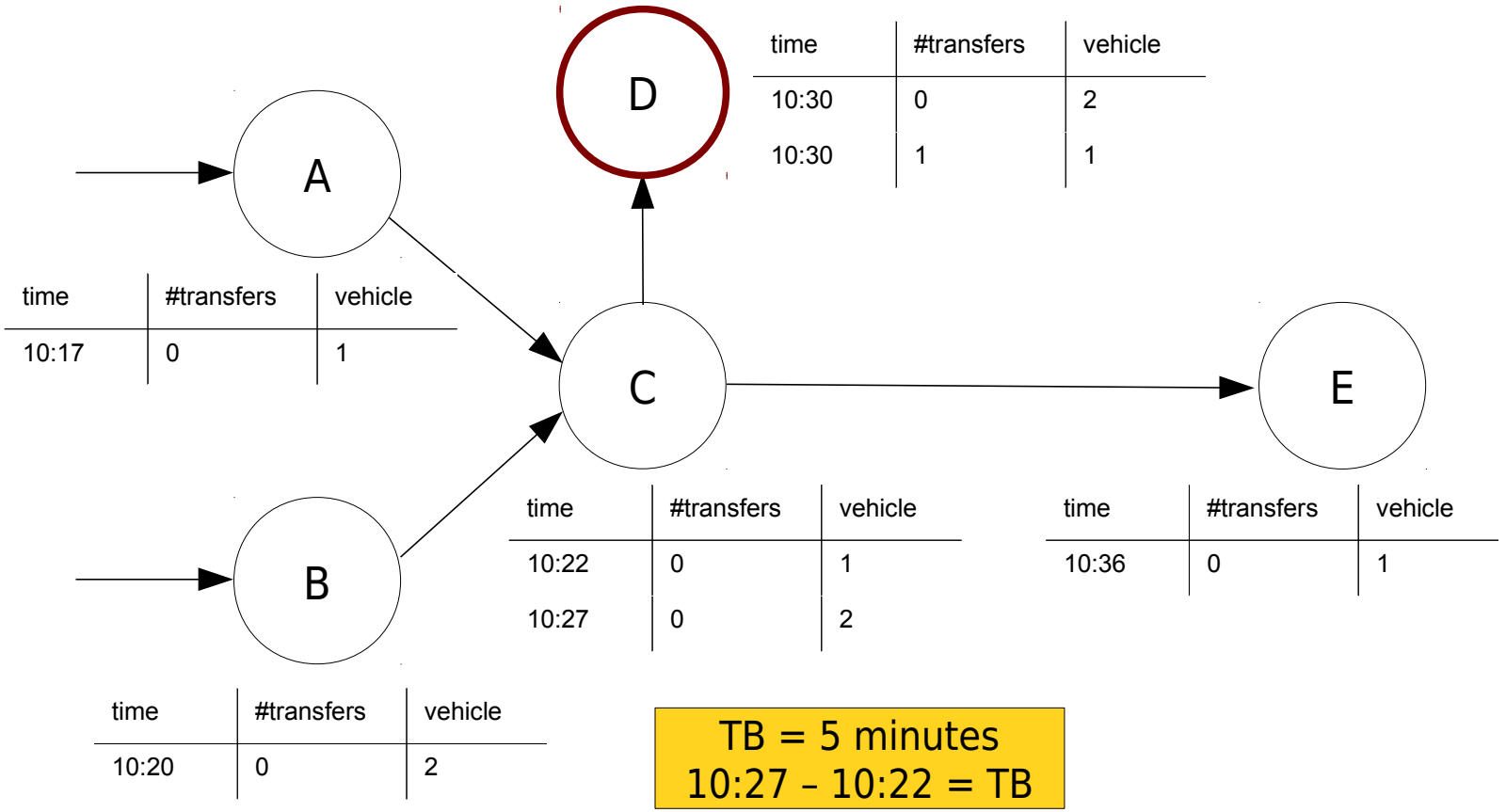
Time-Dependent Problem 2

- Keeps labels where p = p'



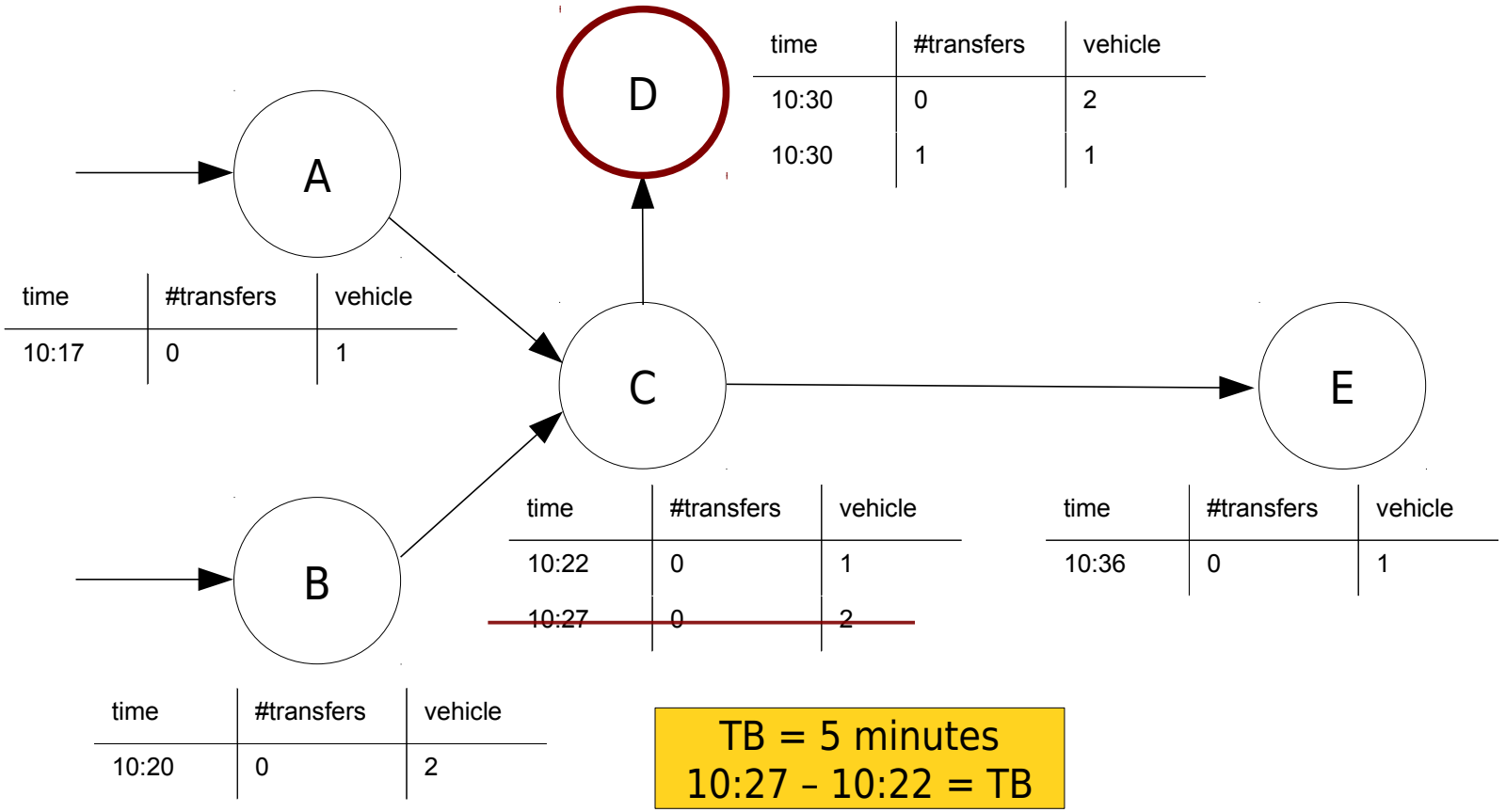
Time-Dependent Problem 2

- Keeps labels where p = p'



Time-Dependent Problem 2

- Keeps labels where p = p'



Time-Dependent Problem 2

- Keeps labels where p = p'