IEEE

ICNP 2008

Routing in Intermittently Connected Sensor Networks

Lu Su*, Changlei Liu†, Hui Song‡ and Guohong Cao†

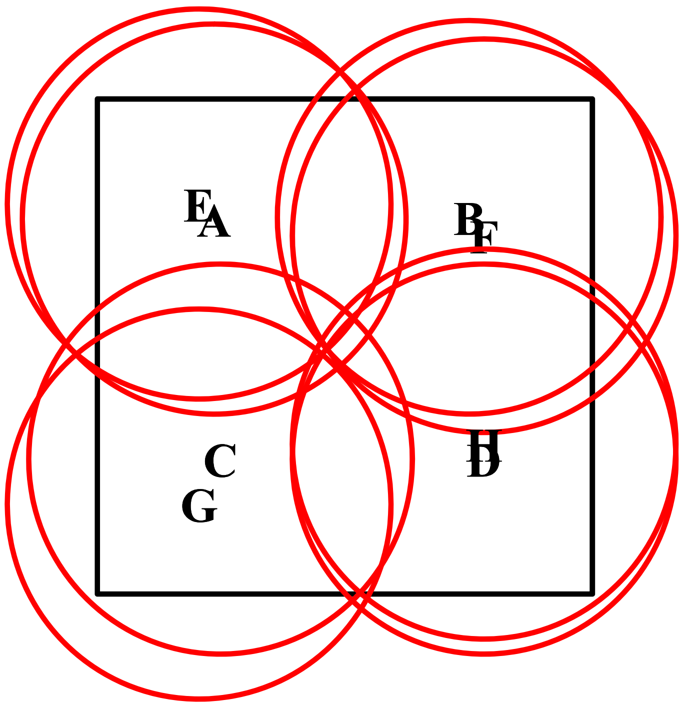
***University of Illinois at Urbana-Champaign**

‡ Frostburg State University

†The Pennsylvania State University

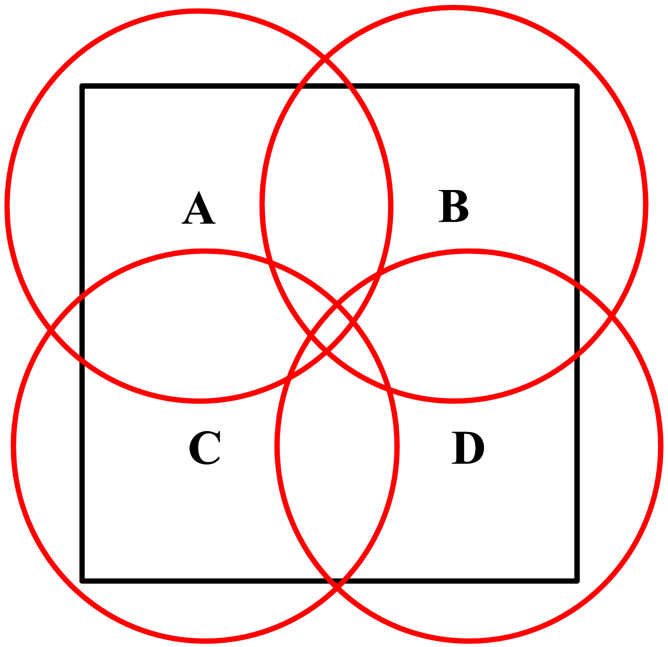
Scheduling in Sensor Networks

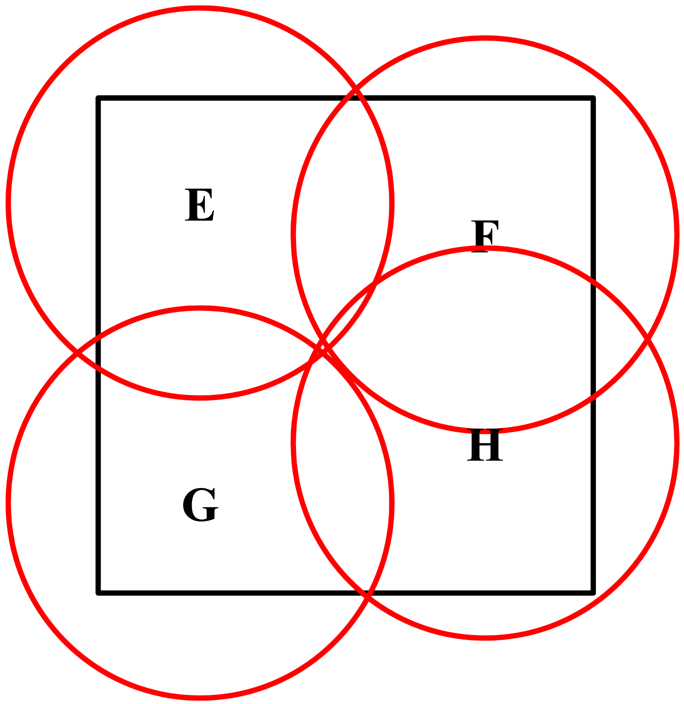
- Deploy more sensors than required.



Scheduling in Sensor Networks

- Deploy more sensors than required.
- A small number of sensors are active at any time.
- Sensors perform the given task in turn.



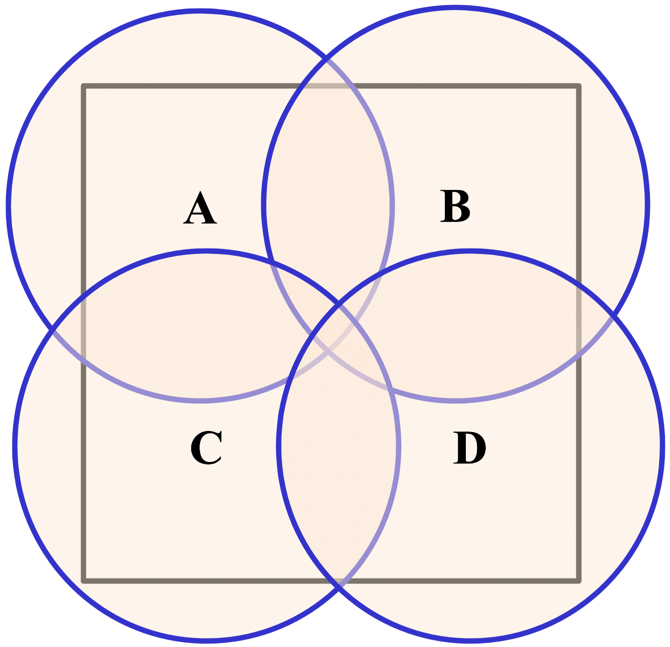


Categories of Scheduling Schemes

- **Full coverage oriented scheduling.**
- **Partial coverage oriented scheduling.**
- **Point coverage oriented scheduling.**

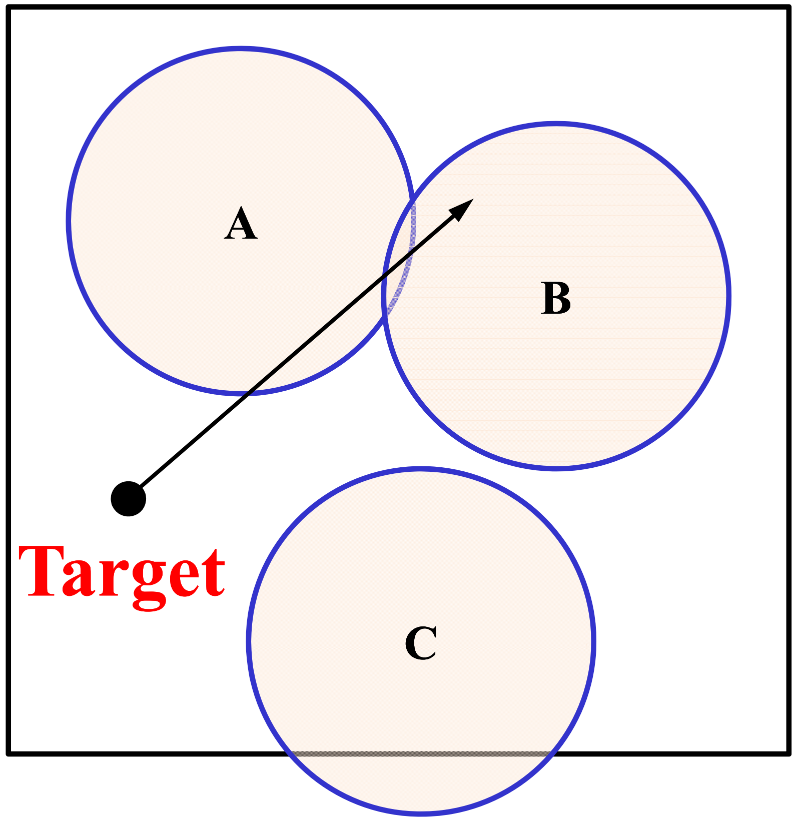
Full coverage oriented scheduling

- **Requirement: Each point inside the deployed region is covered by at least one sensor node.**
- **Application: Forest fire alarming**



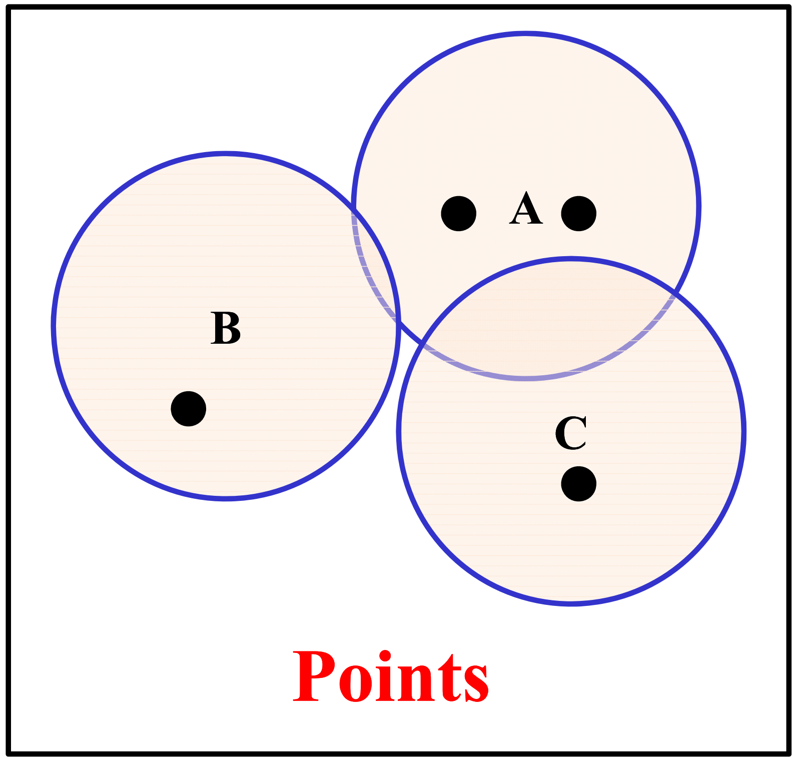
Partial coverage oriented scheduling

- **Requirement**: The scheduling scheme should provide satisfactory **detection delay**.
- **Application**: Mobile target tracking



Point coverage oriented scheduling

- **Requirement: All points** are continuously observed
- **Application:** monitor a set of targets



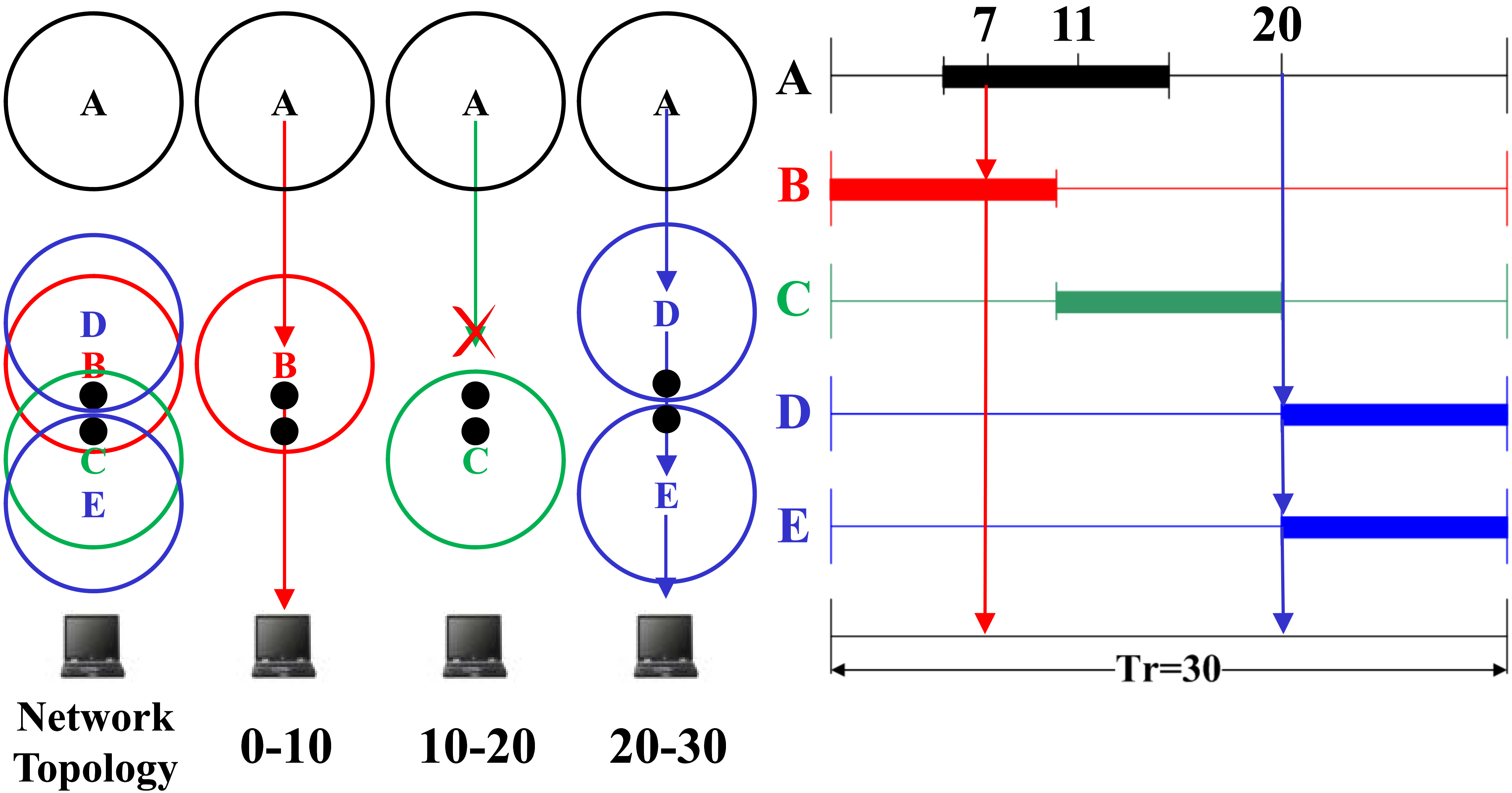
Problem of Scheduling

- The number of active nodes is significantly reduced.
- Some scheduling strategies such as partial coverage scheduling and point coverage scheduling, may result in a sparse distribution of active nodes.
- The network may not be connected at some instant due to the low density of active nodes.

When the network is not fully connected

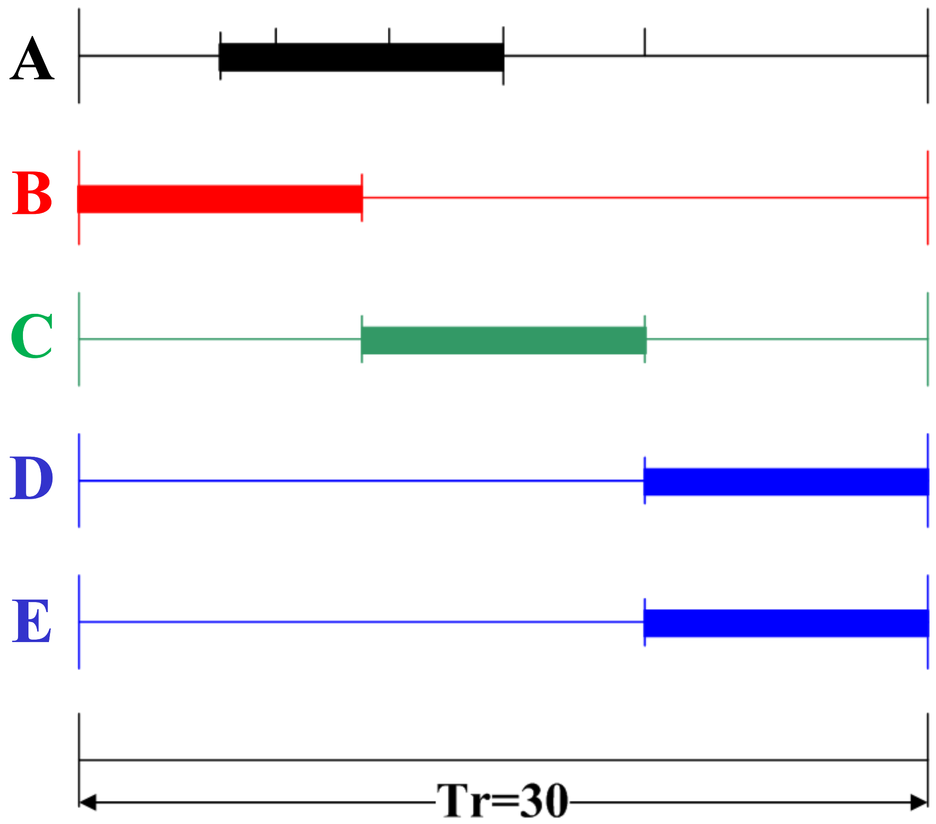
- Traditional routing algorithms may fail.
- The metric of hop number may not work well because the shortest path may not be able to achieve minimum end-to-end packet delivery delay in intermittently connected sensor networks.

Example of intermittently connected networks



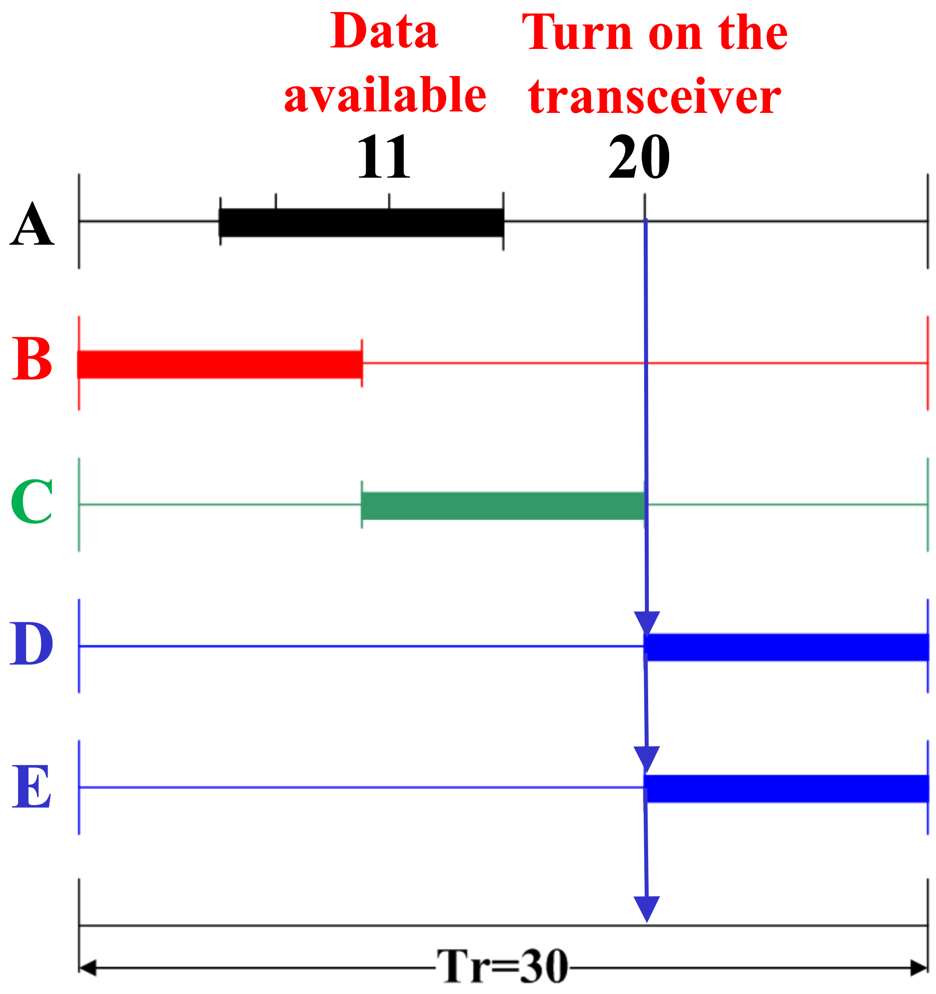
System Model

- Lifetime: Rounds (Tr)
- Round
  - Time slots
  - Working period (Tw)
  - Sleep period



System Model

- **Problem**: A sensor's working period may not overlap with that of its neighbors.
- **Sending rule**: Source node can wake up its transceiver outside its working period and sends packets to its next hop neighbor.
- **Receiving rule**: Each node receives packets within the working period.

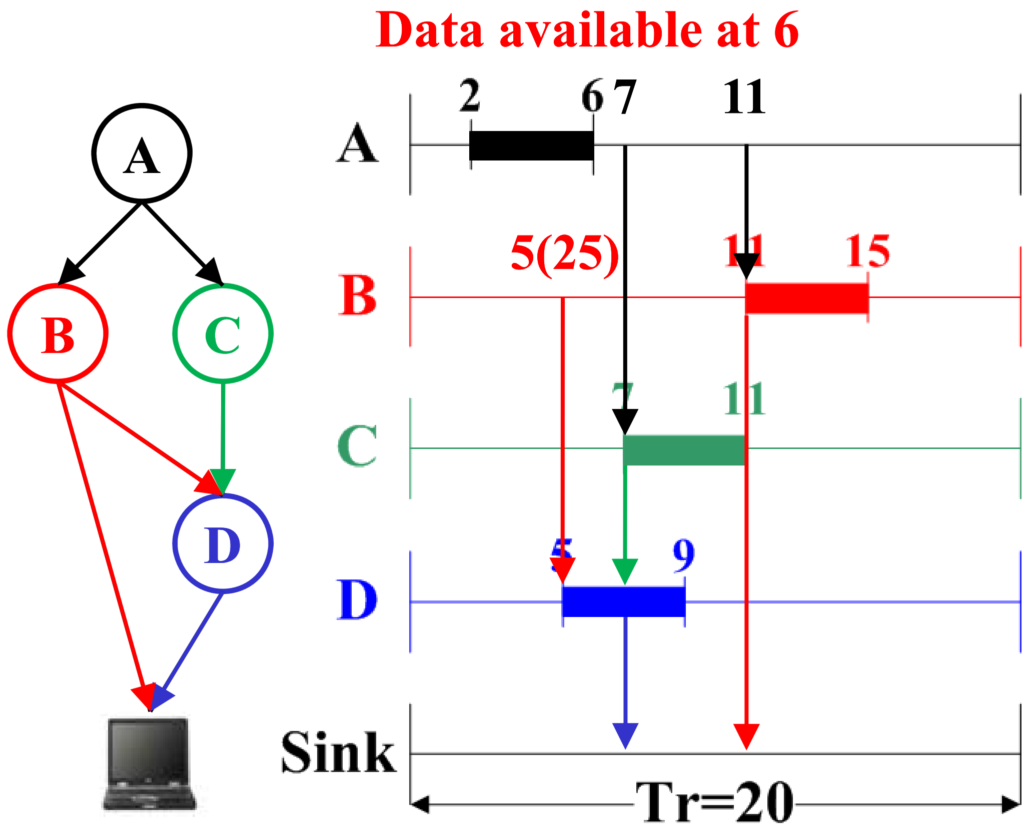


On-Demand Minimum Latency routing algorithm (ODML)

- **Input**
  - **Source node**
  - **Destination node**
  - **Data available time**
- **Output**
  - **Minimum latency route**

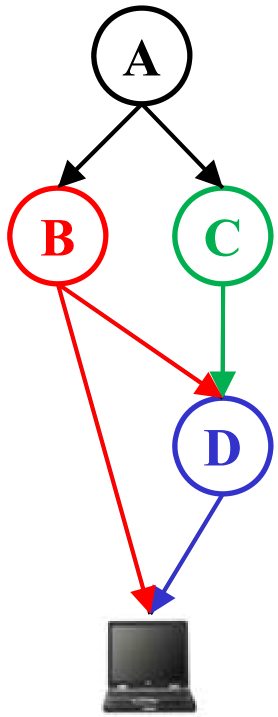
ODML

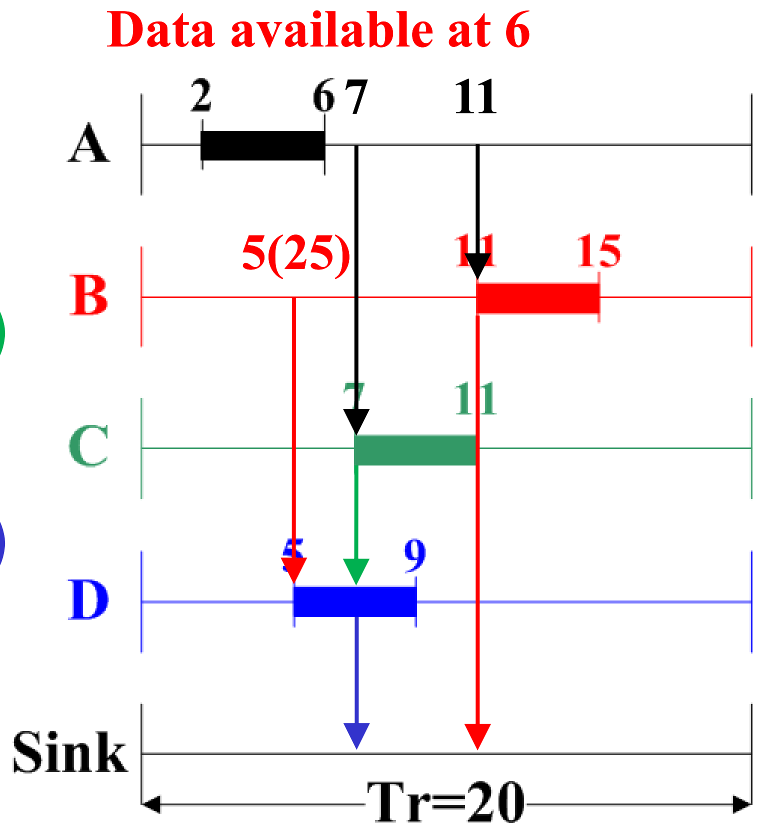
- The source sends a route request (RREQ) to each neighbor.
- Each intermediate node updates the latency of RREQ by adding the buffer delay of the next link to the original value.
- Then, it unicasts the request to other neighbors except the one from which it receives RREQ.



ODML

- The **first RREQ** arriving at the destination went through **the minimum latency route**.
- The destination node then sends a **route reply** (RREP) back along this route.
- Each node along the route update the **routing information**.





ODML

Arrival	Next Hop	Destination	Latency
6	C	sink	1

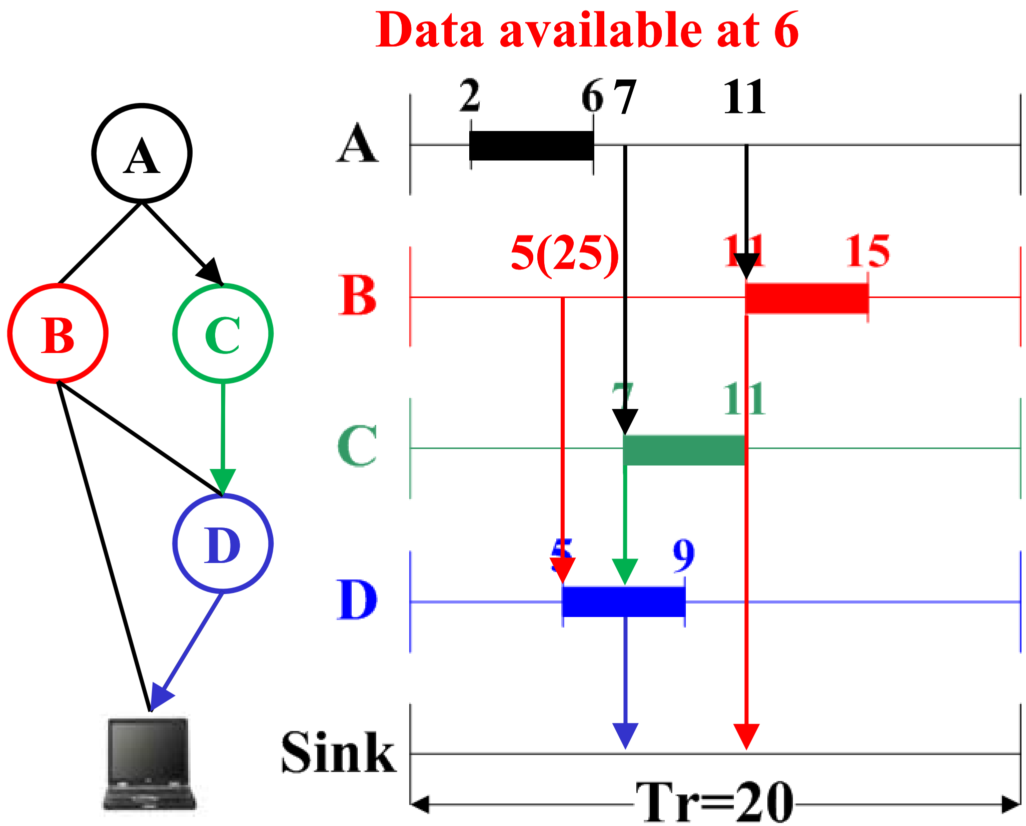
A's Table

Arrival	Next Hop	Destination	Latency
7	D	sink	0

C's Table

Arrival	Next Hop	Destination	Latency
7	sink	sink	0

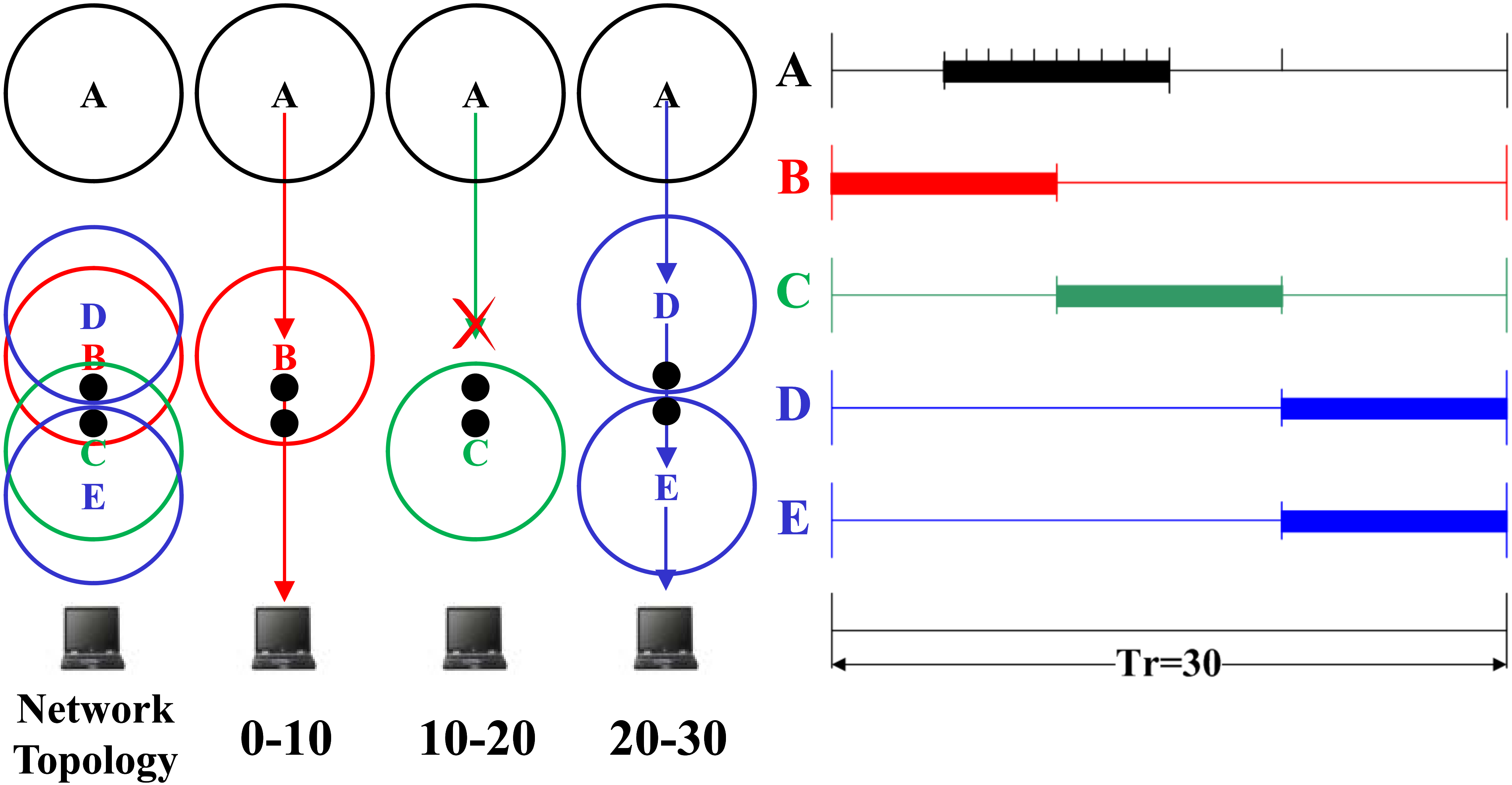
D's Table



Drawback of on-demand Routing

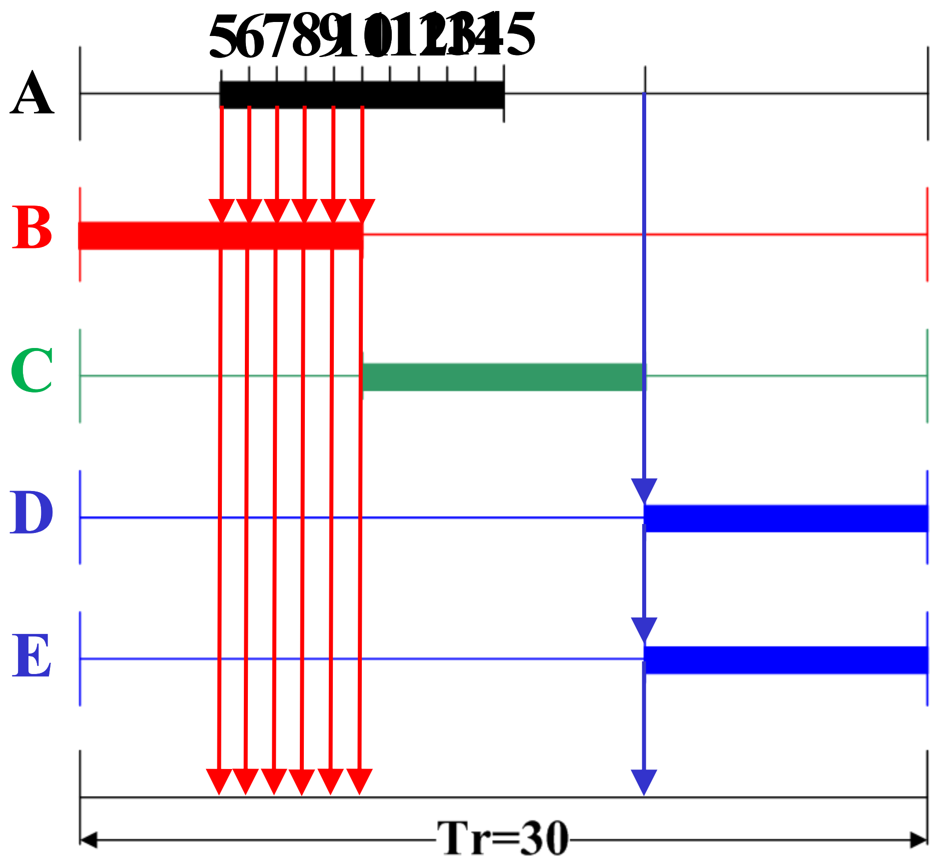
- on-demand routing does not work well when the source and the destination frequently communicate with each other.
- In intermittently connected network, the network topology varies with time, as a result, the minimum latency route may also change accordingly.
- Since the route provided by ODML is only optimal at the time when the packet is ready for sending, if the source node wants to send another packet at different time, it has to invoke ODML again.

Example Revisit



Example Revisit

- **If there is a packet available at each time point within the working period of node A.**
- **A has to run ODML 10 times to find minimum latency route for each packet.**
- **This is not necessary because the minimum latency route does not change within some consecutive time slots**



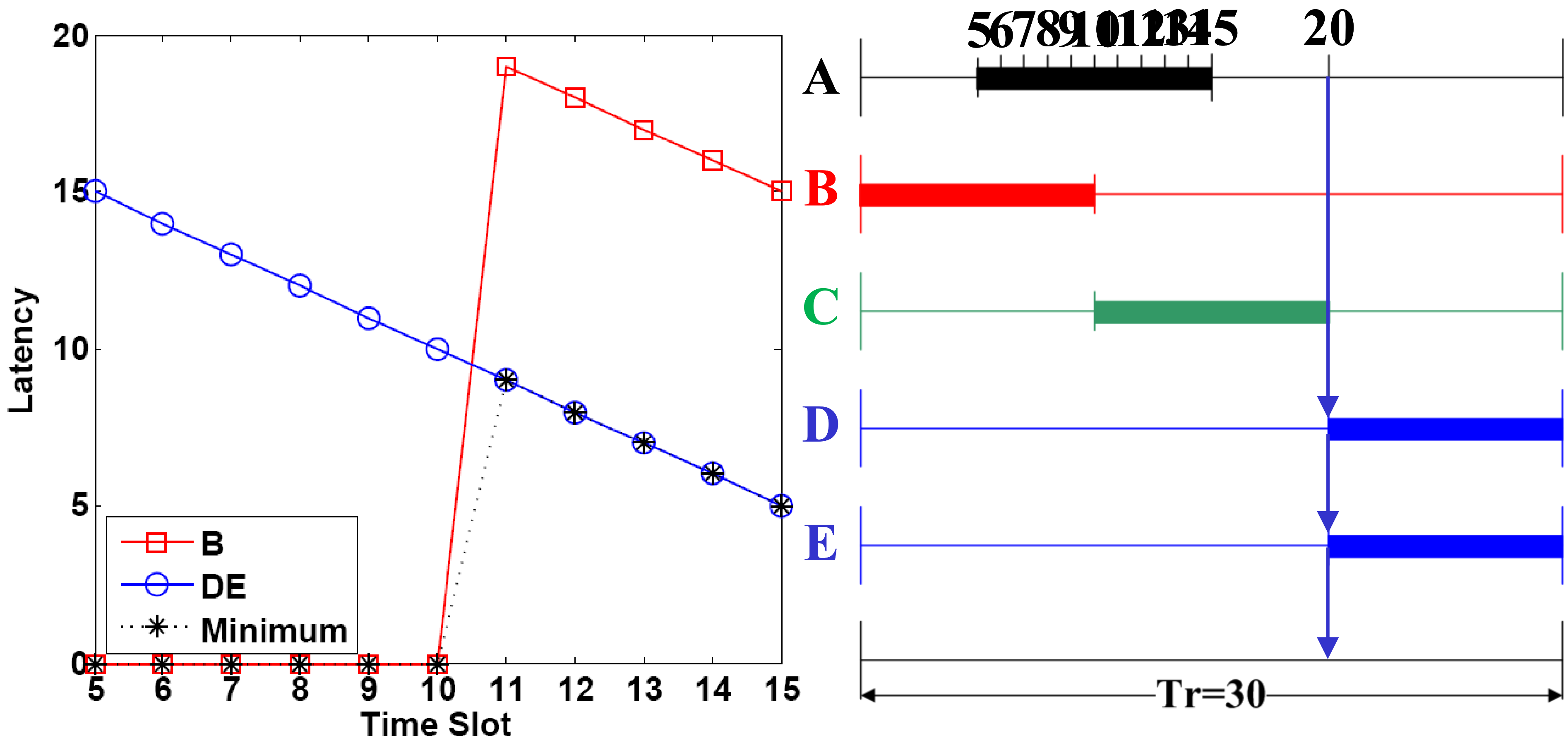
Proactive minimum latency routing

- In proactive minimum latency routing, we aim to find the time points where the minimum latency route changes.
- Route Transition Point (RTP): the time point when the minimum latency route is about to change.
- Two approaches
  - Optimal Proactive Minimum Latency Routing algorithm (optimal-PML)
  - Quick Proactive Minimum Latency Routing algorithm (quick-PML)

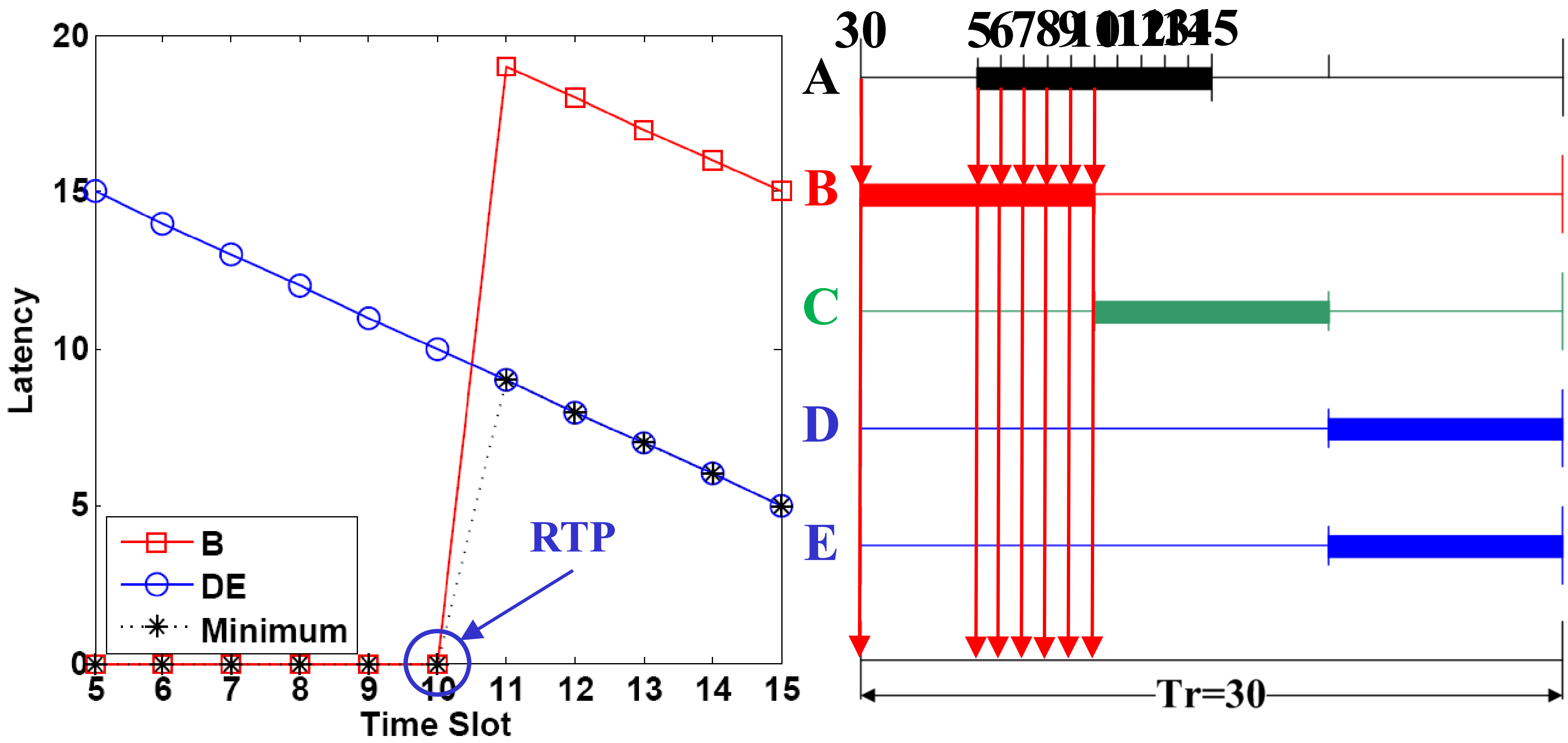
Optimal Proactive Minimum Latency Routing algorithm (optimal-PML)

- The goal of optimal-PML is to find the route transition points.
- If we can identify which points are likely to become the route transition points and which points are not, the search space can be significantly reduced.
- To achieve this, let's observe the latencies of routes in previous example.

Latency of route A→D→E→Sink



Latency of route A→B→Sink



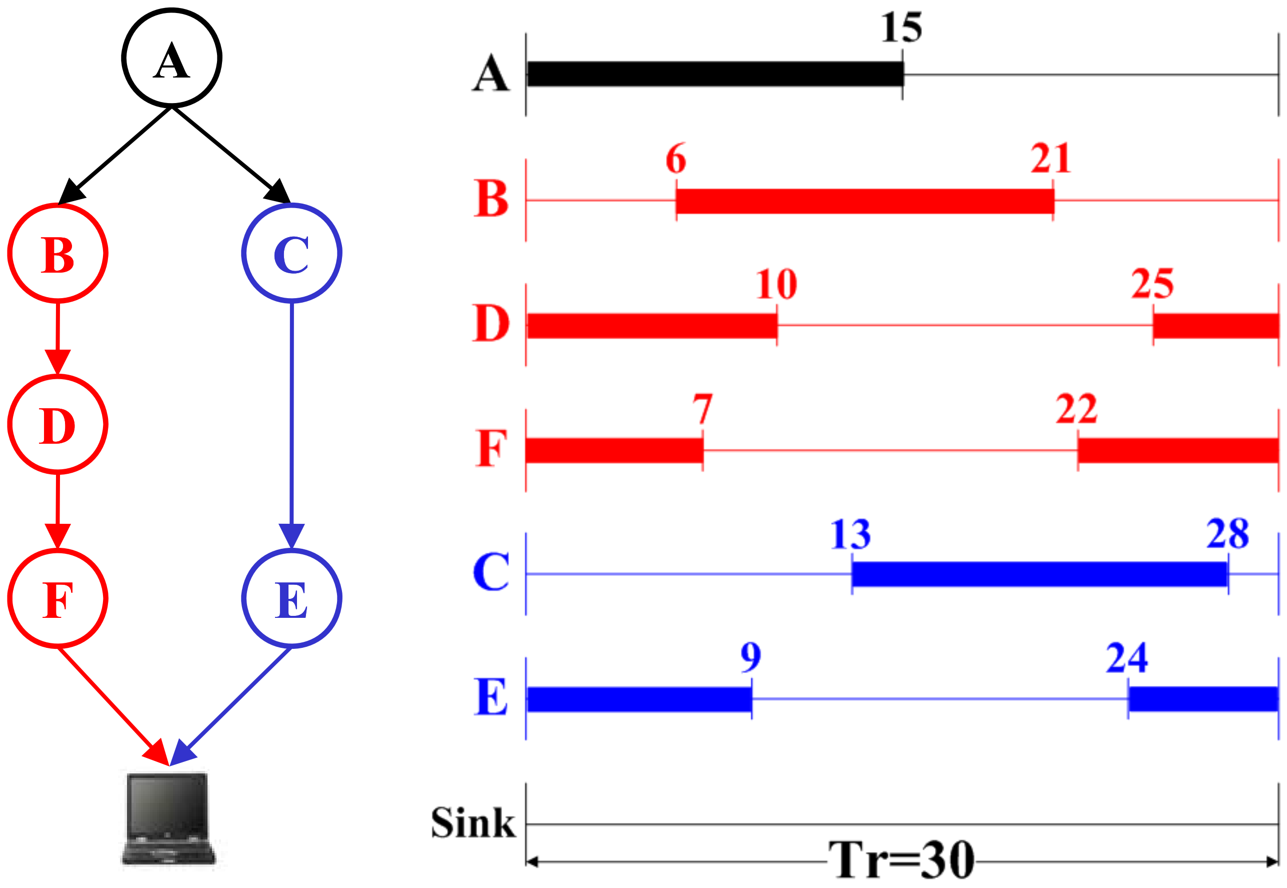
Sleep Transition Point (STP)

- **Sleep Transition Point (STP): end points of working periods of the nodes along a route.**
- **Theorem 2: a route transition must happen at a sleep transition point.**
- **Therefore, to find RTP, only the sleep transition points of the current MLR need to be checked.**

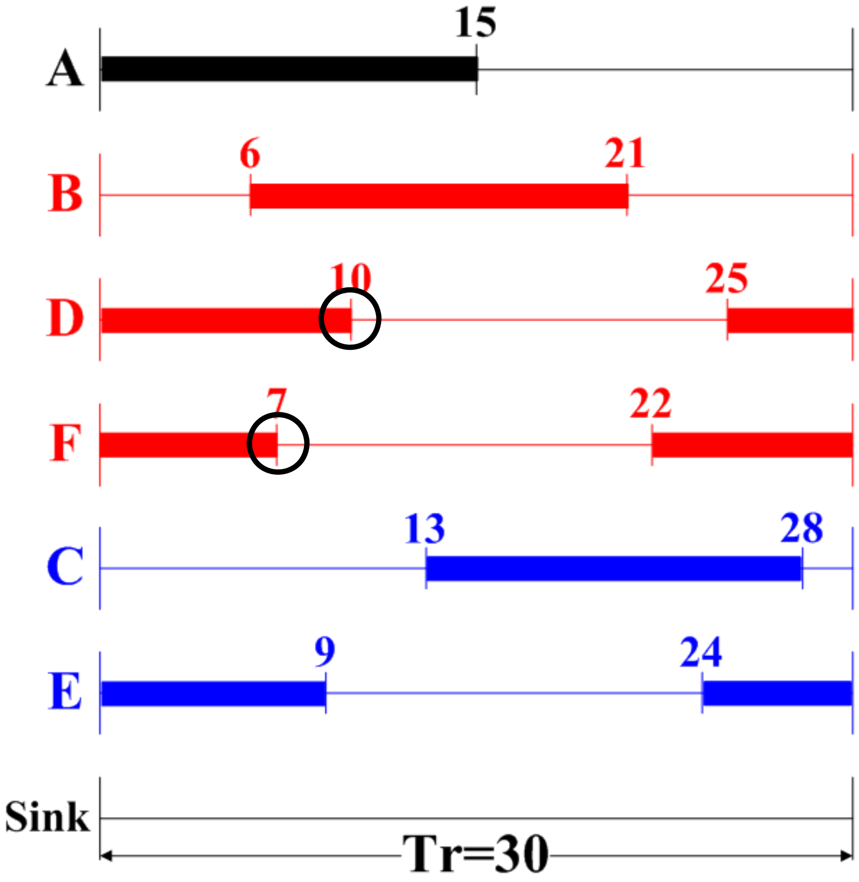
optimal-PML

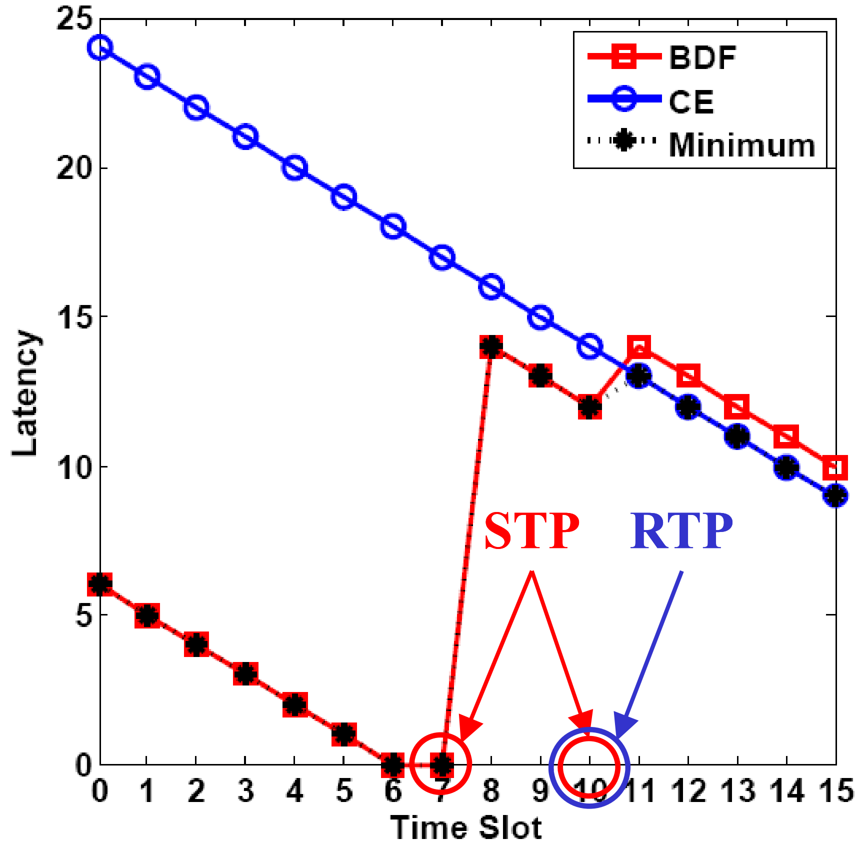
- **Sequentially invokes ODML at STPs of the current minimum latency route.**
- **If the route found by ODML is different from the current MLR, a RTP is identified.**
- **This process repeats until all the RTPs are found.**

Example of optimal-PML



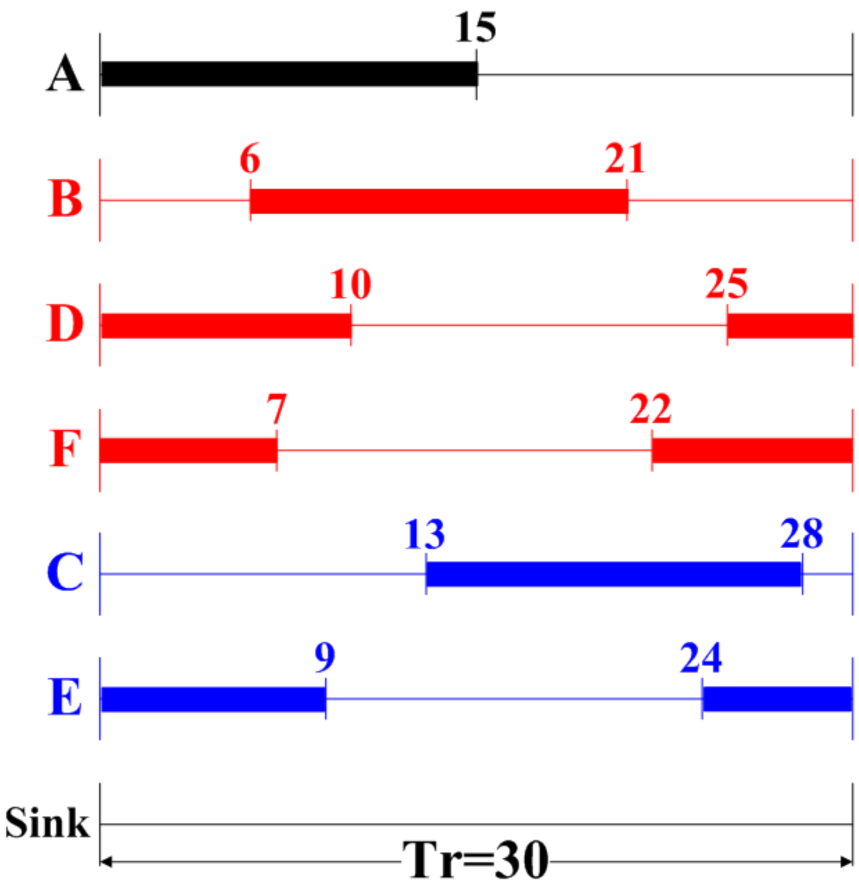
Example of optimal-PML





STP is not necessarily RTP !

Example of optimal-PML



Arrival	Next Hop	Destination
0	B	Sink
11	C	Sink

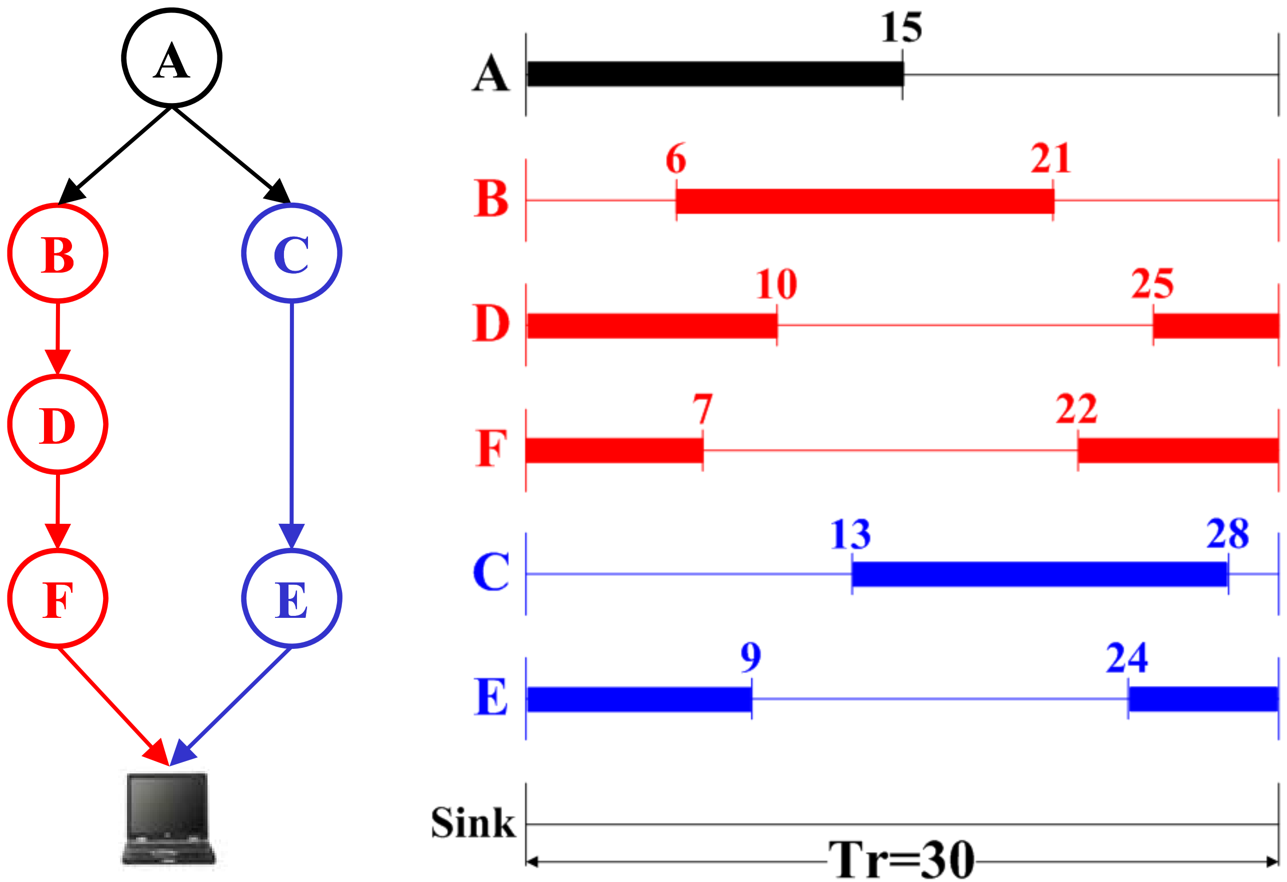
Quick Proactive Minimum Latency Routing algorithm (quick-PML)

- **Although optimal-PML can minimize the call number of ODML, it has long route acquisition delay.**
- **Because optimal-PML can find only one minimum latency route in each route discovery round.**
- **quick-PML applies multiple route discoveries simultaneously.**

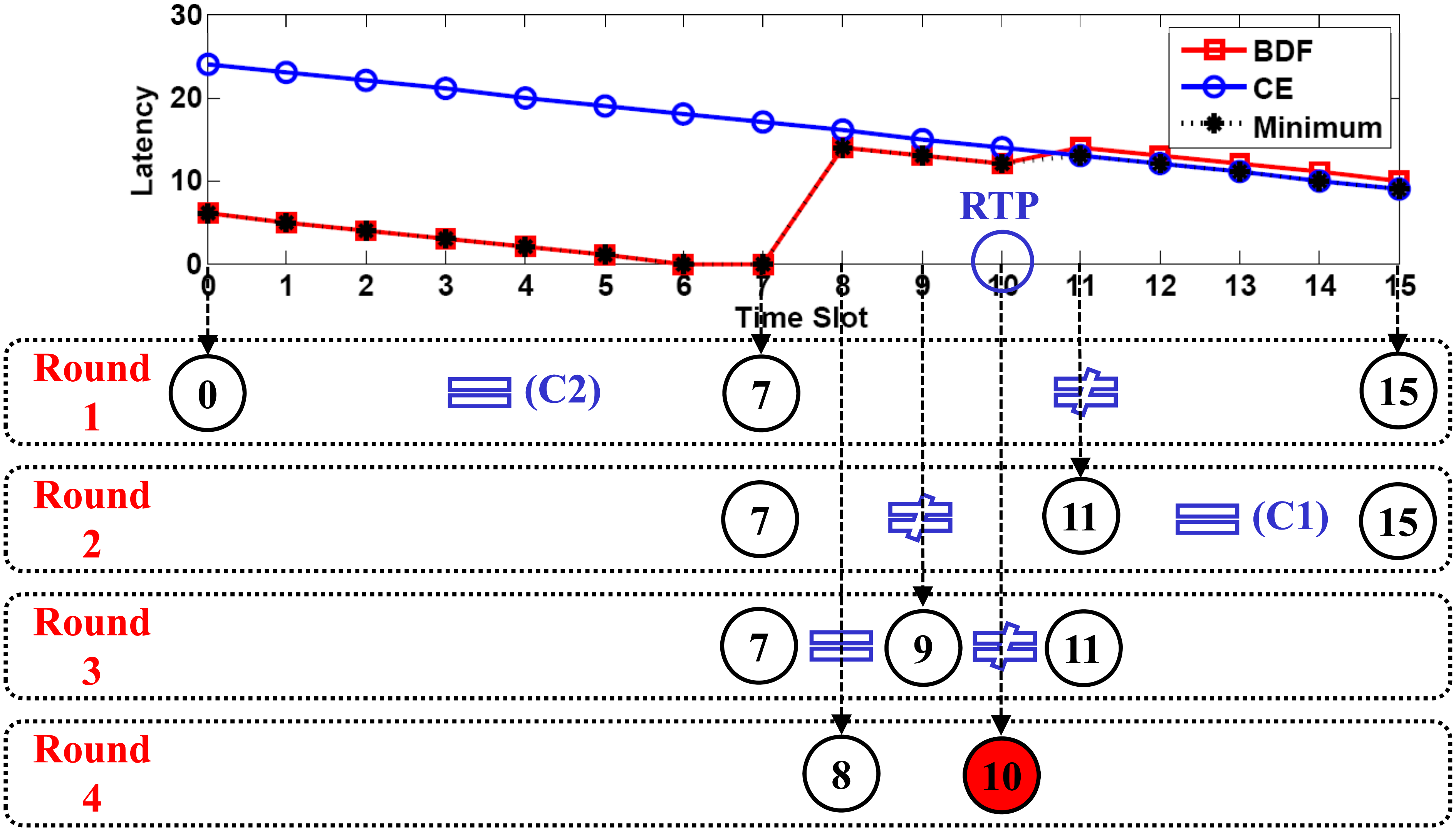
quick-PML

- Assumption: If the minimum latency routes at two time points are identical, there is no route transition point between these two points.
- It is true when one of the following two conditions is satisfied:
  - 1) $MDL(t2) > 0$ and $MDL(t1) - MDL(t2) = (t2 - t1)$
  - 2) $MDL(t2) = 0$ and $t2 - t1 < Tr - Tw$

Example of quick-PML

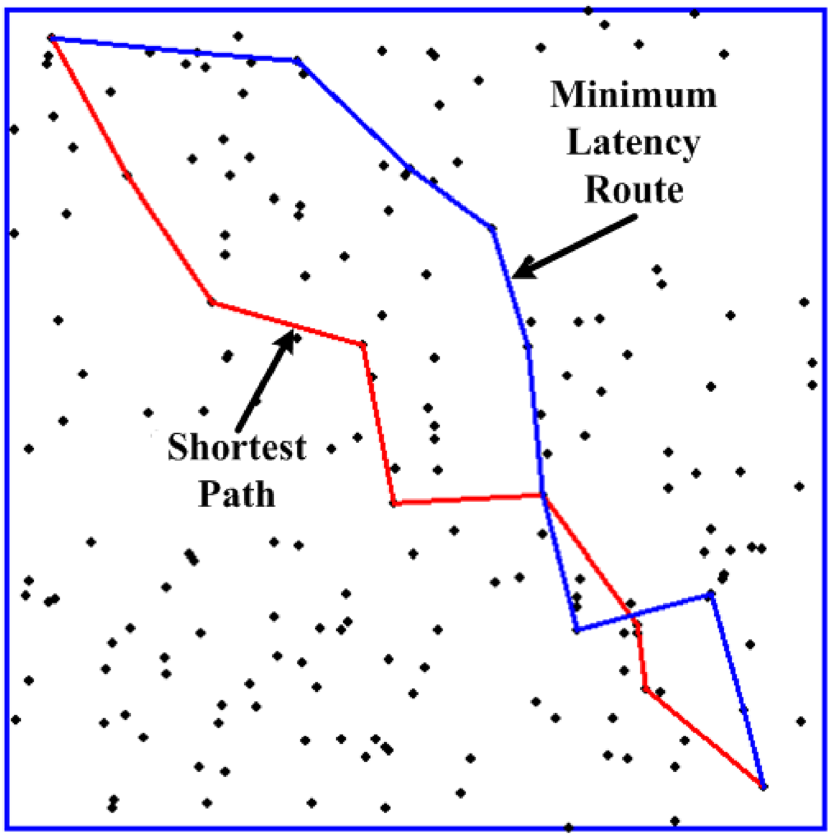


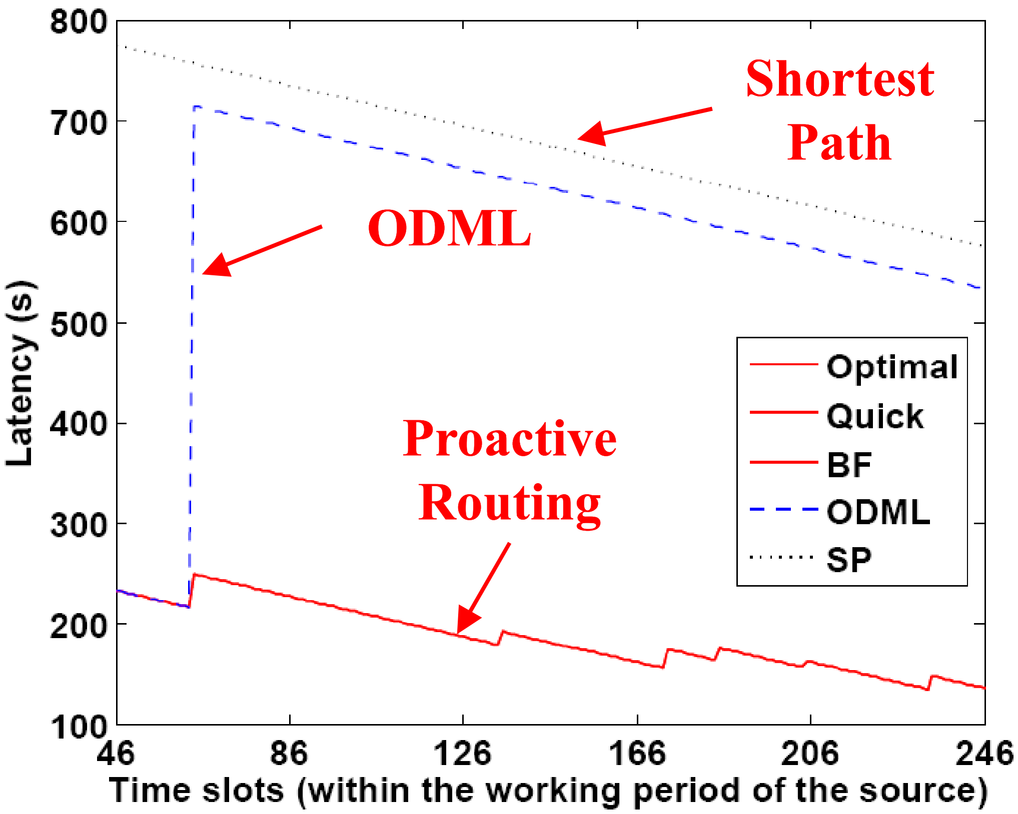
Example of quick-PML



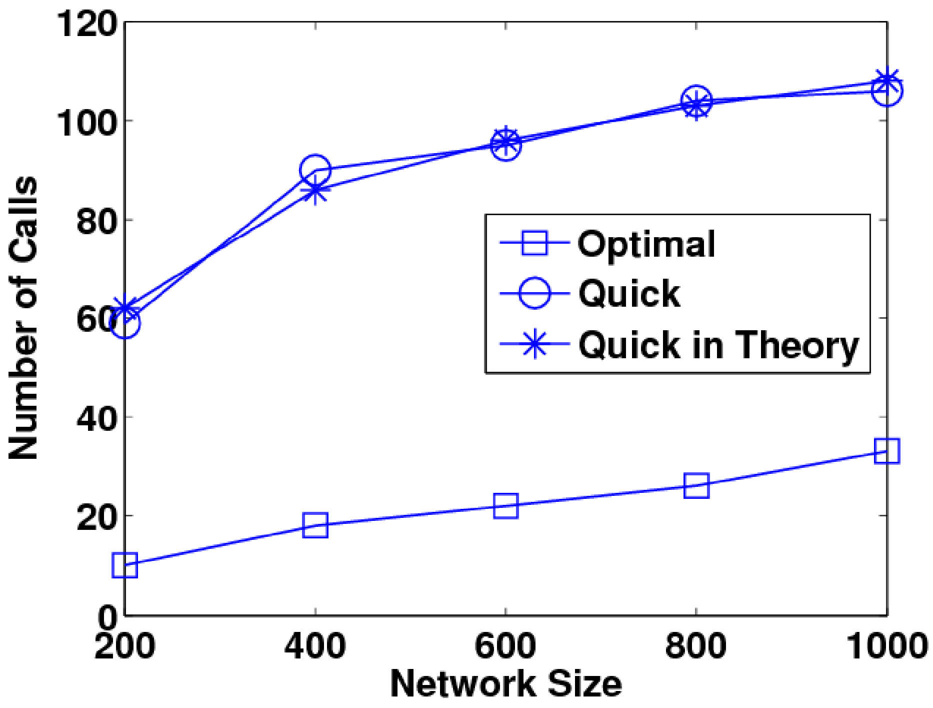
Performance Evaluations

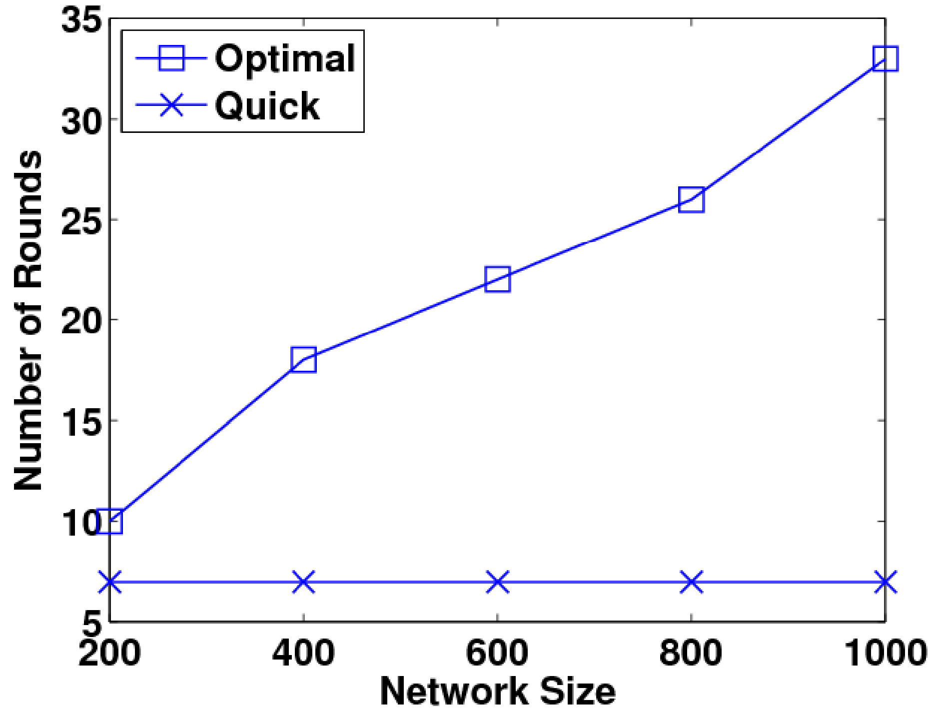
- **A Scenario**





optimal-PML VS quick-PML





Conclusion

- Identify the challenges of routing in intermittently connected sensor networks.
- Propose an on-demand minimum latency routing algorithm (ODML) to find minimum latency routes.
- Proposed two proactive minimum latency routing algorithms: optimal-PML and quick-PML.
- The schemes proposed in this paper can provide generic routing functionalities for most of the existing scheduling schemes.

Thank You !