IEEE INFOCOM 2016

IEEE 35th International Conference on Computer Communications (INFOCOM'16)
10-15 April 2016 San Francisco, CA, USA

Mobility-Aware Real-Time Scheduling for Low-Power Wireless Networks

- **Behnam Dezfouli**
- Marjan Radi
- Octav Chipara

Department of Computer Science
The University of Iowa

Contact:
http://behnam.dezfouli.com
dezfouli [at] ieee [dot] org

The University of Iowa

Non Real-Time vs Real-Time Wireless Networks

Non Real-Time Networks

- Provide a best-effort service
- No guarantee of timeliness or reliability
- Network dynamics affect the service provided

Nodes contend for transmission whenever they have data

- For example: Connecting devices using WiFi

Real-Time Networks

- Packets should be delivered in a timely and reliable manner
- Network dynamics do not affect the service provided

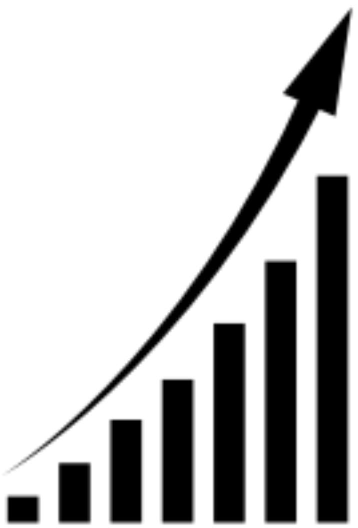
Nodes' transmission schedules are predetermined

- For example: Connecting devices using WirelessHART

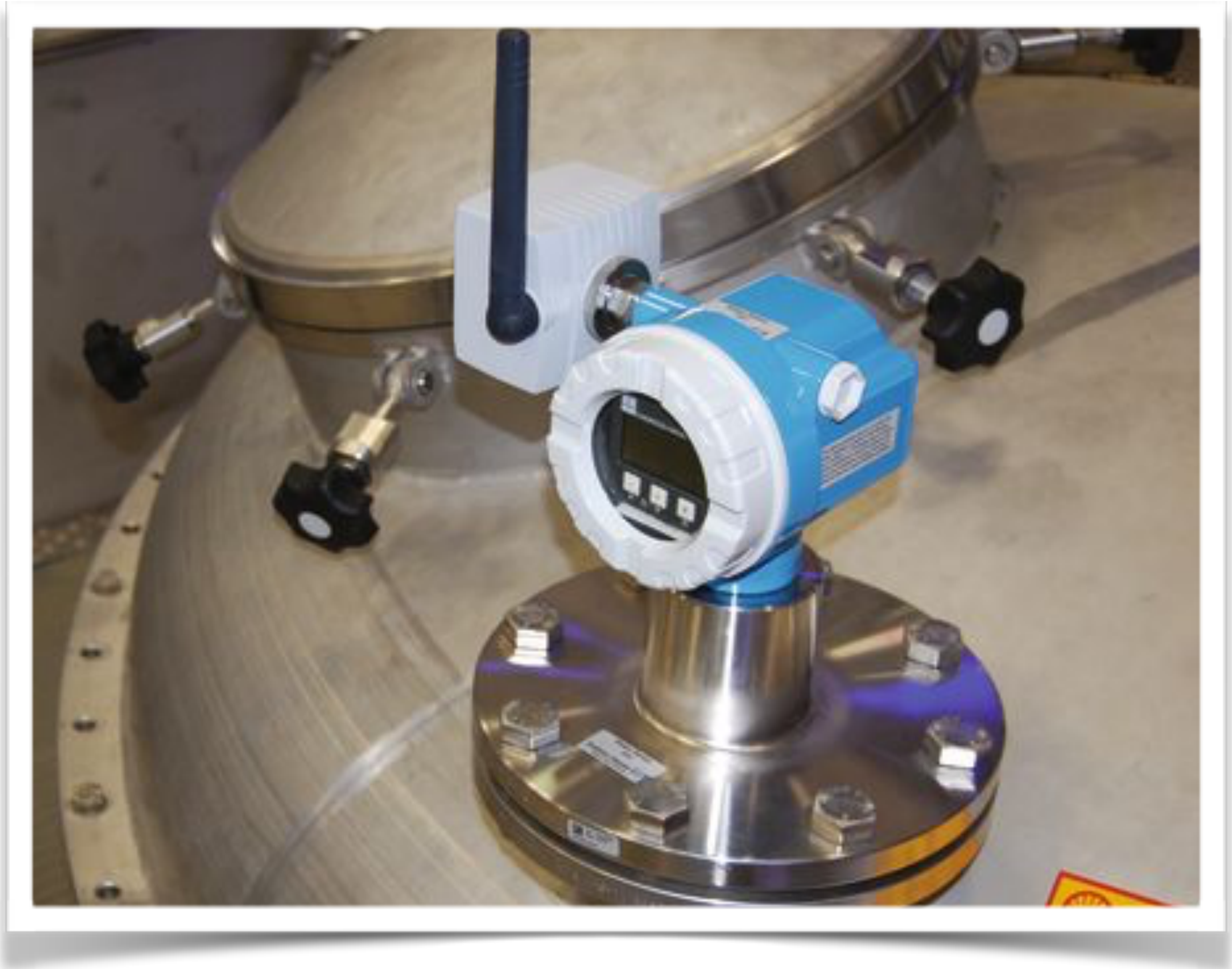
Industrial Real-Time Wireless Networks

Make wireless technology an attractive solution for process monitoring and control applications

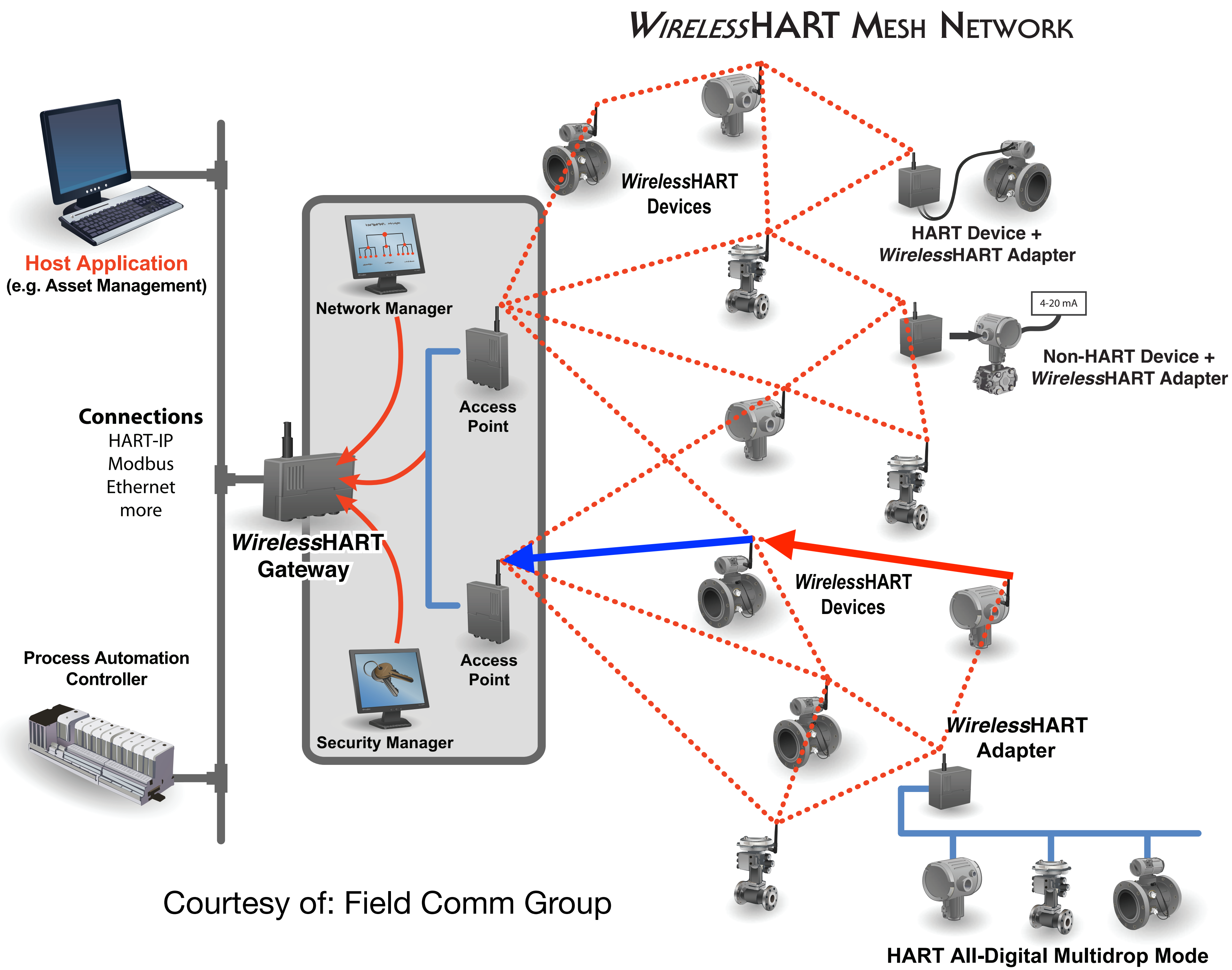
- **Reducing the cost**
- **Simplifying the deployment**



- Wireless devices in industrial applications: **annual growth rate of 27.2%**
- ***43.5 million devices by 2020***



WirelessHART



Courtesy of: Field Comm Group

WirelessHART

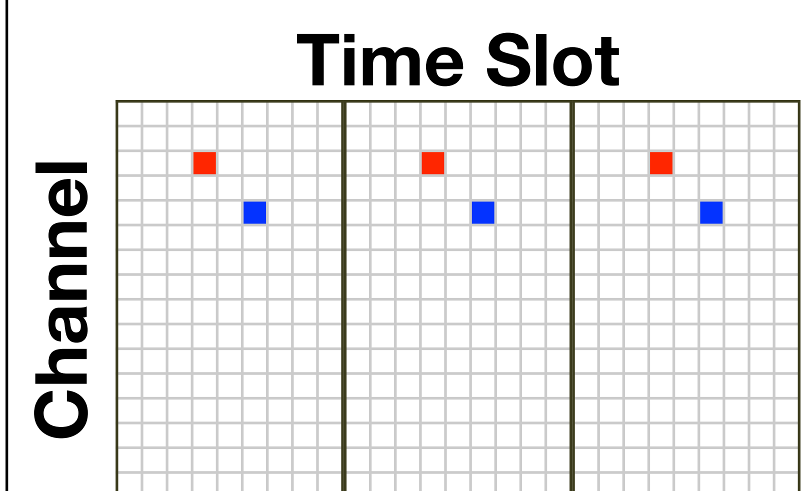
Central Network Management

Gateway **is responsible for managing medium access schedules**

Through *time slot* and *channel* assignment

(FTDMA: Frequency-Time Division Multiple Access)

The schedules assigned to the red and blue links



Why Centralized Medium Access Scheduling?

1. **Shortcomings of contention-based medium access:**
   - Does not guarantee end-to-end delay
   - Significant packet collision and loss

2. **Shortcomings of distributed schedule-based medium access:**
   - Does not guarantee end-to-end delay
   - Moderate packet loss due to intra-network interference

3. **Benefits of centralized schedule-based medium access:**
   - Guaranteed end-to-end delay
   - Avoids packet loss due to intra-network interference

Research Gap

Existing real-time wireless networks assume:
Nodes are stationary, and
The set of traffic flows are fixed

↓

Limits the applicability of these solutions to dynamic applications with mobile entities such as patients, robots, firefighters, etc.

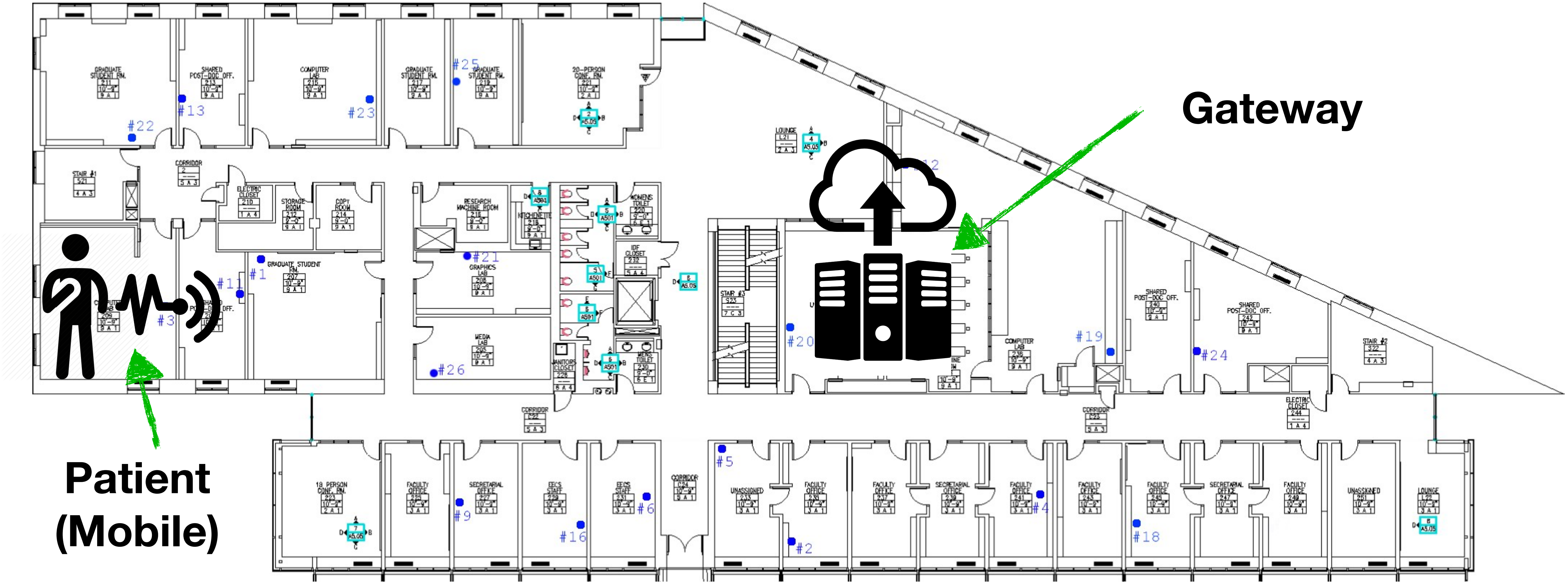
↓

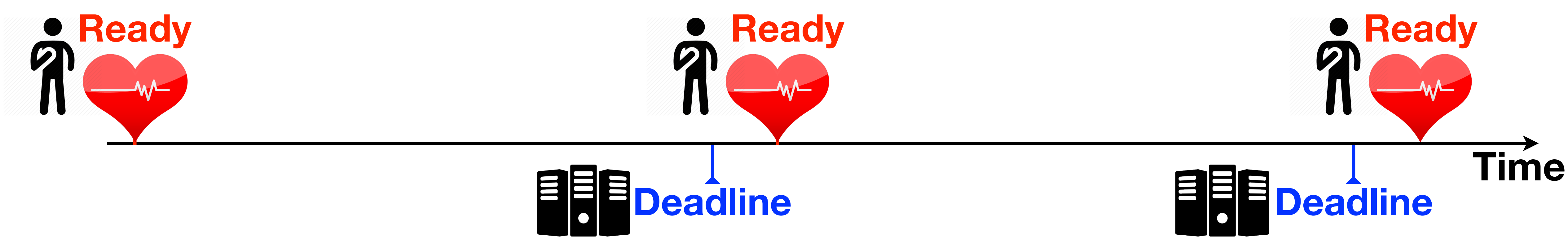
How to support real-time communication with mobile nodes?

Objective

Sample Application

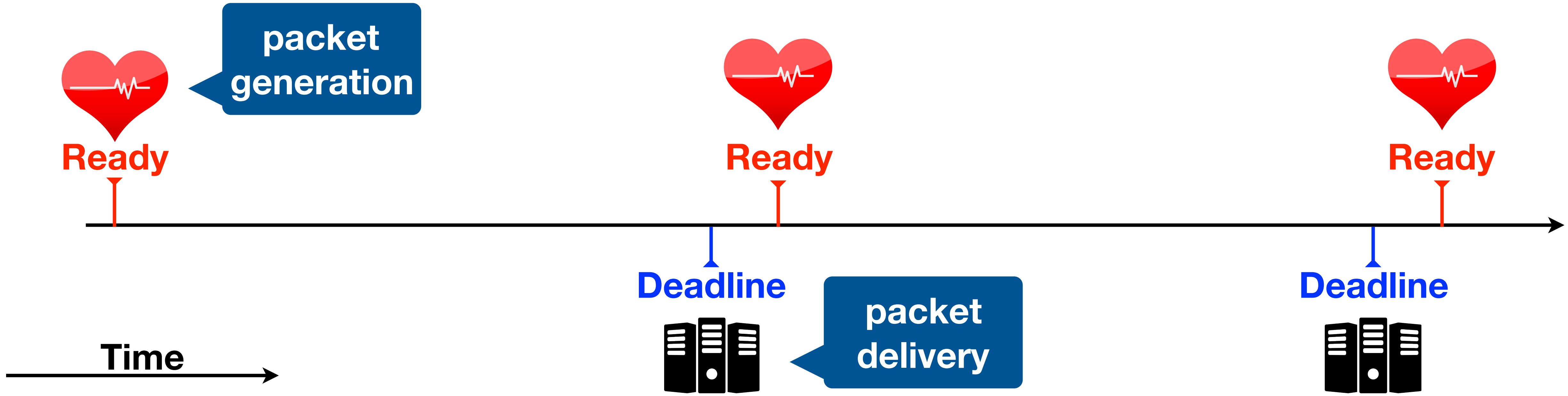
— **Timely and reliable delivery of patients' vital signs to the Gateway**





Basic Assumptions and Requirements

- Each mobile node can generate one or more data flows
- Each flow *i* is characterized by its period (P_i) and deadline (D_i)
- **The mobility pattern of the mobile nodes is unknown**
- **Packets of each data flow should be delivered to the Gateway before their deadline**
- **For example:**
  - A mobile node samples heart rate every 1 sec
  - The sample should be delivered to the Gateway no later than 0.9 sec after its generation



Network Design

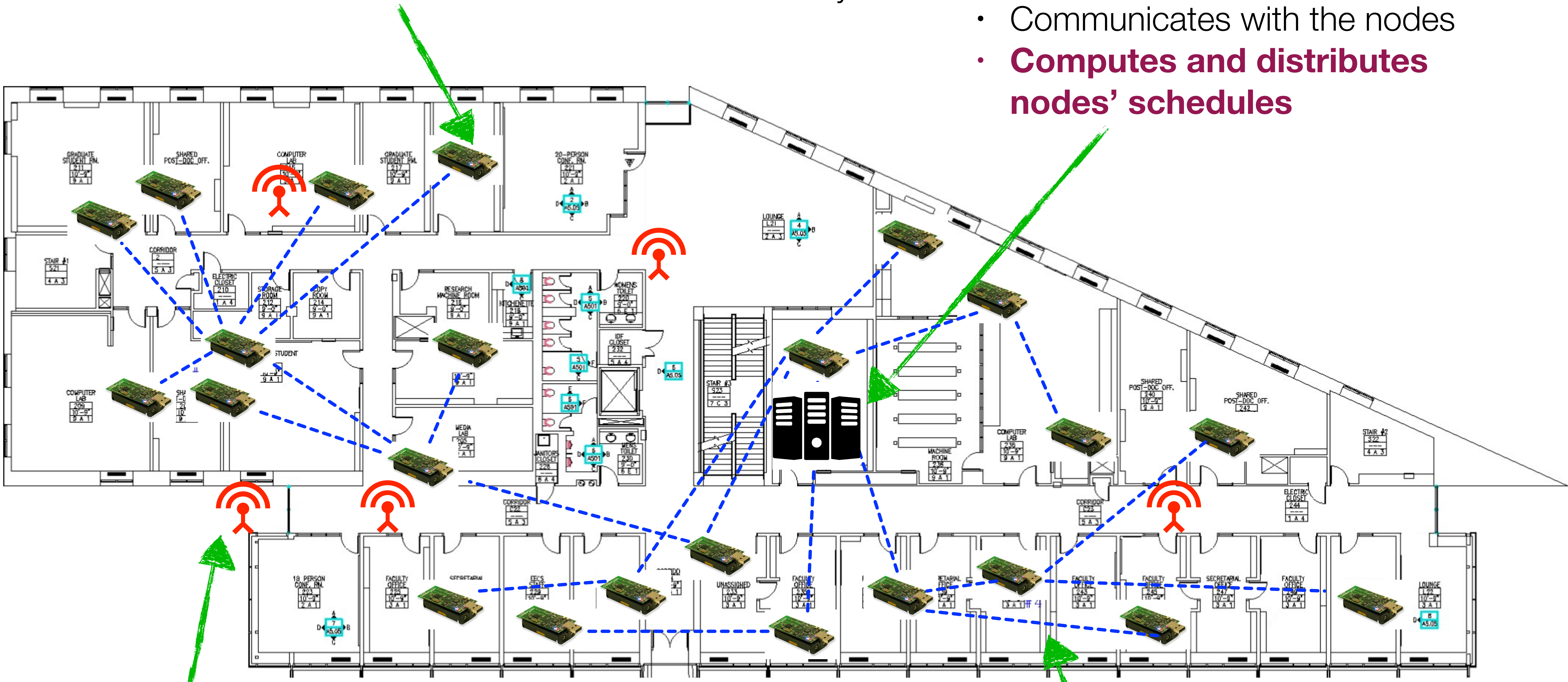
Architecture

A Low-Power Wireless Infrastructure Node

- Communicates in a real-time manner with the Gateway

Gateway

- Communicates with the nodes
- **Computes and distributes nodes' schedules**



A Low-Power Wireless Mobile Node

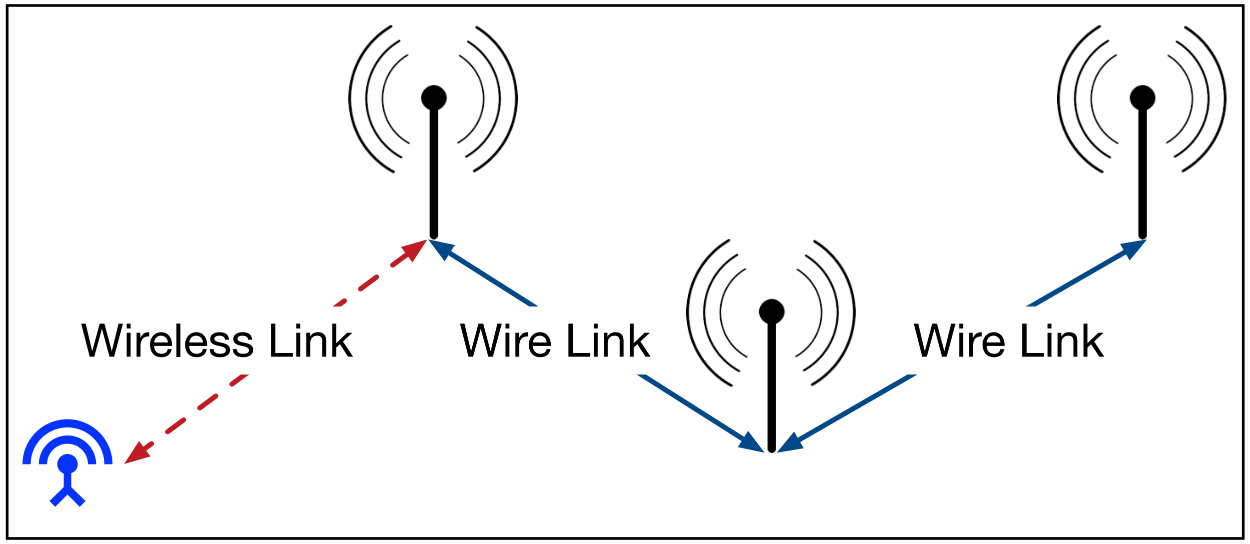
- Communicates in a real-time manner with the Gateway

Communication Between Infrastructure Nodes

Architecture

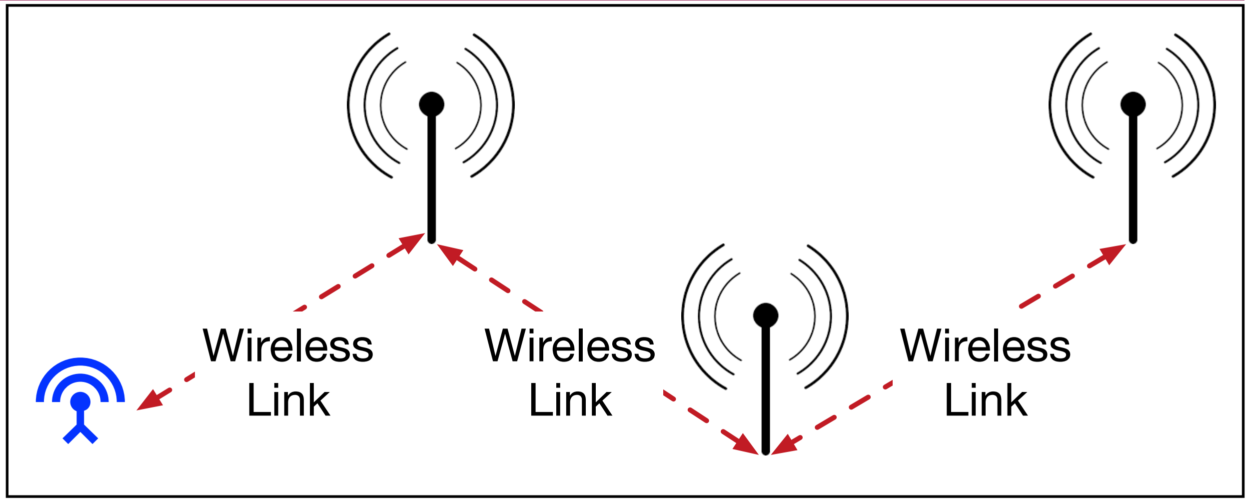
Wired infrastructure

- **Base stations are connected through wire links**
- Similar to cellular (3G, 4G) and most WiFi networks
- ***Hard network deployment***
- Bandwidth reservation only between mobile-infrastructure



***Wireless* infrastructure (our choice)**

- **A multi-hop wireless infrastructure**
- ***Easy network deployment***
- Bandwidth reservation between infrastructure-infrastructure as well as mobile-infrastructure



Implication of Assumptions on Scheduling

Unpredictable mobility paths

+

Low energy consumption:
Short communication ranges

+

The need to deliver data in a timely and reliable manner

}

How these assumptions affect our network design?

Mobility and Data Forwarding Paths

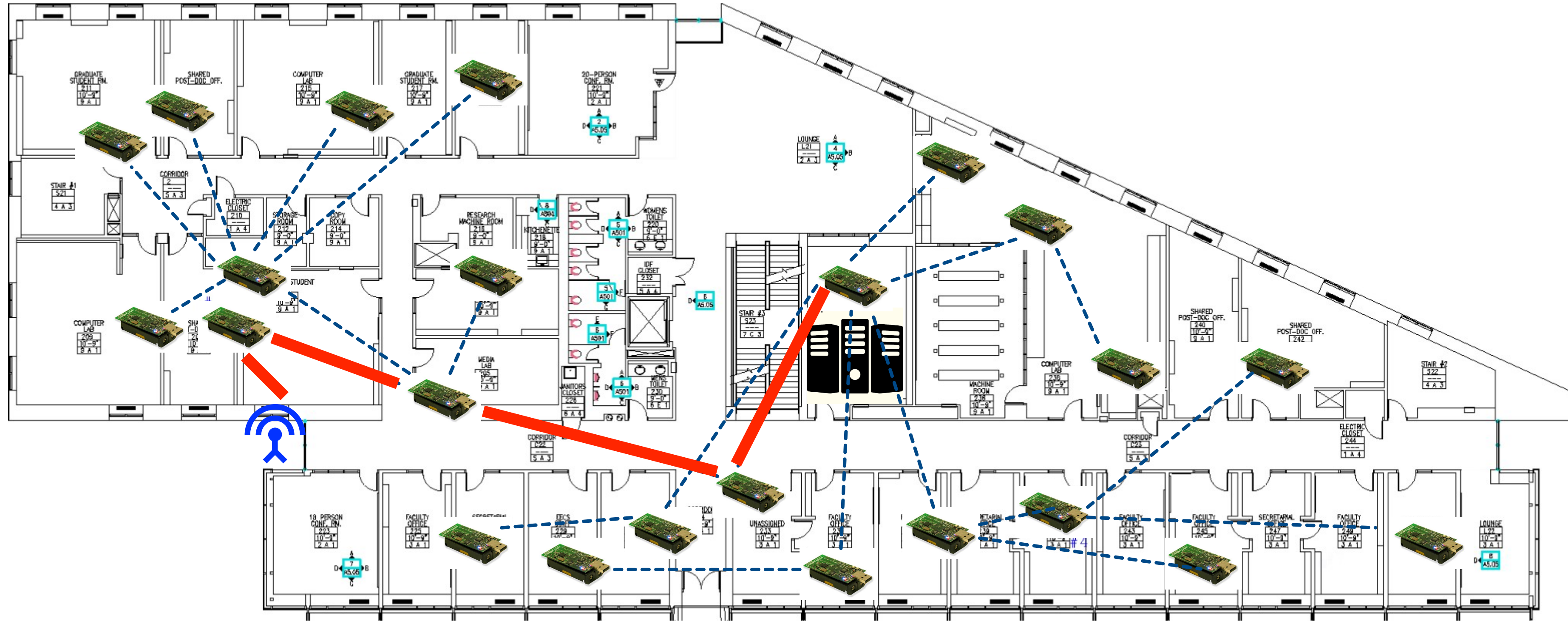
Low Power Consumption

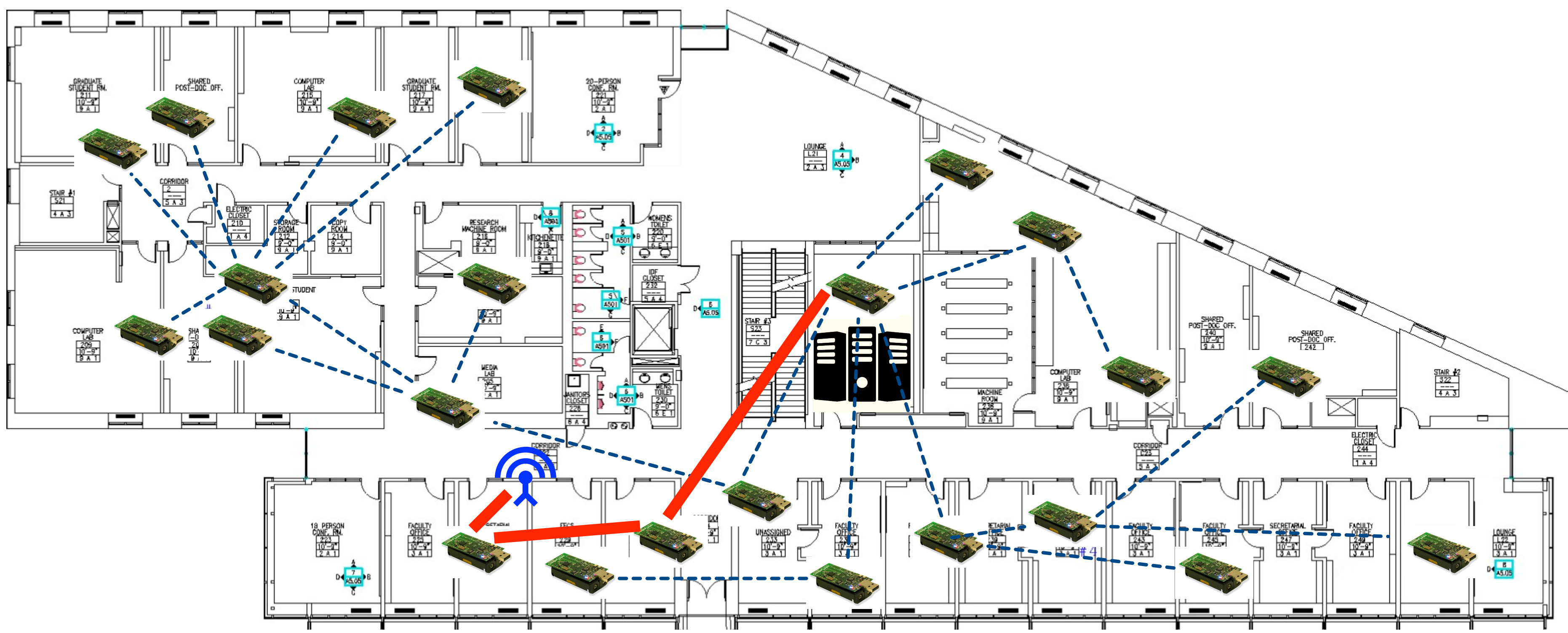
Short Communication Range

Frequent Association with Infrastructure Nodes

Frequent Changes in Data Forwarding Paths

Bandwidth Reservation Upon Node Admission

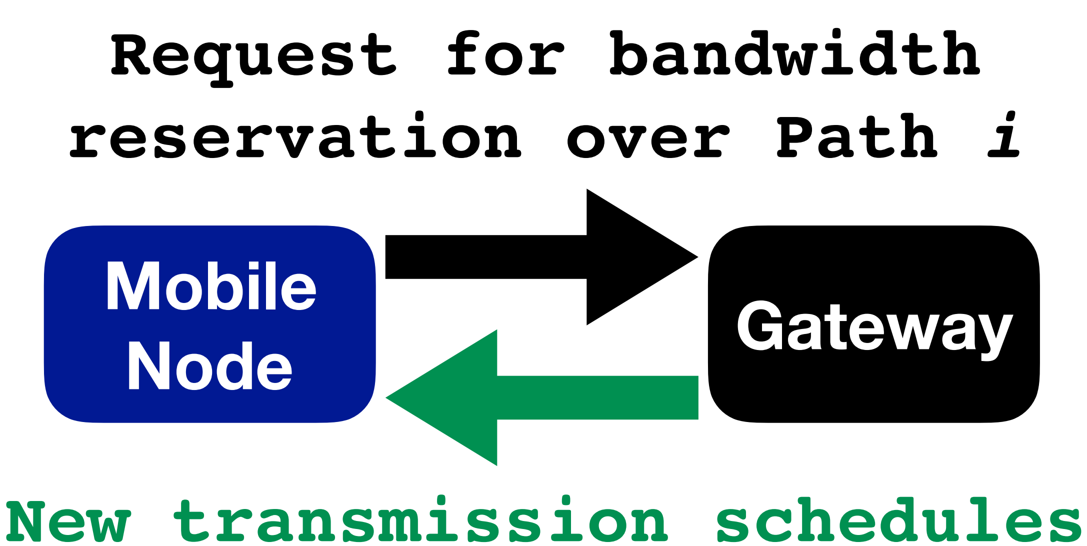


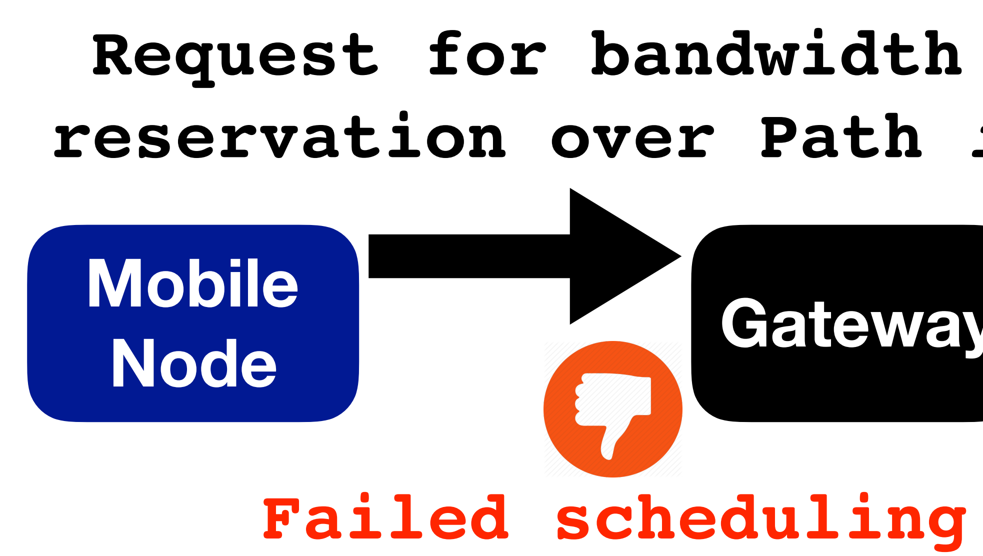


Two Bandwidth Reservation Strategies

1: On-Demand Bandwidth Reservation

- Whenever a mobile node needs to communicate over a path, it sends a request to the Gateway
  - Shortcoming #1: Huge bandwidth should be reserved for exchanging control data
- Gateway performs bandwidth reservation over the new communication path after receiving a request
  - Shortcoming #2: The Gateway may not be able to reserve bandwidth over the new communication path: CONNECTION LOSS!

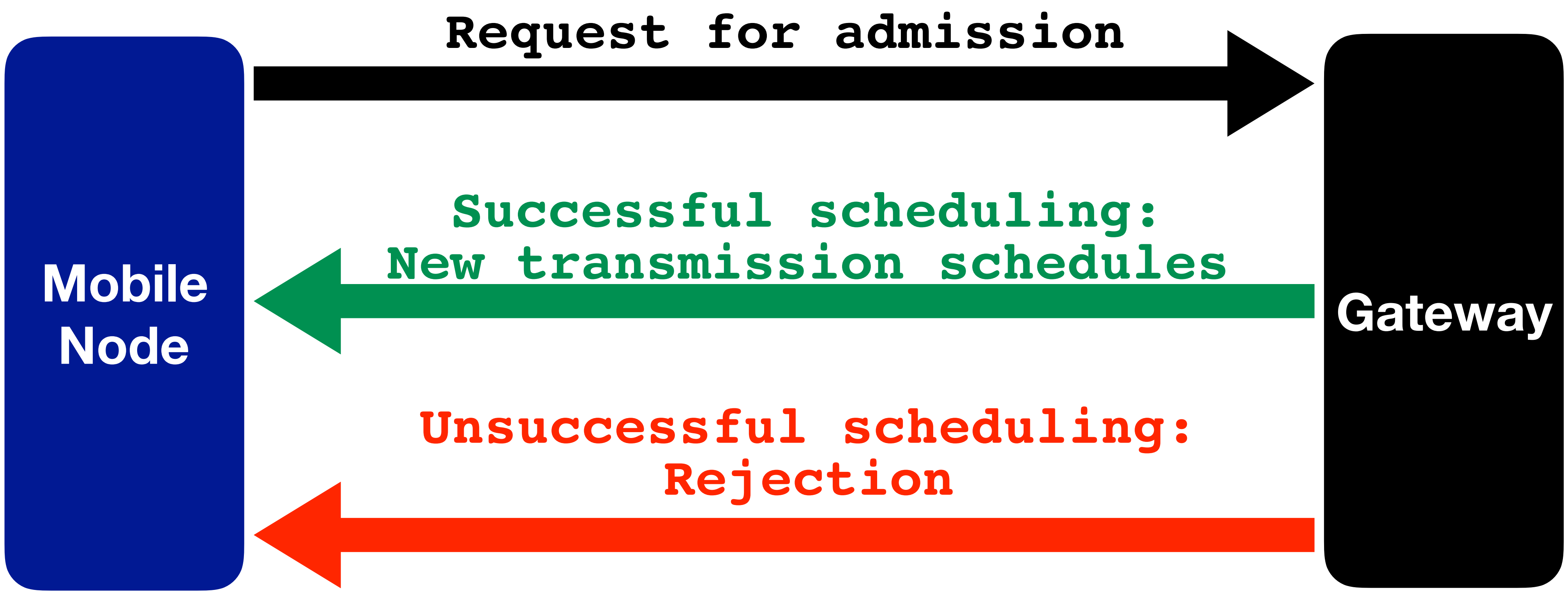




Two Bandwidth Reservation Strategies

2: On-Join Bandwidth Reservation (our choice)

- Bandwidth is reserved *over all the potential communication paths upon node join*
- Gateway admits a mobile node if bandwidth reservation over all the potential communications paths was successful
  - Shortcoming: If performed naively, the number of admitted mobile nodes would be very small
  - We propose techniques to address this shortcoming



Mobile Node Admission

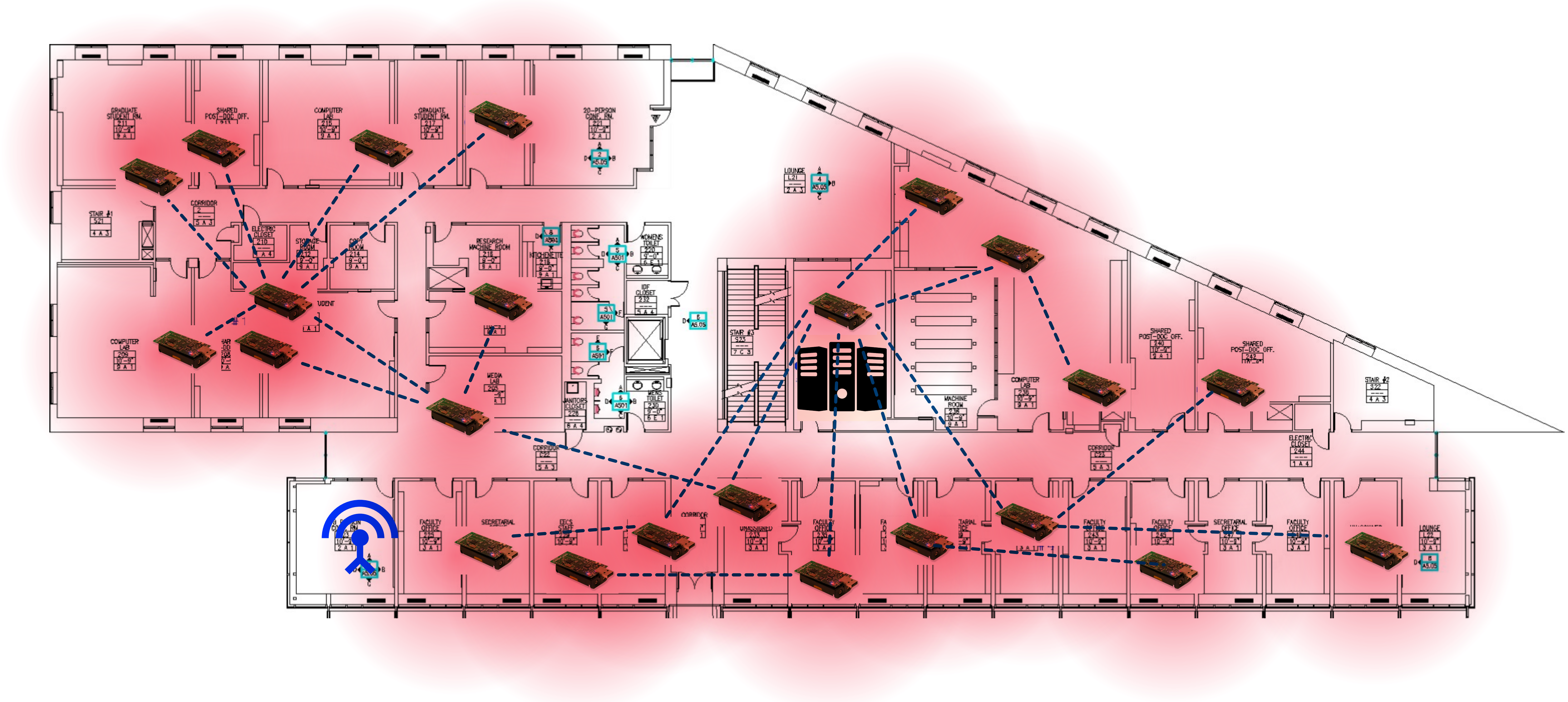
Admitting a mobile node:

1. **Beaconing:**
   - Infrastructure nodes periodically broadcast beacon packets
   - Mobile node can discover nearby infrastructure nodes
2. **Request for Join:**
   - Mobile node sends a request for join
   - Infrastructure nodes forward the request towards the Gateway
3. **Schedule Computation and Dissemination**
   - The Gateway computes a new schedule to accommodate for the new node
   - Infrastructure nodes distribute the computed schedule
   - The mobile node receives the schedule

Admission: First Step

1. Beaconing

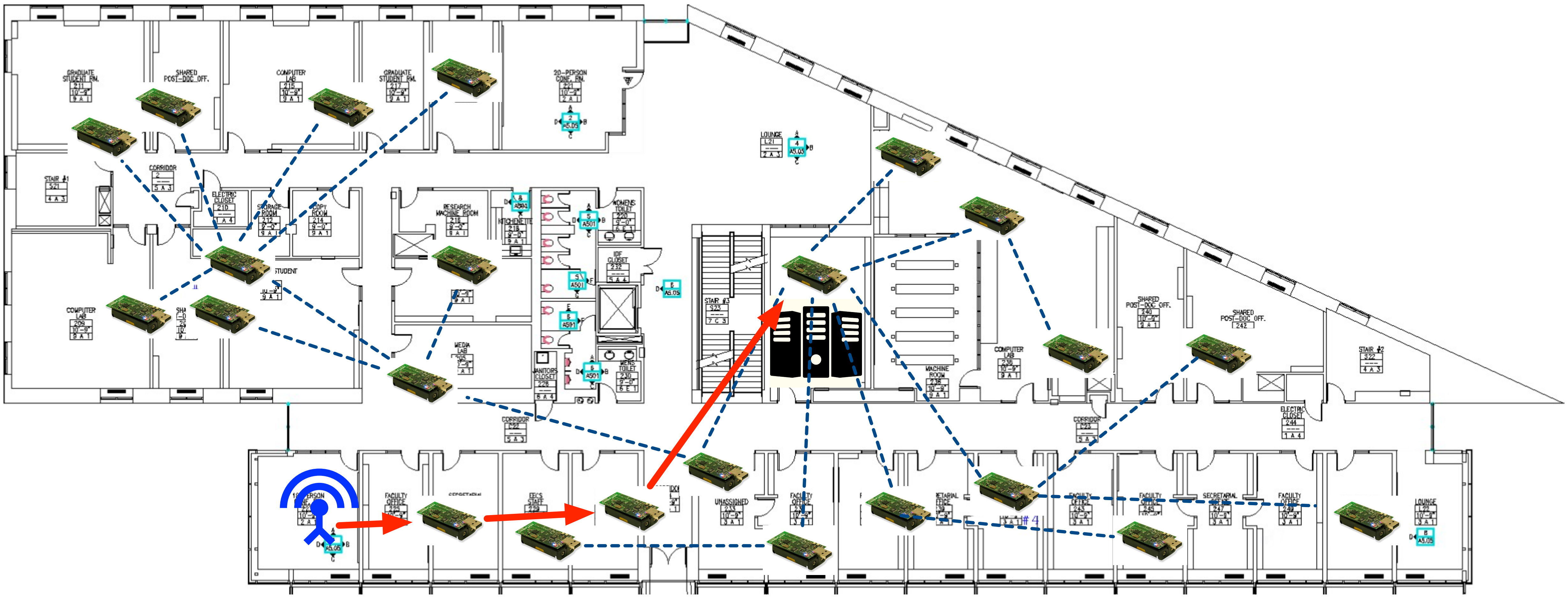
- Infrastructure nodes regularly broadcast beacon packets
- Mobile nodes discover nearby nodes



Admission: Second Step

2. Join Request

- The mobile node sends a join request
- Infrastructure nodes forward the request to the Gateway
- Gateway decides about the admission of the mobile node



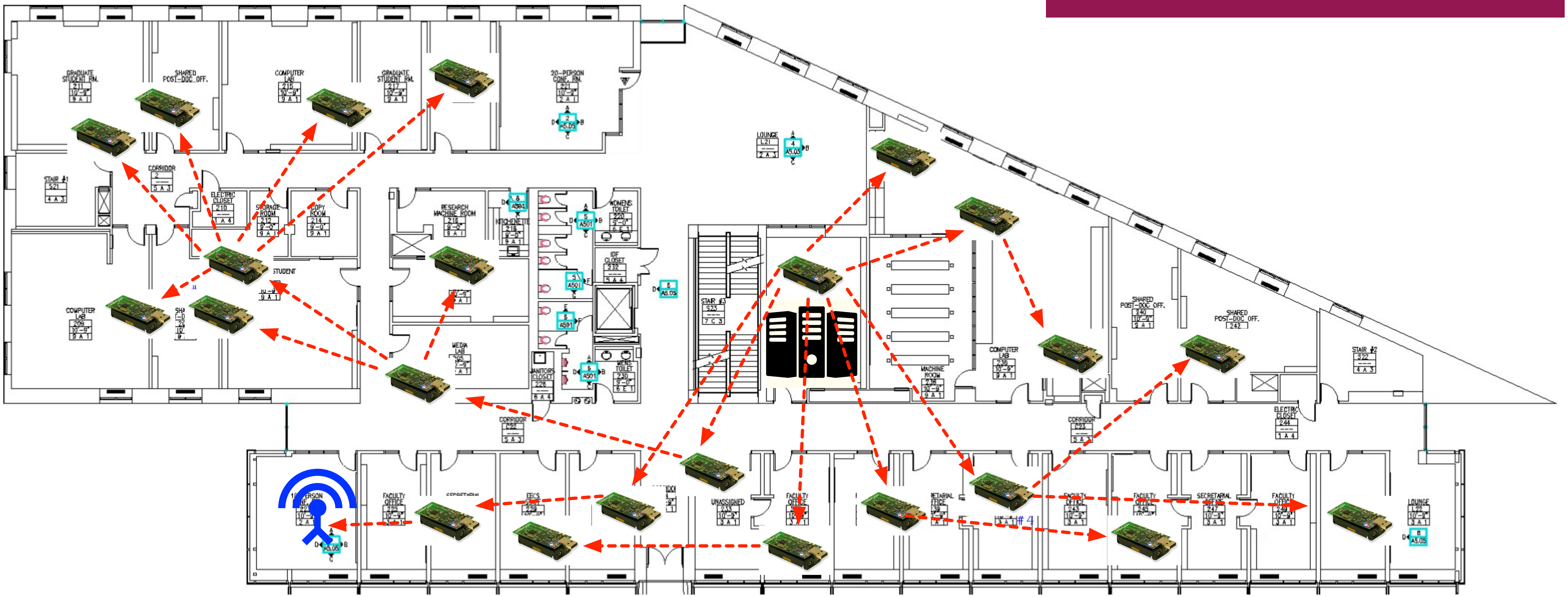
Gateway implements a scheduling algorithm that reserves bandwidth for the new mobile node

Admission: Third Step

3. Schedule Computation and Dissemination

```
If:
 The mobile node can be admitted
Then:
 Distribute transmission schedules
```

How much data should be distributed?



Schedule Computation and Dissemination

How much data should be distributed when a mobile node is admitted?

Existing scheduling strategies:

Scheduling a new flow may modify the schedules of existing flows

↓

Every admission requires distributing the transmissions of:
new mobile node + existing mobile nodes

↓

A huge amount of control data should be distributed after each join

↓

Long node admission delay

Increasing #Admitted Nodes → Increasing Admission Delay

Observations

Observation 1

We should employ on-join scheduling instead of on-demand scheduling

Contribution:

We propose mechanisms that increase real-time capacity

Observation 2

We should minimize the amount of control data required for schedule dissemination

Contribution:

We propose additive scheduling

Frequency-Time Division Multiple Access

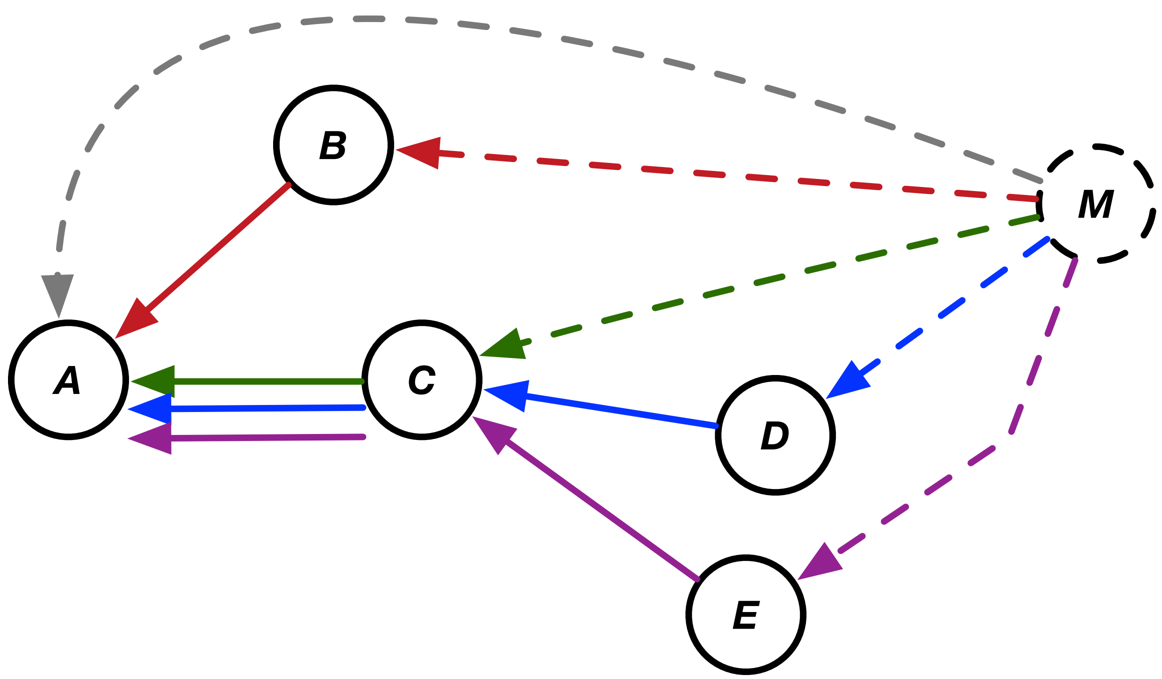
Mobility, Association, and Routing Paths

- To forward a flow *i*:
  - ***M* associates with infrastructures nodes, depending on its location**

— On-join bandwidth reservation:

- **Reserve bandwidth for *M* over all the potential communication paths**

Potential Communication Paths			
Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*	*CA*	
Path 4	*MD*	*DC*	*CA*
Path 5	*ME*	*EC*	*CA*



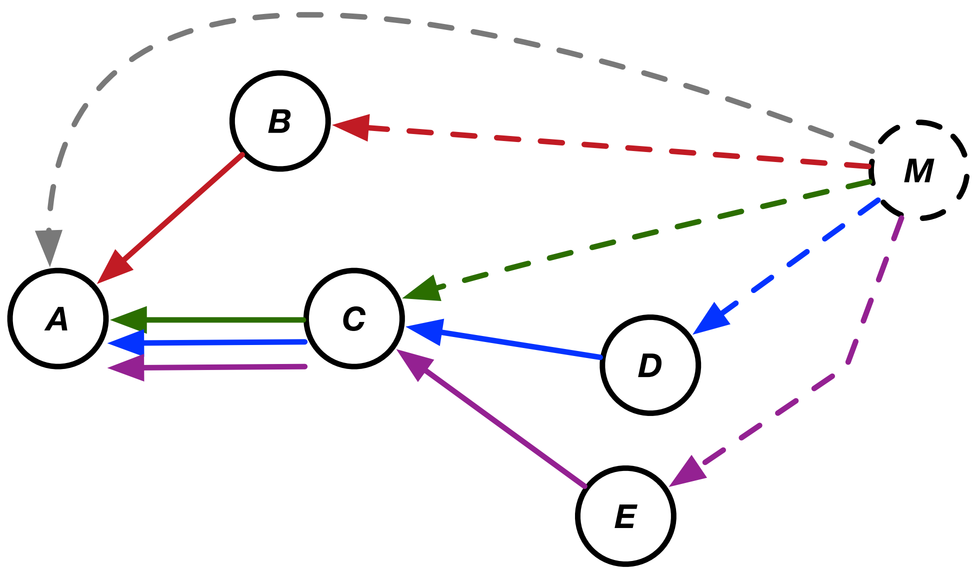
- How **a scheduling algorithm designed for *stationary* real-time networks** would perform the scheduling?
- We refer to this algorithm as **Static Real-time Scheduling (SRS)**

SRS: Static Real-time Scheduling

Scheduling Constraints:

- A node cannot send and receive simultaneously
- On a path, a transmission `BC` can be scheduled after transmission `AB` has been scheduled

Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*	*CA*	
Path 4	*MD*	*DC*	*CA*
Path 5	*ME*	*EC*	*CA*



The scheduling matrix produced by SRS

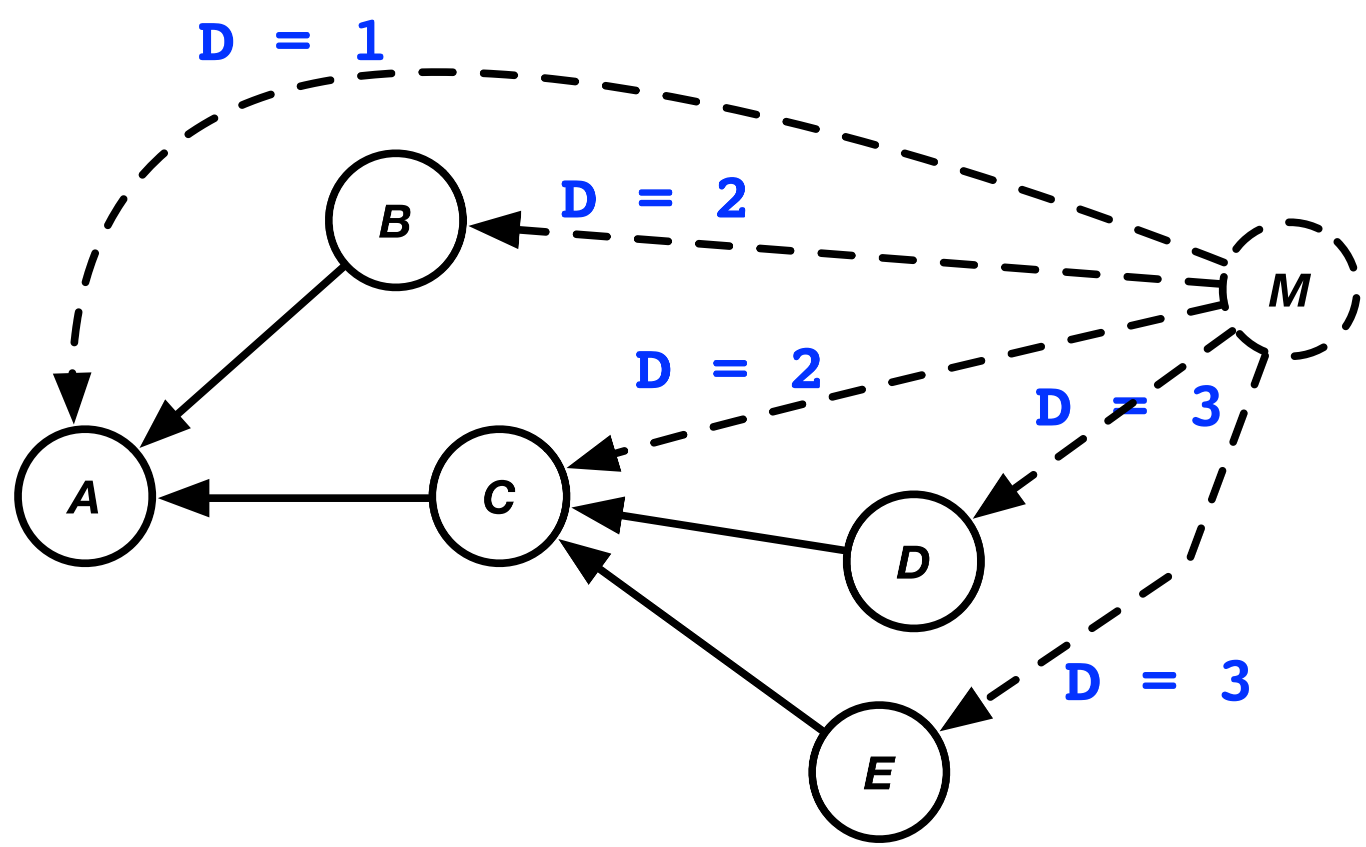
	0	1	2	3	4	5	6	7	8	9	10	…
C_1	*ME*	*MD*	*MB*	*MC*	*MA*	*CA*	*CA*	*CA*				
C_2		*EC*	*DC*	*BA*								
C_3												
…												

This schedule is inefficient !

SRS: Static Real-time Scheduling

Potential Communication Paths

Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*	*CA*	
Path 4	*MD*	*DC*	*CA*
Path 5	*ME*	*EC*	*CA*



ready transmissions =
{(*ME*), (*MD*), (*MC*), (*MB*), (*MA*)}

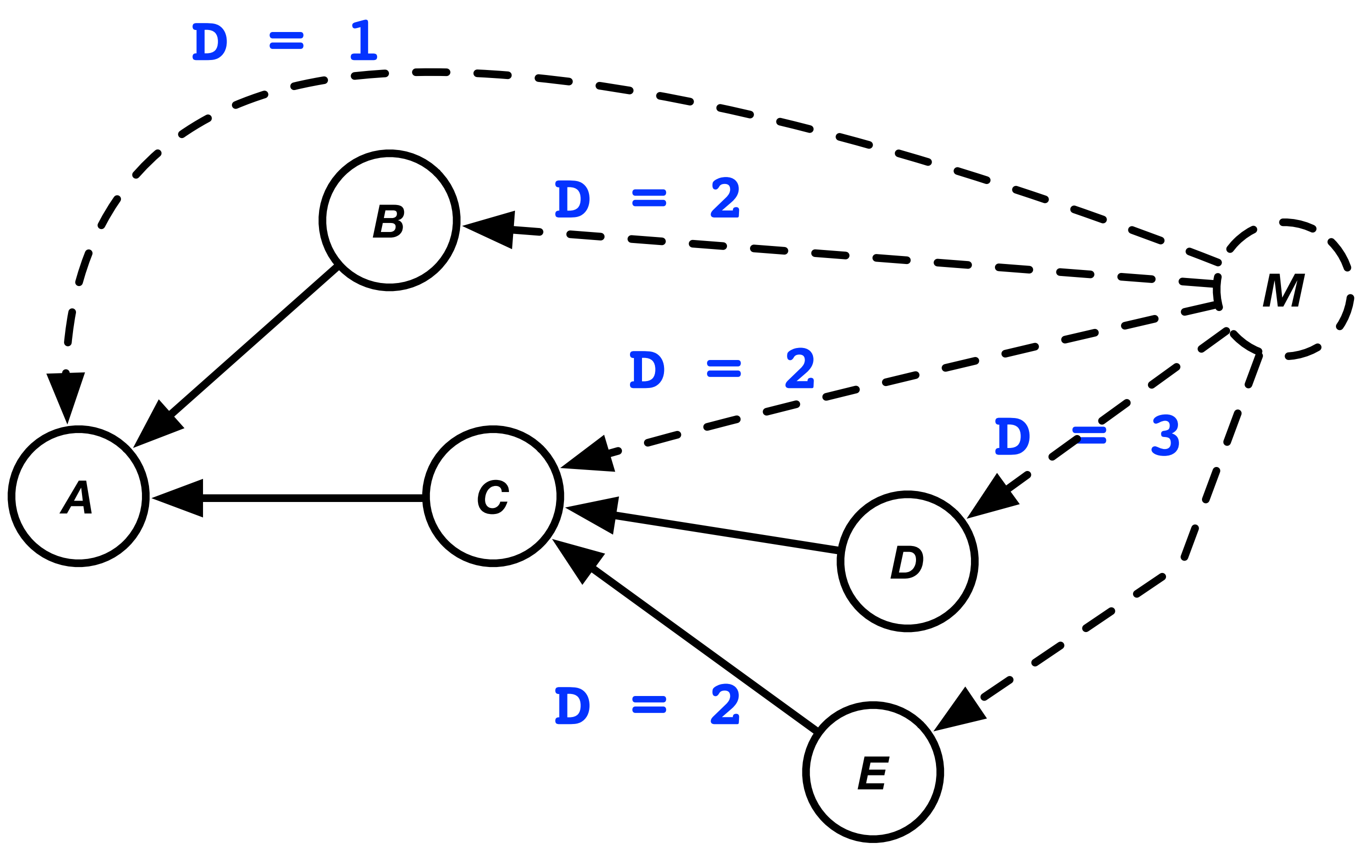
highest depth: higher priority → lowest depth: lower priority

	0	1	2	3	4	5	6	7	8	9	10	…
c_1	*ME*											
c_2												
c_3												
…												

SRS: Static Real-time Scheduling

Potential Communication Paths

Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*	*CA*	
Path 4	*MD*	*DC*	*CA*
Path 5		*EC*	*CA*

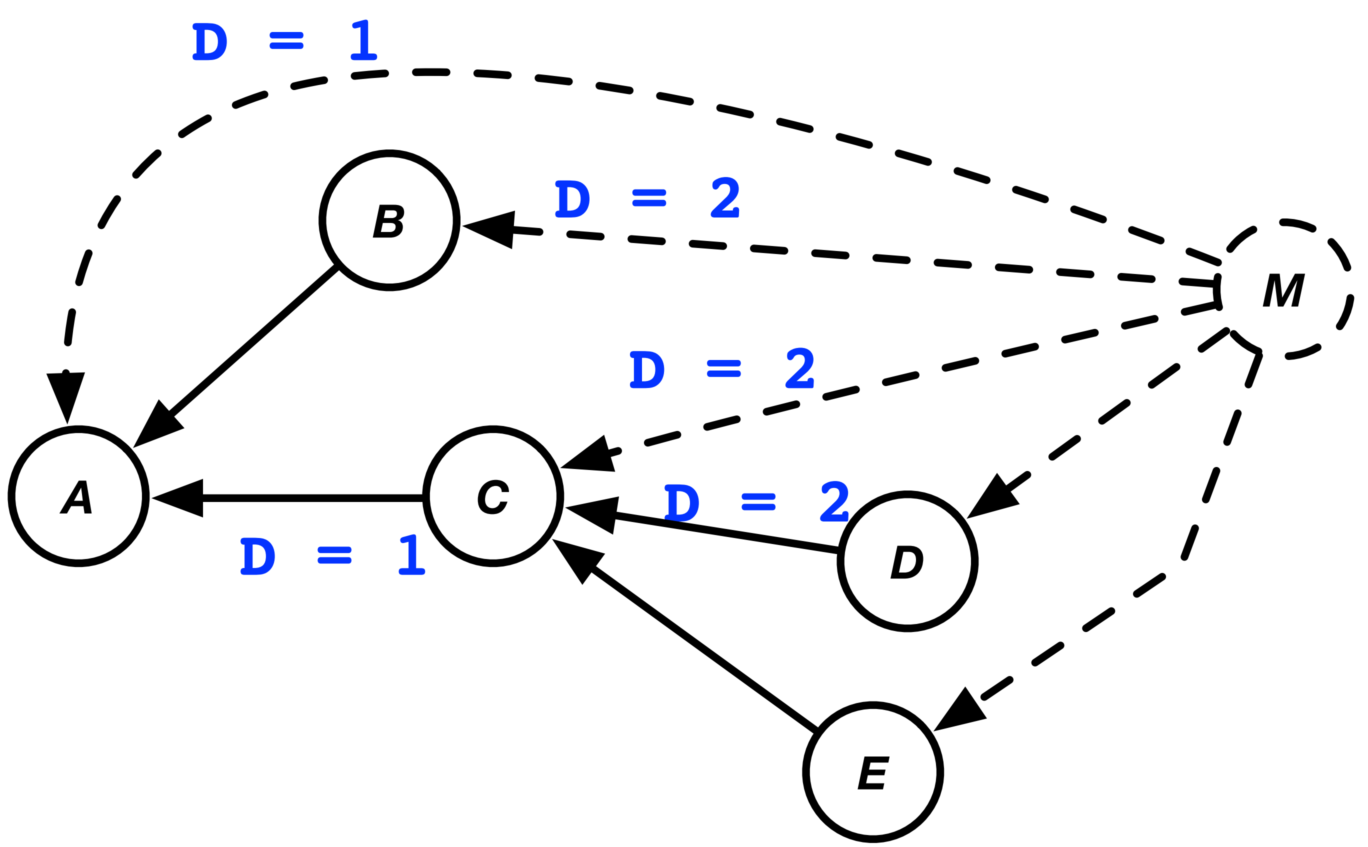


ready transmissions = {(*MD*), (*EC*), (*MC*), (*MB*), (*MA*)}

	0	1	2	3	4	5	6	7	8	9	10	...
c_1	*ME*	*MD*										
c_2		*EC*										
c_3												
...												

SRS: Static Real-time Scheduling

Potential Communication Paths			
Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*	*CA*	
Path 4		*DC*	*CA*
Path 5			*CA*

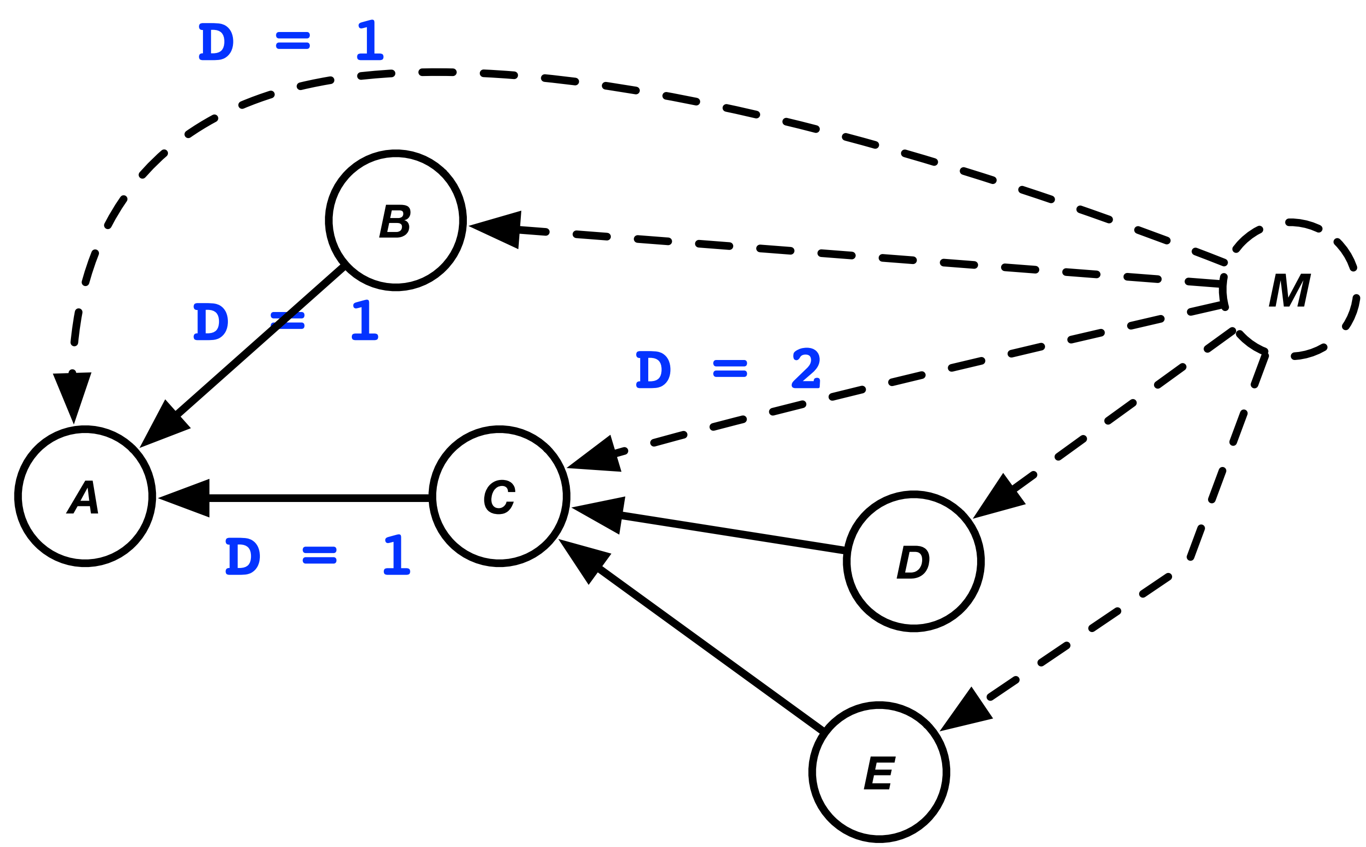


ready transmissions =
{(*MB*), (*MC*), (*DC*), (*MA*), (*CA*)}

	0	1	2	3	4	5	6	7	8	9	10	…
c_1	*ME*	*MD*	*MB*									
c_2		*EC*	*DC*									
c_3												
…												

SRS: Static Real-time Scheduling

Potential Communication Paths			
Path 1	*MA*		
Path 2		*BA*	
Path 3	*MC*	*CA*	
Path 4			*CA*
Path 5			*CA*

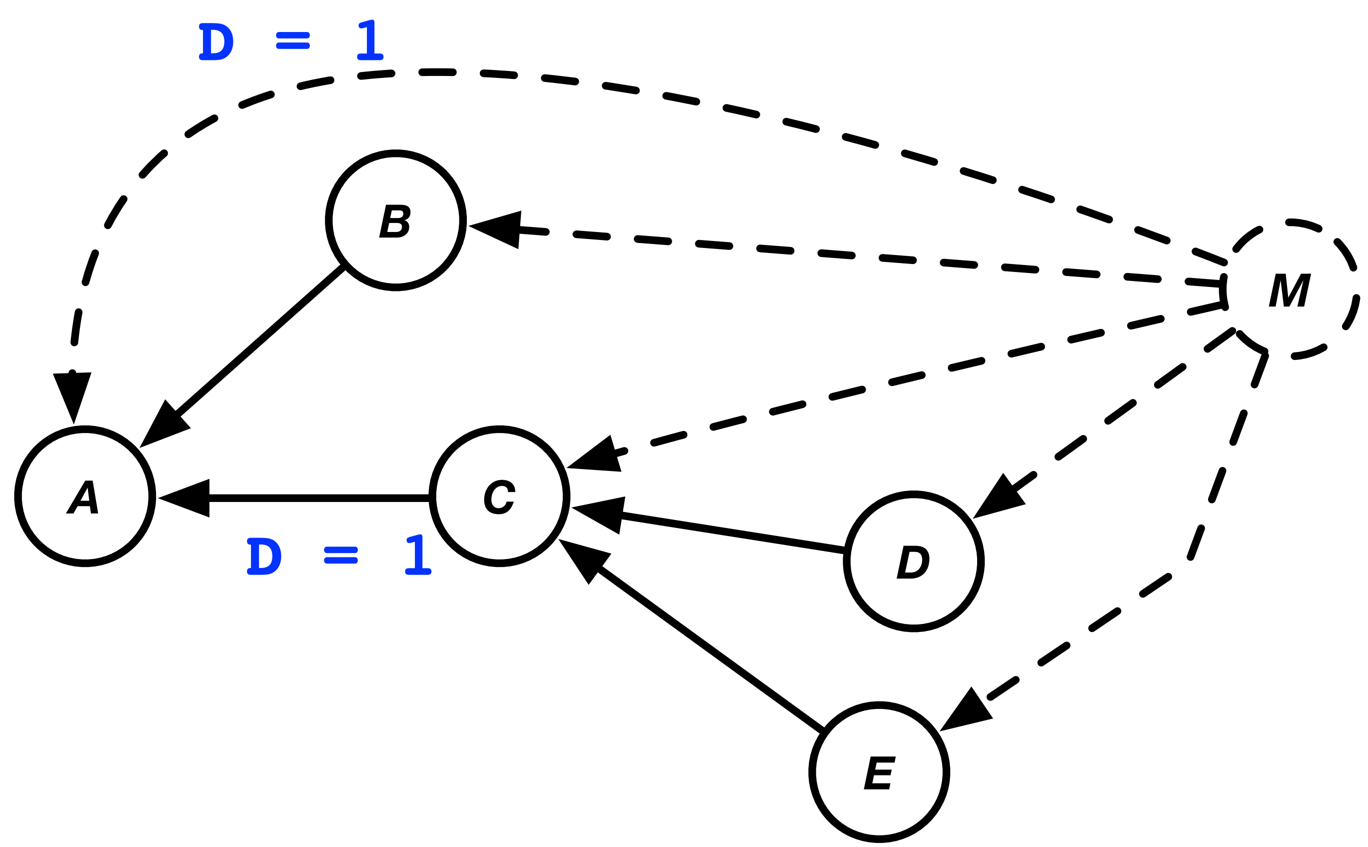


ready transmissions =
{(MC), (BA), (MA), (CA), (CA)}

	0	1	2	3	4	5	6	7	8	9	10	...
c_1	*ME*	*MD*	*MB*	*MC*								
c_2		*EC*	*DC*	*BA*								
c_3												
...												

SRS: Static Real-time Scheduling

Potential Communication Paths			
Path 1	*MA*		
Path 2			
Path 3		*CA*	
Path 4			*CA*
Path 5			*CA*

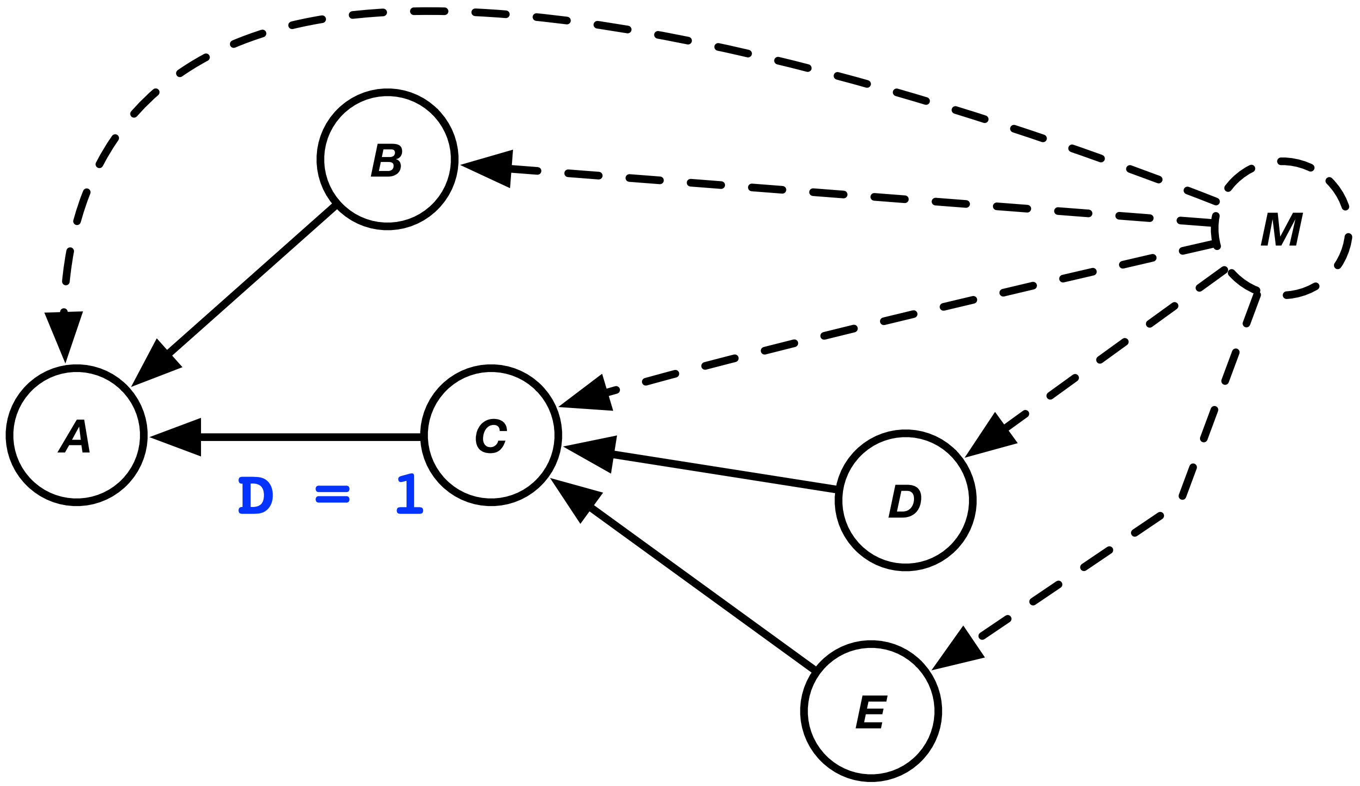


ready transmissions =
{*(MA)*, *(CA)*, *(CA)*, *(CA)*}

	0	1	2	3	4	5	6	7	8	9	10	...
c_1	*ME*	*MD*	*MB*	*MC*	*MA*							
c_2		*EC*	*DC*	*BA*								
c_3												
...												

SRS: Static Real-time Scheduling

Potential Communication Paths			
Path 1			
Path 2			
Path 3		CA	
Path 4			CA
Path 5			CA

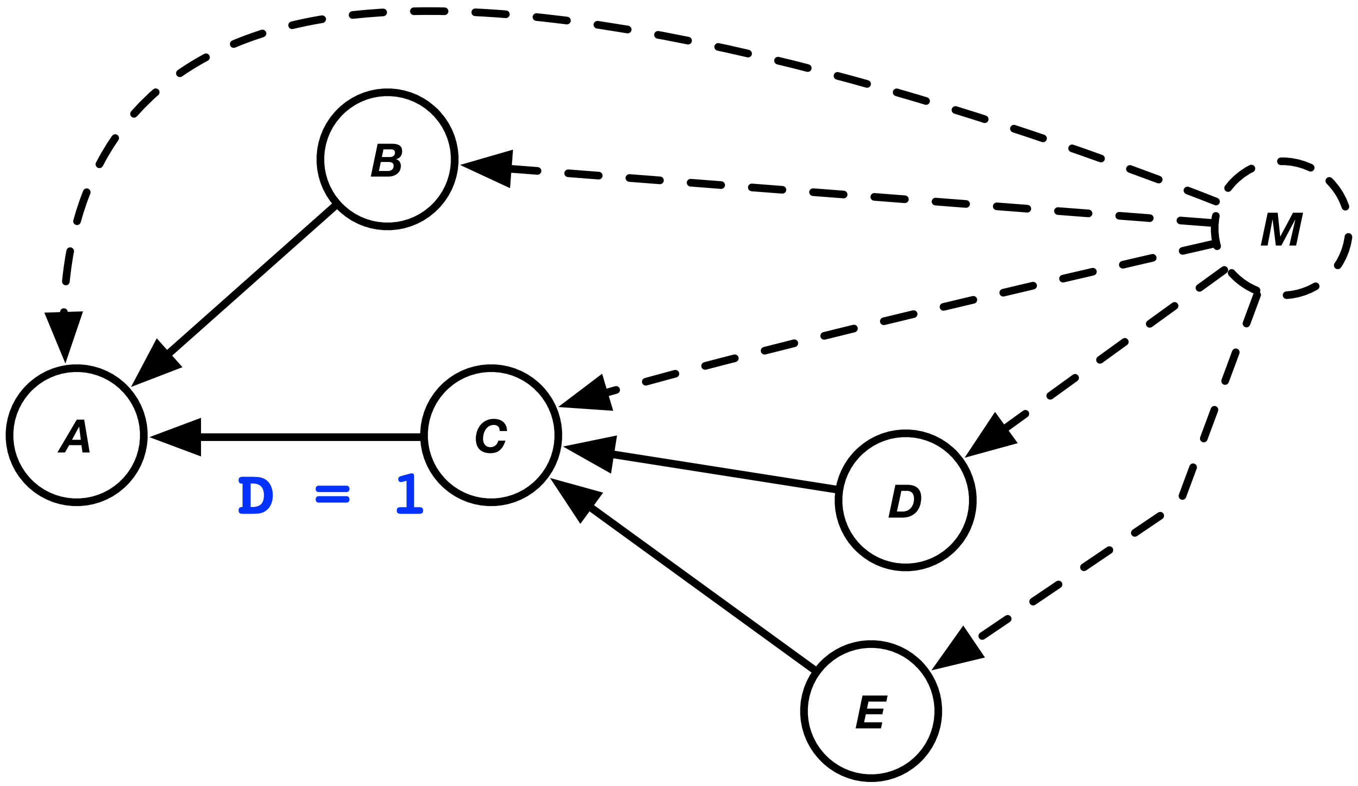


ready transmissions = {(CA), (CA), (CA)}

	0	1	2	3	4	5	6	7	8	9	10	...
c_1	ME	MD	MB	MC	MA	CA						
c_2		EC	DC	BA								
c_3												
...												

SRS: Static Real-time Scheduling

Potential Communication Paths			
Path 1			
Path 2			
Path 3		CA	
Path 4			CA
Path 5			

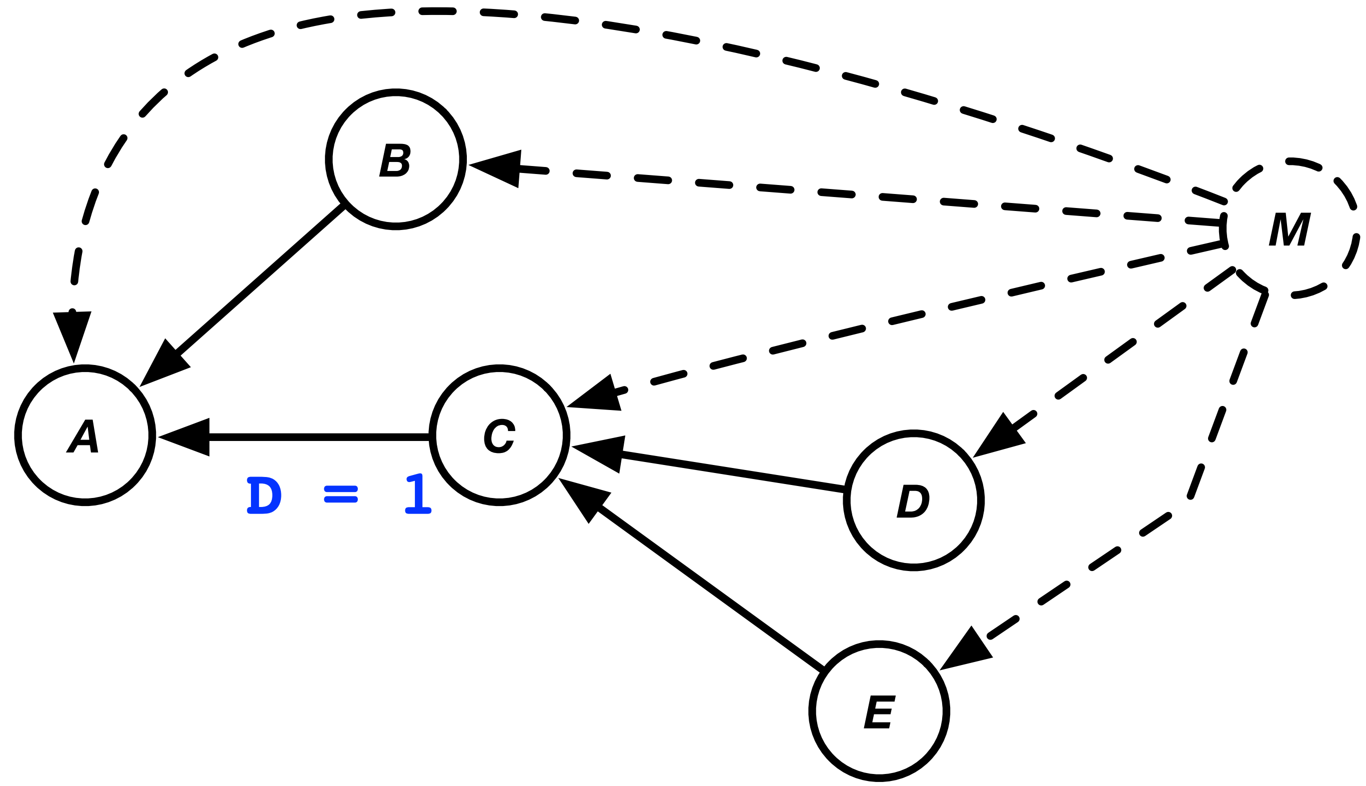


ready transmissions = {(CA), (CA)}

	0	1	2	3	4	5	6	7	8	9	10	...
c_1	ME	MD	MB	MC	MA	CA	CA					
c_2		EC	DC	BA								
c_3												
...												

SRS: Static Real-time Scheduling

Potential Communication Paths			
Path 1			
Path 2			
Path 3		CA	
Path 4			
Path 5			

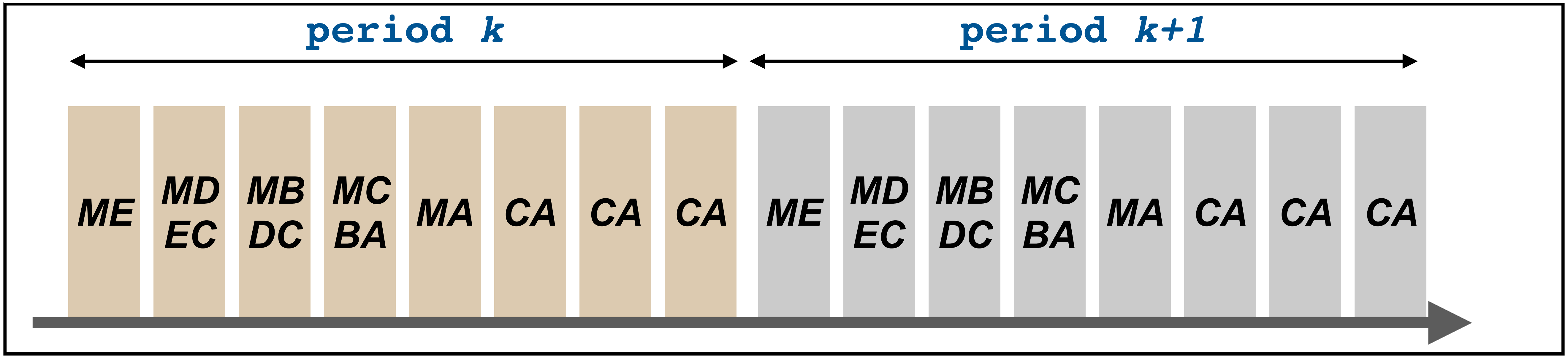


ready transmissions = {(CA)}

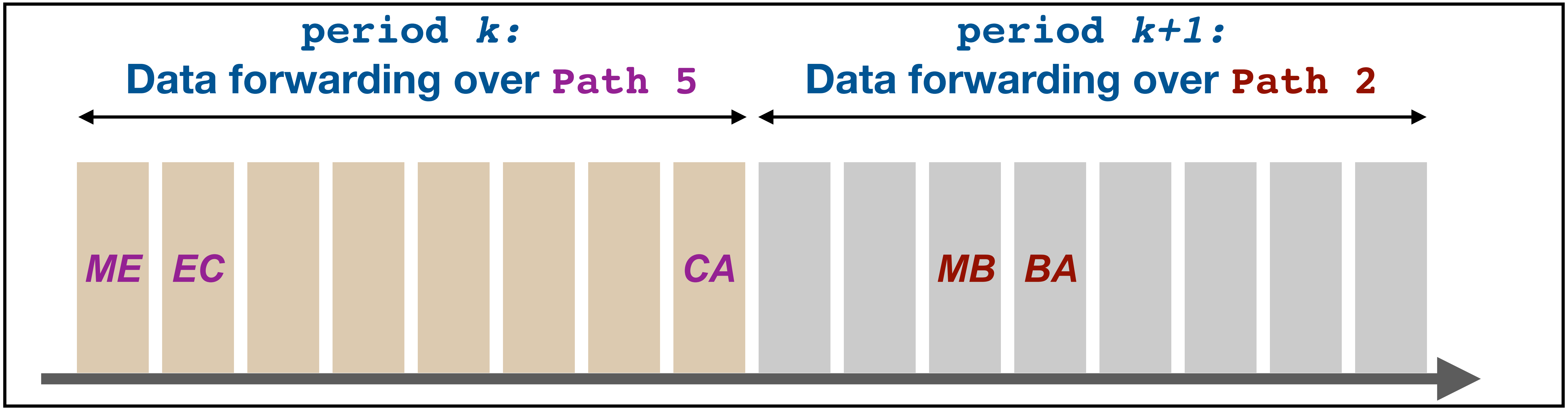
	0	1	2	3	4	5	6	7	8	9	10	...
c_1	ME	MD	MB	MC	MA	CA	CA	CA				
c_2		EC	DC	BA								
c_3												
...												

Path-Dependent Schedule Activation

Actual Transmission Schedules Assignment

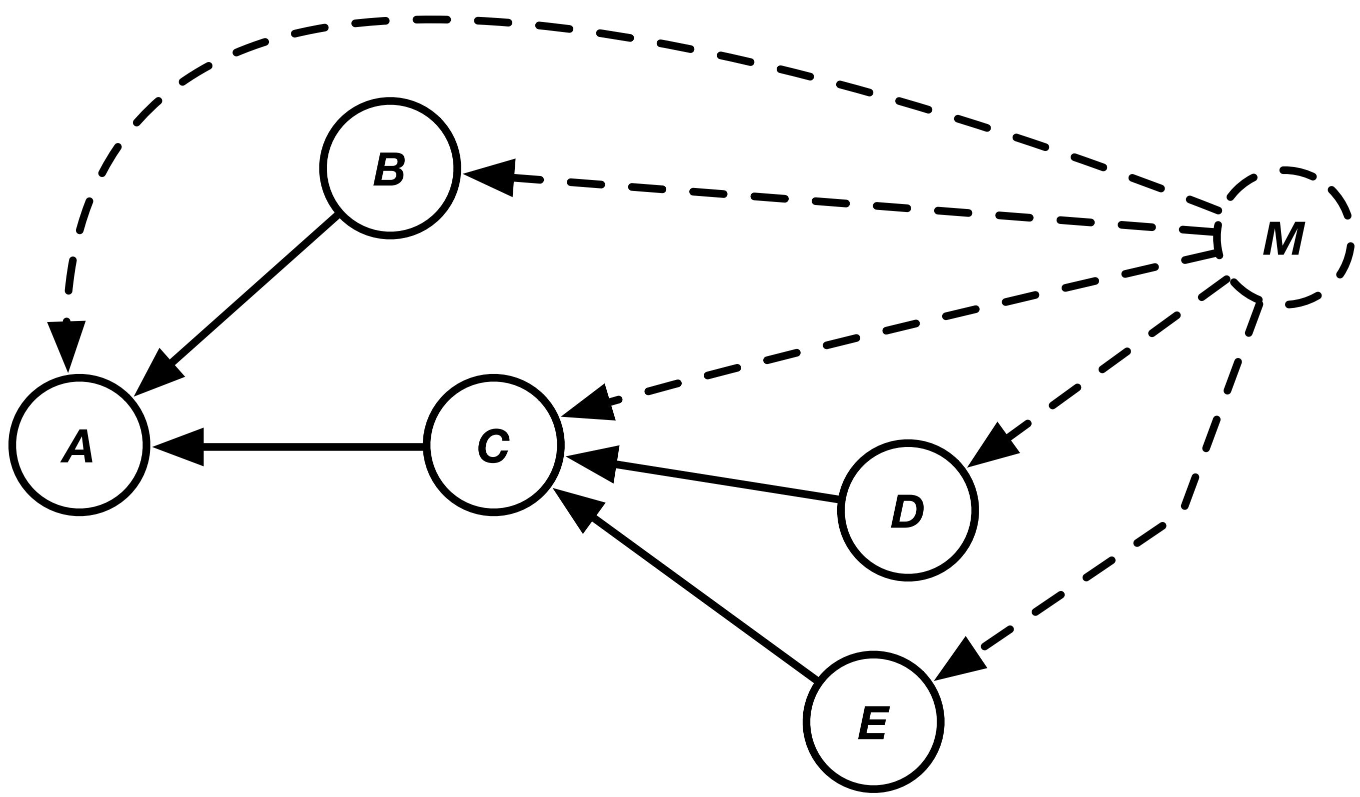


Path-Dependent Schedule Activation



Technique 1: Schedule Combination

For a flow `i`, two transmissions belonging to two different paths can be combined in one entry of the scheduling matrix.

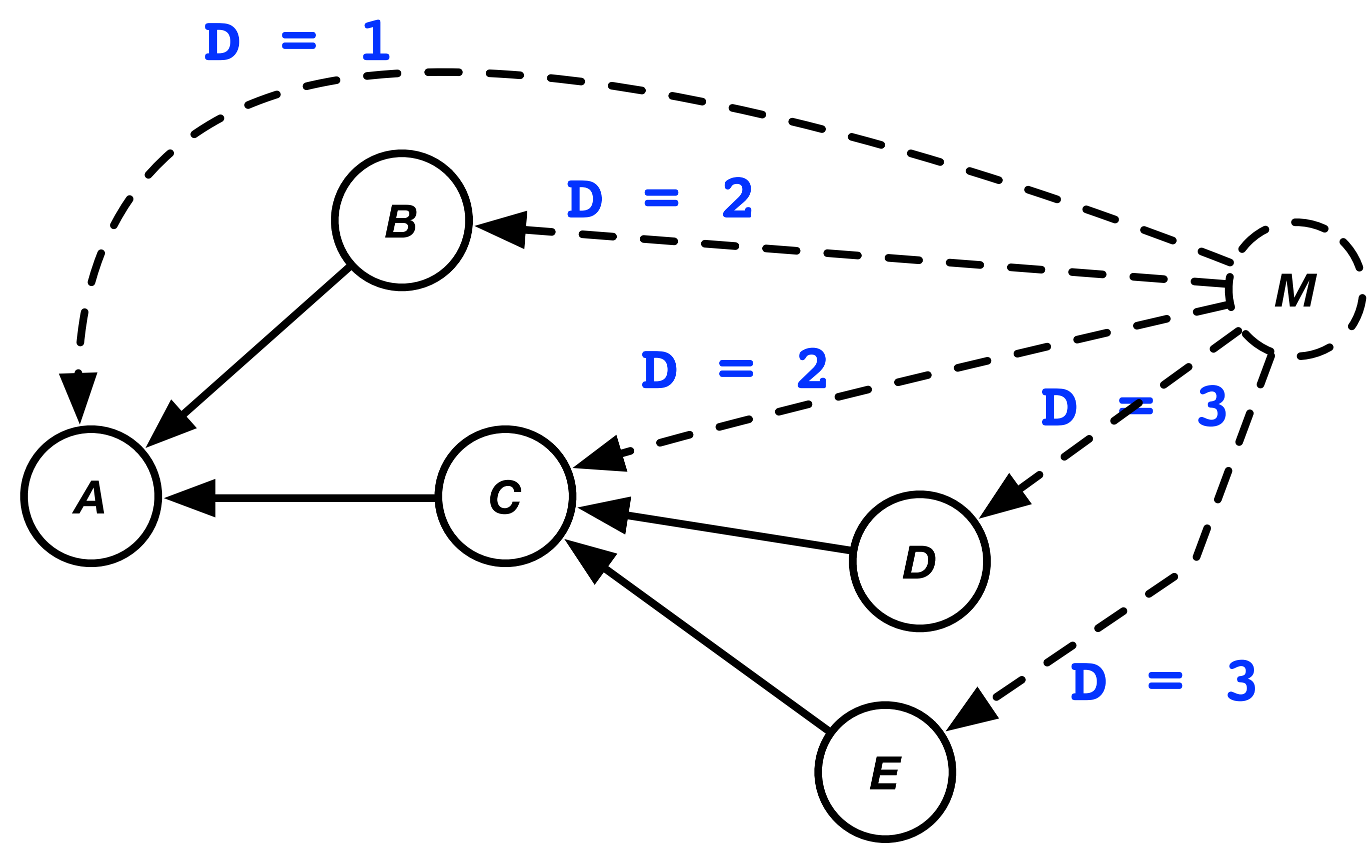


For Example:

- Transmissions (*MA*), (*MB*), (*MC*), (*MD*) and (*ME*) can be combined.
- Transmissions (*MC*), (*DC*) and (*EC*) can be combined.
- …

Technique 1: Schedule Combination

Potential Communication Paths			
Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*	*CA*	
Path 4	*MD*	*DC*	*CA*
Path 5	*ME*	*EC*	*CA*

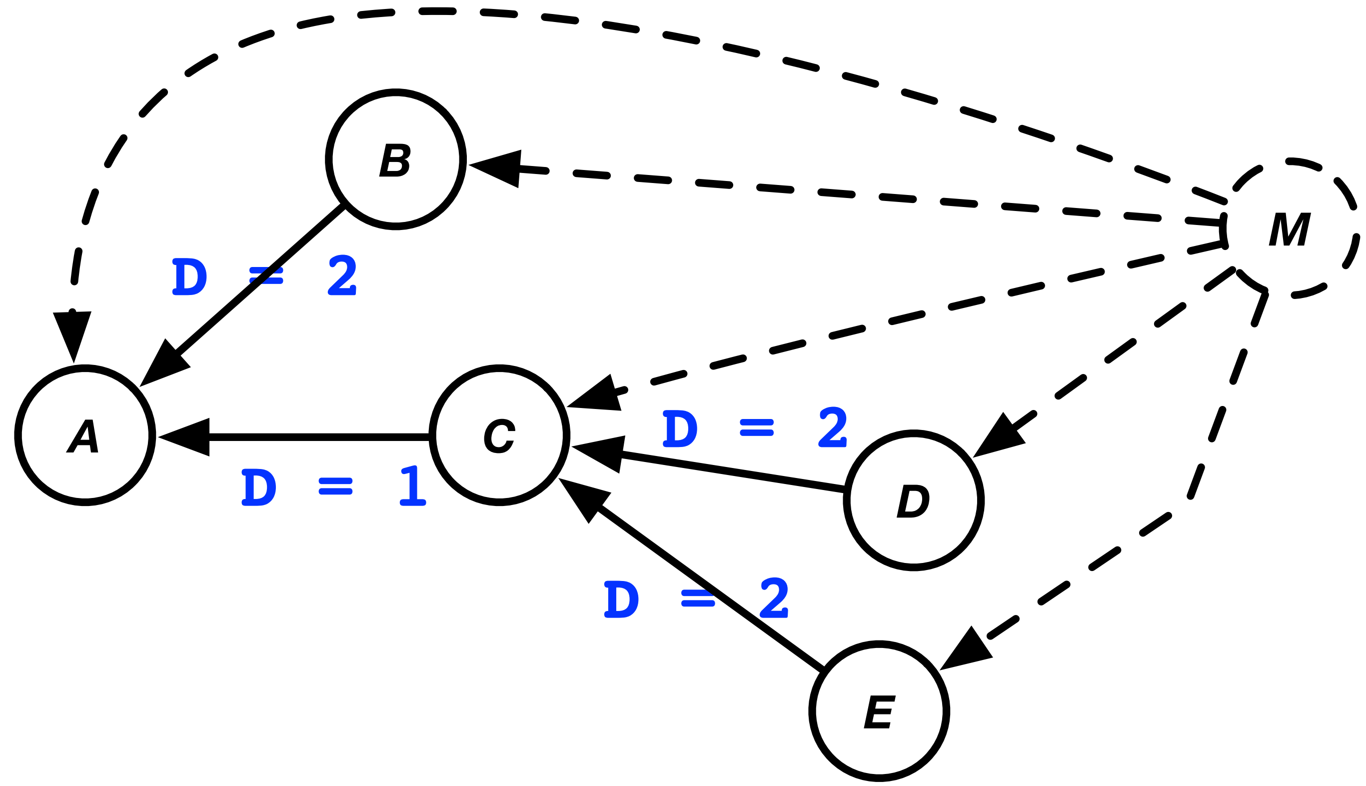


ready transmissions =
{(*ME*), (*MD*), (*MC*), (*MB*), (*MA*)}

	0	1	2	3	4	5	6	7	8	9	10	...
C_1	*ME* *MD* *MC* *MB* *MA*											

Technique 1: Schedule Combination

Potential Communication Paths			
Path 1			
Path 2		BA	
Path 3		CA	
Path 4		DC	CA
Path 5		EC	CA

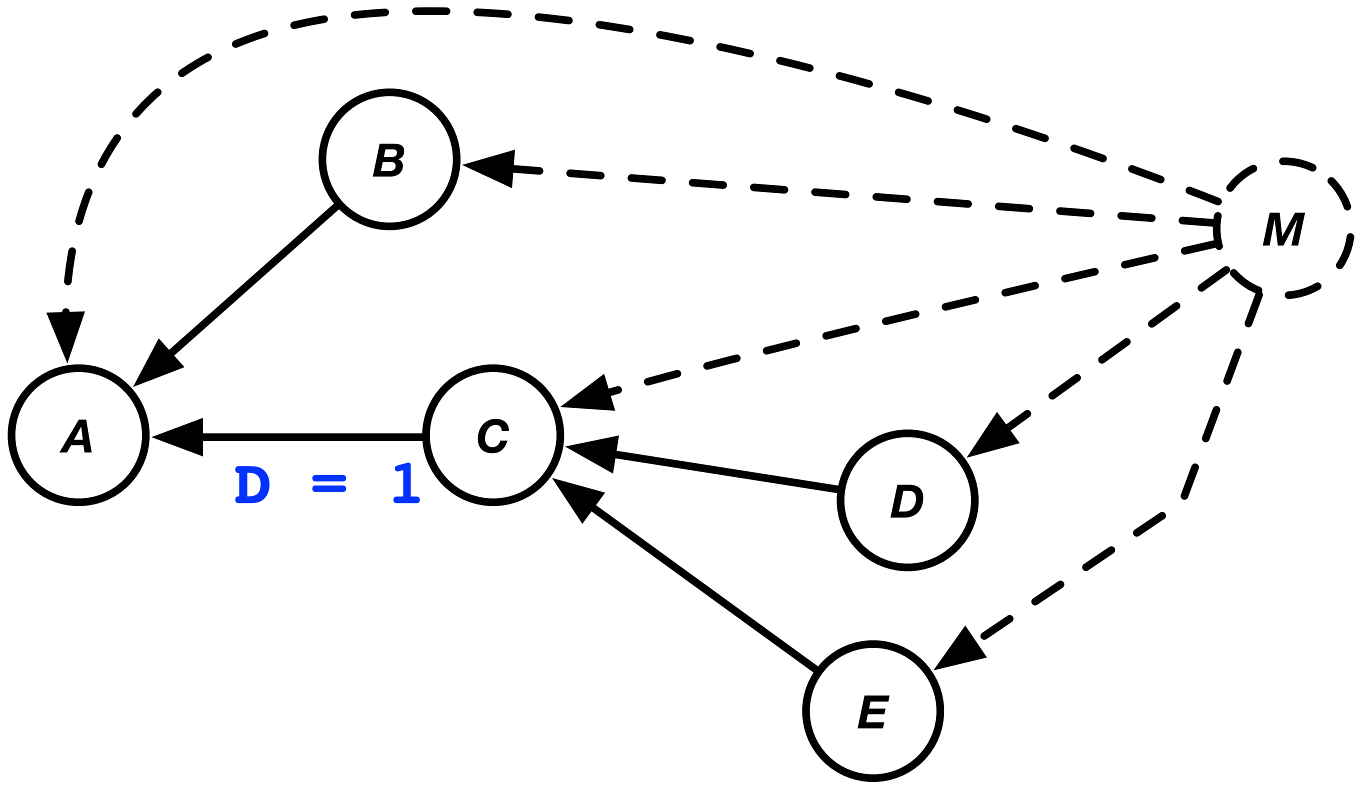


ready transmissions =
{(BA), (CA), (DC), (EC)}

	0	1	2	3	4	5	6	7	8	9	10	...
c_1	ME MD MC MB MA	BA CA DC EC										

Technique 1: Schedule Combination

Potential Communication Paths			
Path 1			
Path 2			
Path 3			
Path 4			CA
Path 5			CA



ready transmissions = {(CA), (CA)}

	0	1	2	3	4	5	6	7	8	9	10	...
c_1	ME MD MC MB MA	BA CA DC EC	CA CA									

Technique 1: Schedule Combination

The scheduling matrix produced by employing: Schedule Combination

	0	1	2	3	4	5	6	7	8	9	10	…
c_1	*ME MD MC MB MA*	*BA CA DC EC*	*CA CA*									

3 entries of the scheduling matrix are used

We have scheduled the five paths more efficiently
However:
Data flows over these five paths can be coordinated

We propose the *Flow Coordination* technique to:
Reduce the number of transmissions scheduled in each entry

Technique 2: Flow Coordination

Replicated transmission schedules can be eliminated through coordinating the scheduling of potential communication paths

Without Flow Coordination

Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*	*CA*	
Path 4	*MD*	*DC*	*CA*
Path 5	*ME*	*EC*	*CA*

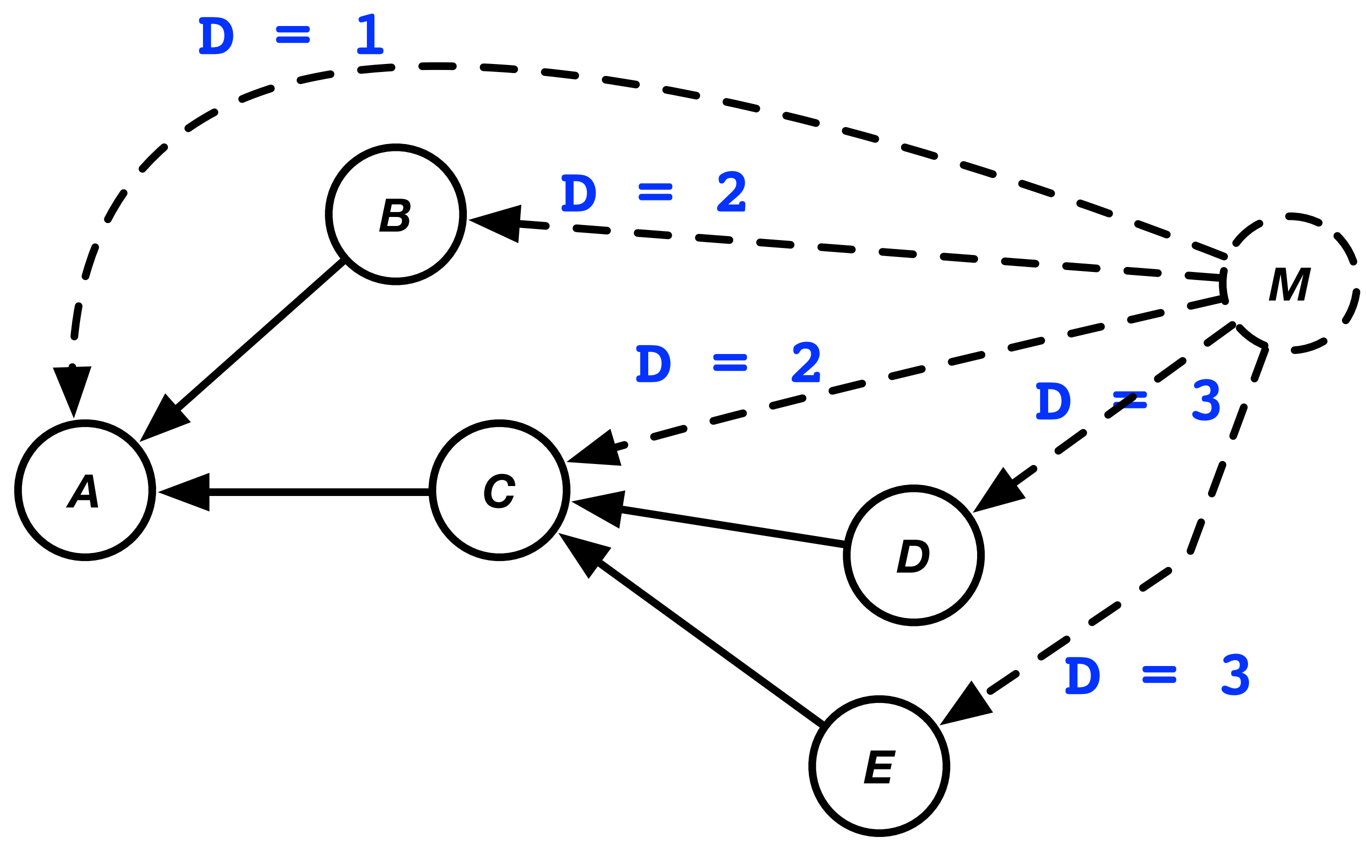
With Flow Coordination

Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*		
Path 4	*MD*	*DC*	
Path 5	*ME*	*EC*	*CA*

Transmission *CA* is scheduled once after transmissions *MC*, *DC*, and *EC* have been scheduled

Technique 2: Flow Coordination

Potential Communication Paths			
Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*		
Path 4	*MD*	*DC*	
Path 5	*ME*	*EC*	*CA*

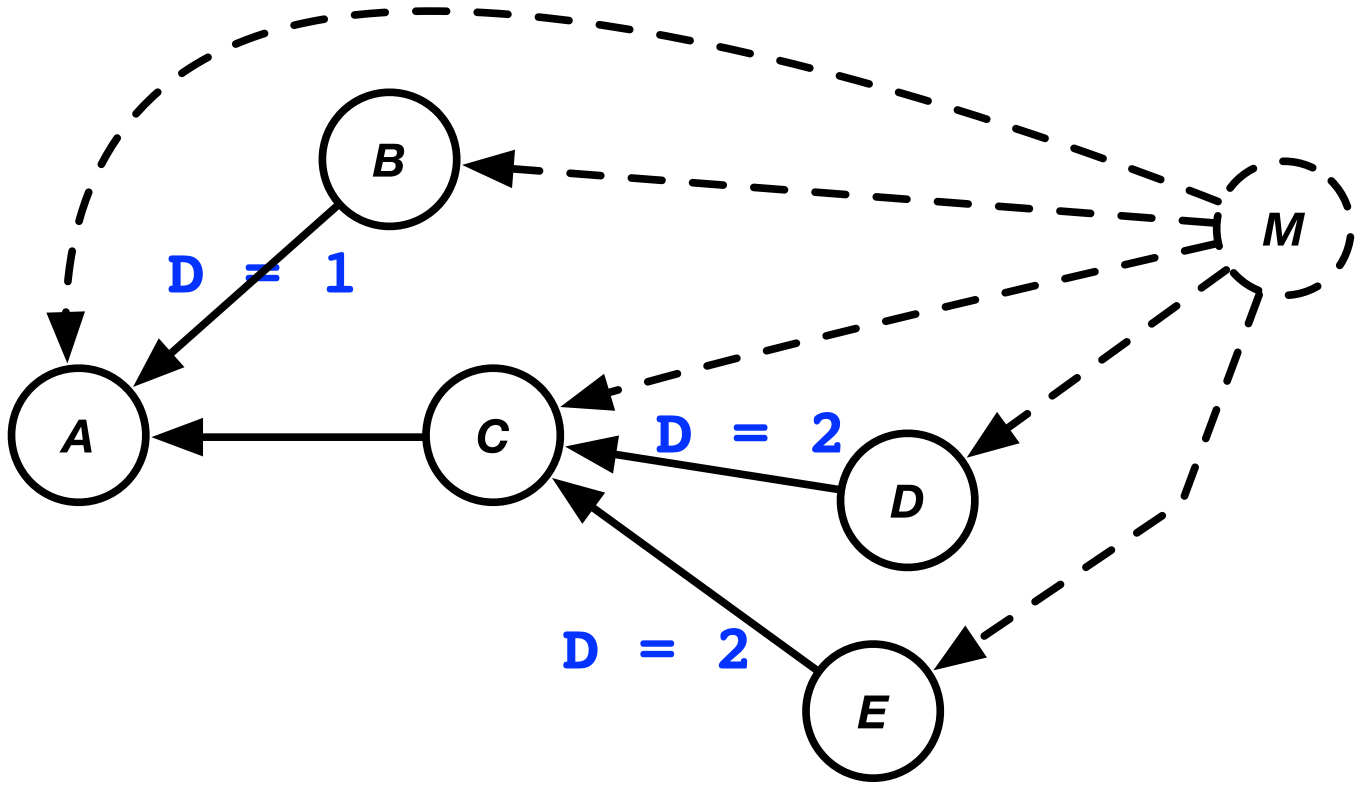


ready transmissions =
{(*ME*), (*MD*), (*MC*), (*MB*), (*MA*)}

	0	1	2	3	4	5	6	7	8	9	10	...
c_1	*ME* *MD* *MC* *MB* *MA*											

Technique 2: Flow Coordination

Potential Communication Paths			
Path 1			
Path 2		BA	
Path 3			
Path 4		DC	
Path 5		EC	CA



ready transmissions = {(BA), (DC), (EC)}

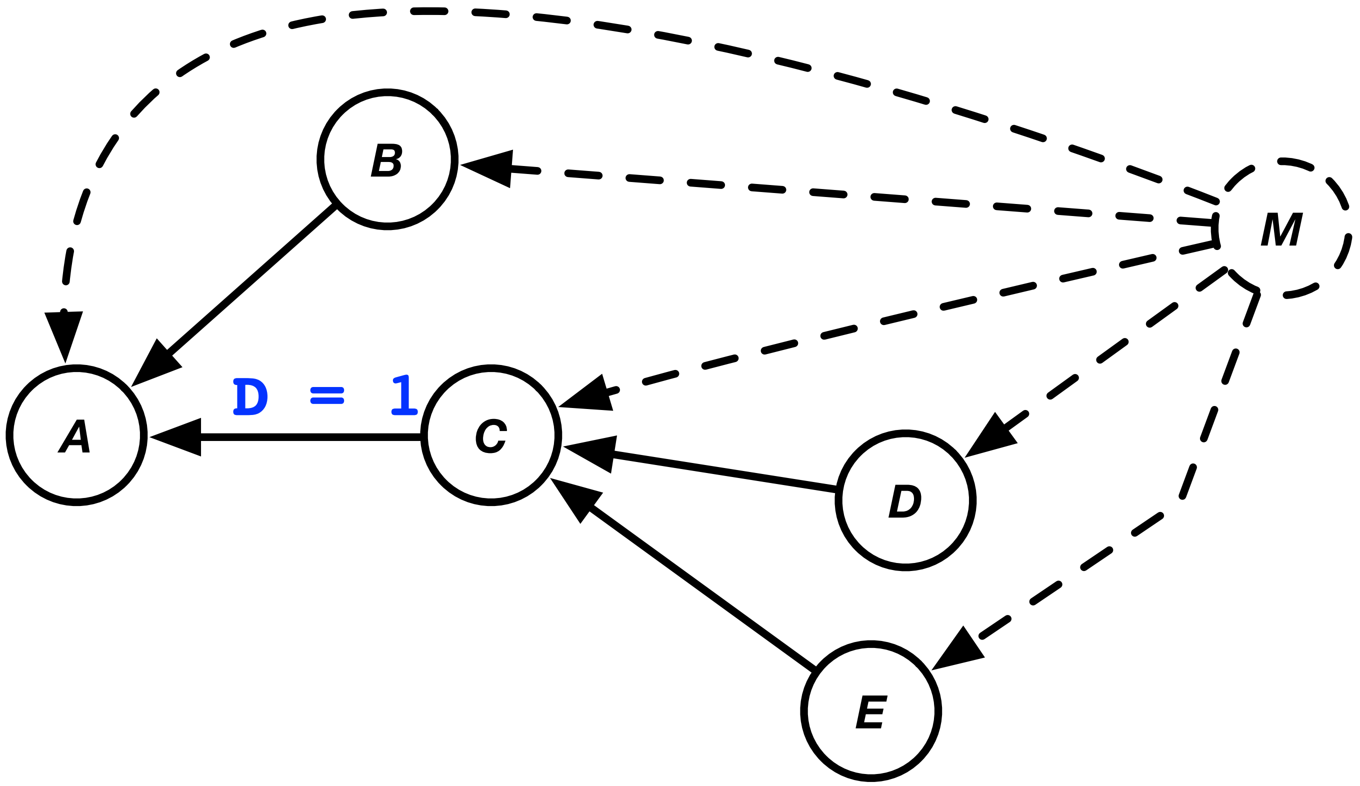
	0	1	2	3	4	5	6	7	8	9	10	...
c_1	ME MD MC MB MA	BA DC EC										

Technique 2: Flow Coordination

Potential Communication Paths

Path 1			
Path 2			
Path 3			
Path 4			
Path 5			CA

ready transmissions = {(CA)}



	0	1	2	3	4	5	6	7	8	9	10	...
C_1	ME MD MC MB MA	BA DC EC	CA									

Technique 3: Reverse Scheduling

- So far we have employed **"forward scheduling"**

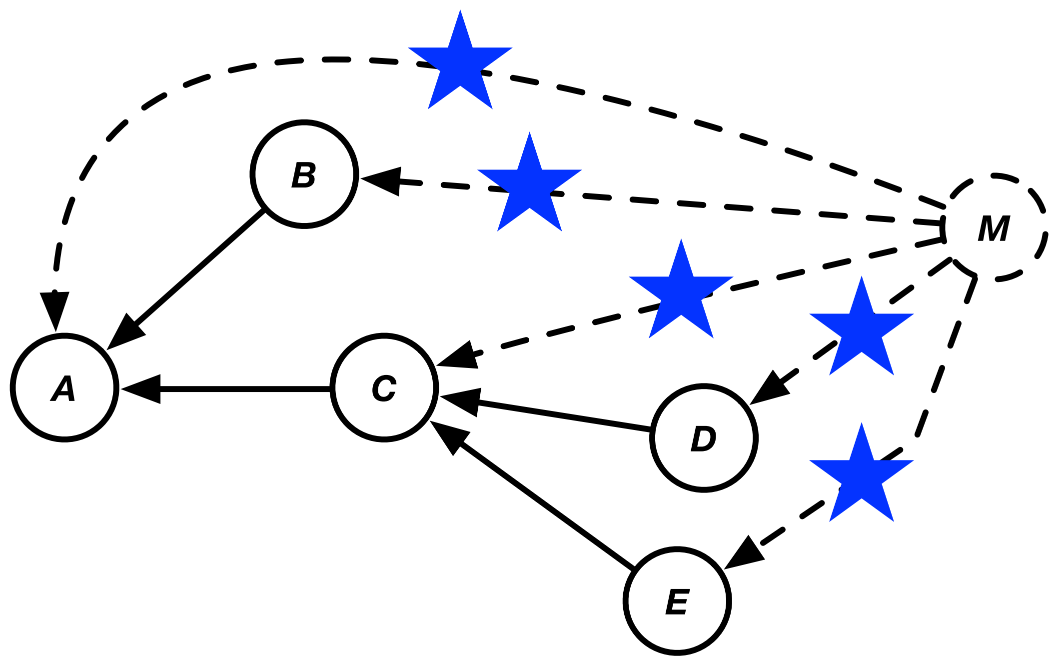
— **Forward Scheduling:**

- **We perform link scheduling starting from the flow generator**

— **Cannot effectively use the schedule combination technique**

	0	1	2	. .
c_1	*ME* *MD* *MC* *MB* *MA*	*BA* *DC* *EC*	*CA*	

	0	1	2	...
A	■	■	■	
B	■	■		
C	■	■	■	
D	■	■		
E	■	■		



For Example:

Using forward scheduling, node *A* is blocked in 3 slots, i.e., node *A* cannot be used for scheduling other flows in time slots 0, 1, 2

Technique 3: Reverse Scheduling

— Forward Scheduling:

- The set of ready transmissions initially includes the transmissions that origin from the flow generator
- Scheduling is started from time slot 0

Unfortunately, **forward scheduling does not effectively benefit from the schedule combination technique we proposed earlier**

	0	1	2	3	4	5	6	7	8	9	10	…
c_1	*ME* *MD* *MC* *MB* *MA*	*BA* *DC* *EC*	*CA*									

Transmissions *MA* and *BA* could be scheduled in the same entry as that of *CA* → **Blocking slots of *A* is reduced from 3 to 1**

Technique 3: Reverse Scheduling

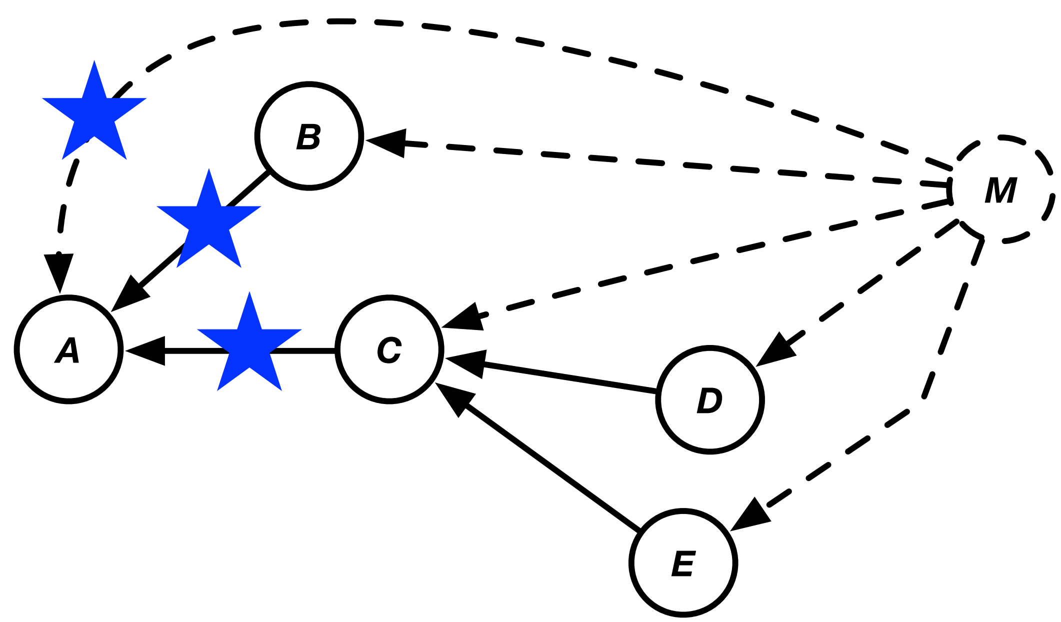
- We propose **"Reverse Scheduling"** to improve schedule combination

—Reverse Scheduling:

- **We perform link scheduling from the destination**

	...	5	6	7
c_1		*ME MD*	*MB DC EC MC*	*MA BA CA*

	...	5	6	7
A				■
B			■	■
C			■	■
D		■	■	
E		■	■	



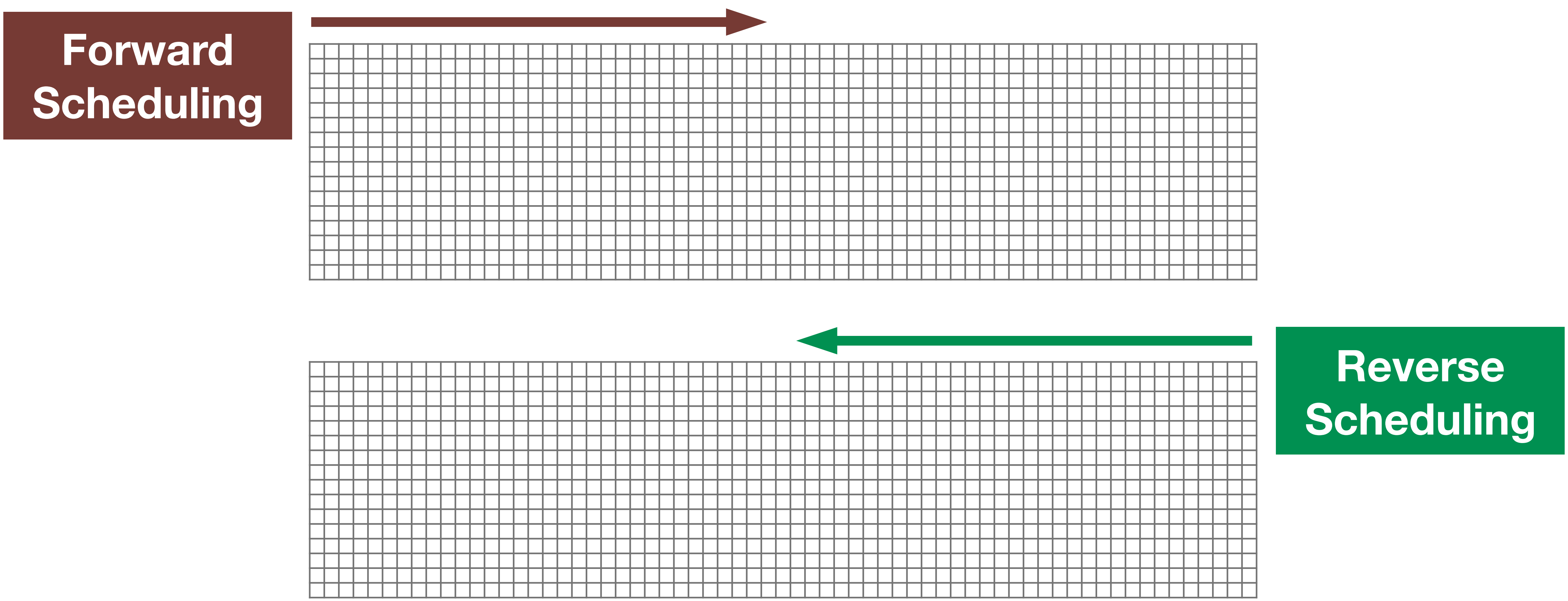
For Example:
Using reverse scheduling, node *A* is blocked in 1 slot only.

Technique 3: Reverse Scheduling

- To improve the efficiency of Flow Merging, we propose **"Reverse Scheduling"**

— Reverse Scheduling:

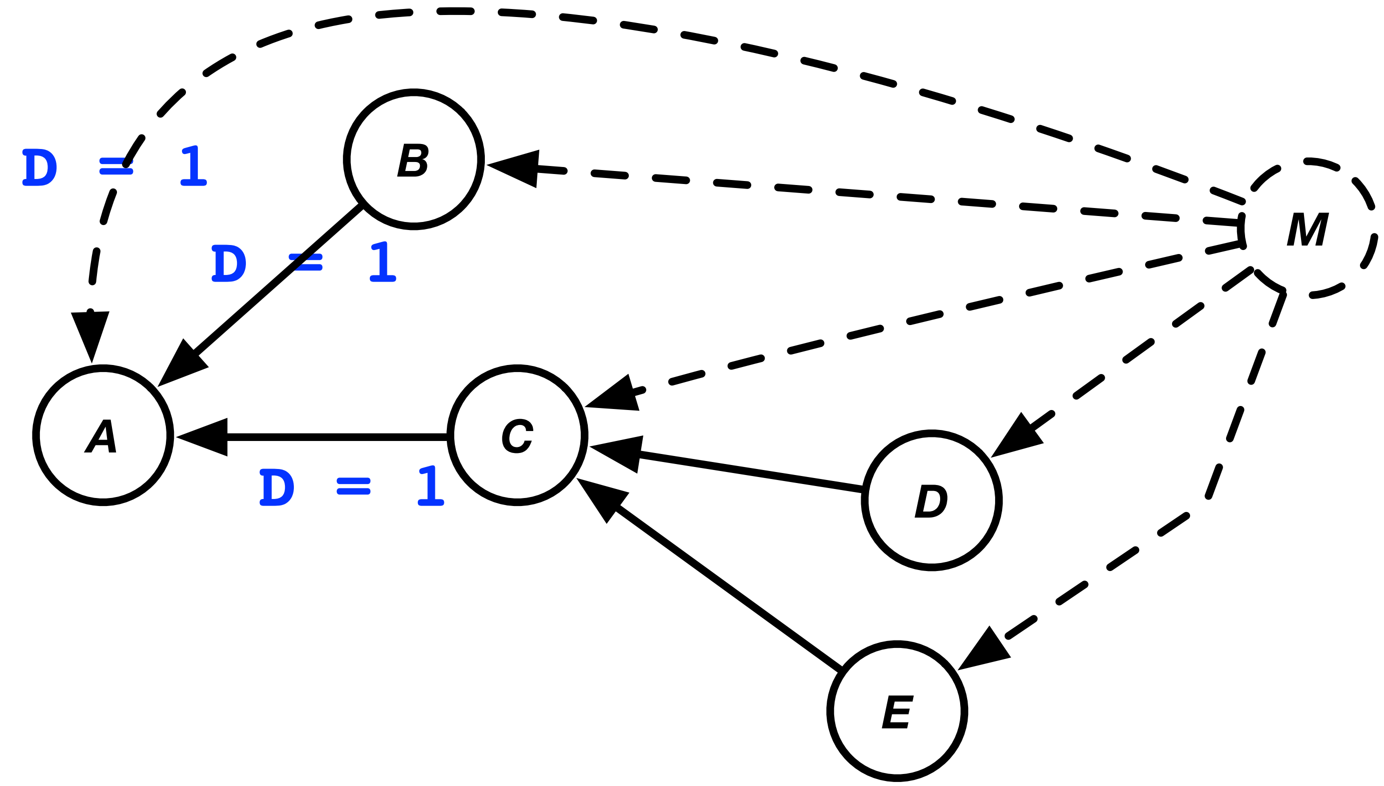
- **The set of ready transmissions initially includes the transmissions that deliver a flow to its destination**
- **Scheduling is started from the deadline of the flow**



Technique 3: Reverse Scheduling

Potential Communication Paths			
Path 1	*MA*		
Path 2	*MB*	*BA*	
Path 3	*MC*		
Path 4	*MD*	*DC*	
Path 5	*ME*	*EC*	*CA*

ready transmissions =
{(*CA*), (*BA*), (*MA*)}

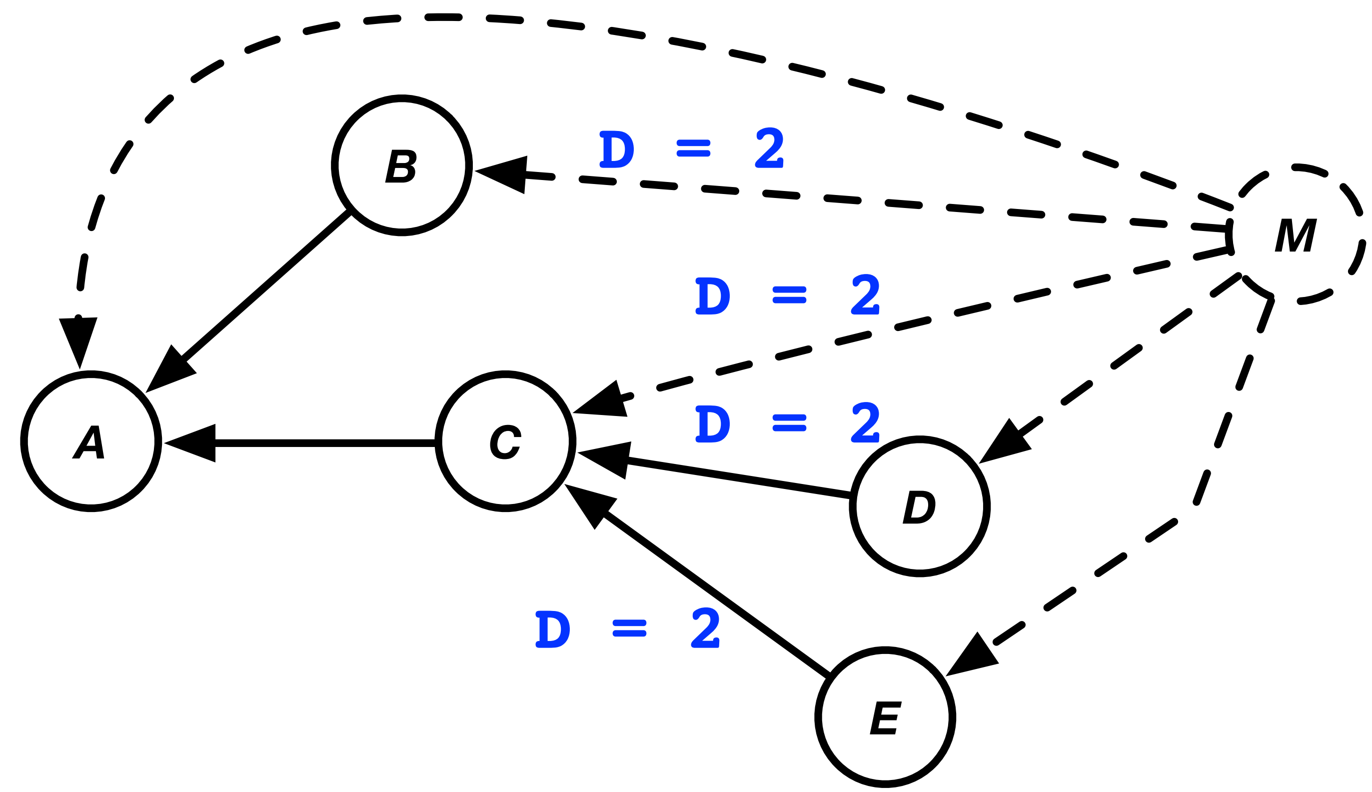


	0	1	2	3	4	5	6	7	8	9	10	...
c_1								*MA* *BA* *CA*				

Technique 3: Reverse Scheduling

Potential Communication Paths			
Path 1			
Path 2	*MB*		
Path 3	*MC*		
Path 4	*MD*	*DC*	
Path 5	*ME*	*EC*	

ready transmissions =
{(*MB*), (*DC*), (*EC*), (*MC*)}

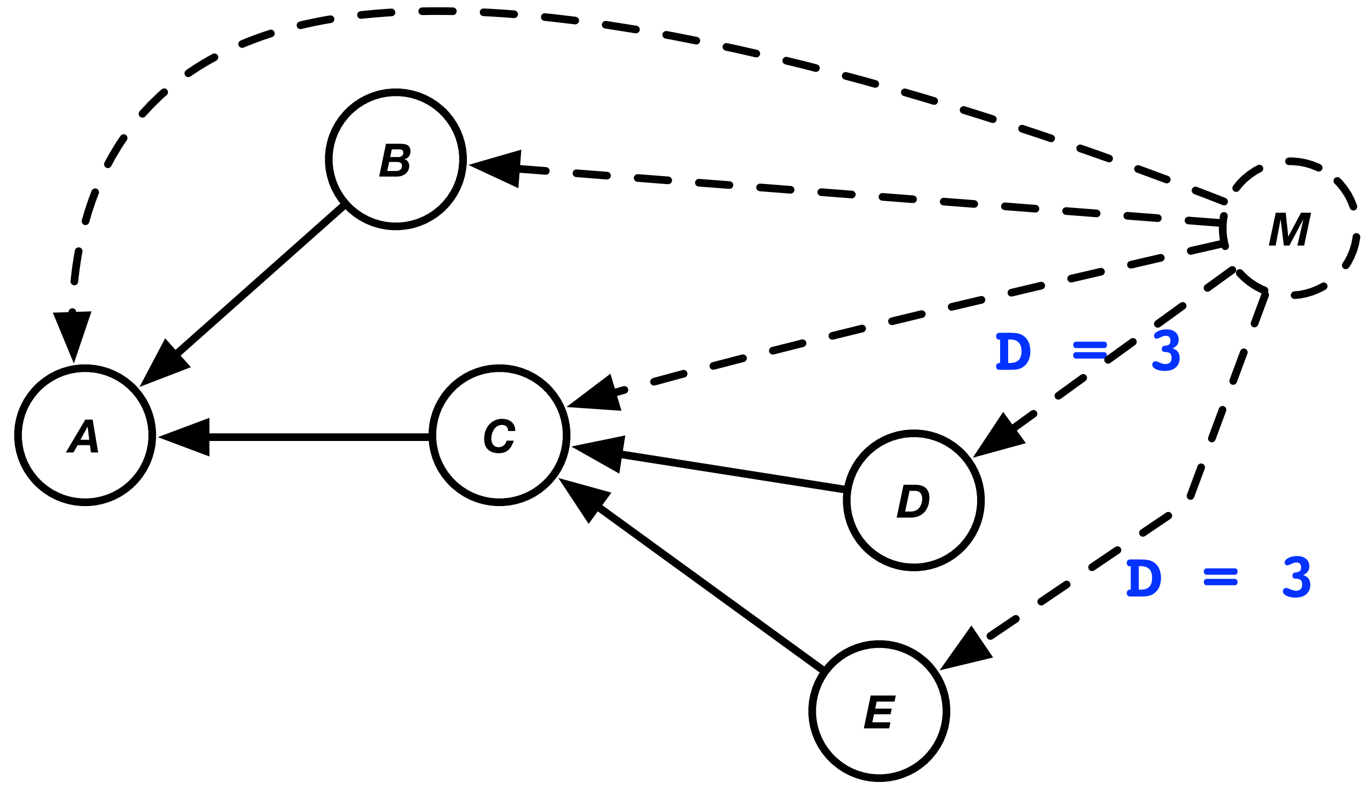


	0	1	2	3	4	5	6	7	8	9	10	...
c_1							*MB* *DC* *EC* *MC*	*MA* *BA* *CA*				

Technique 3: Reverse Scheduling

Potential Communication Paths			
Path 1			
Path 2			
Path 3			
Path 4	*MD*		
Path 5	*ME*		

ready transmissions =
{(*ME*), (*MD*)}

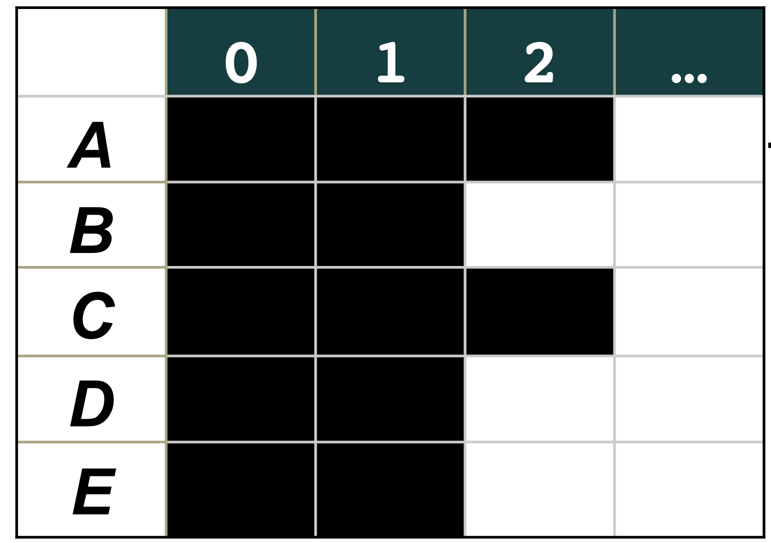


	0	1	2	3	4	5	6	7	8	9	10	...
C_1						*ME* *MD*	*MB* *DC* *EC* *MC*	*MA* *BA* *CA*				

Slot Blocking: Forward and Reverse Scheduling

Slot Blocking with “Forward Scheduling”

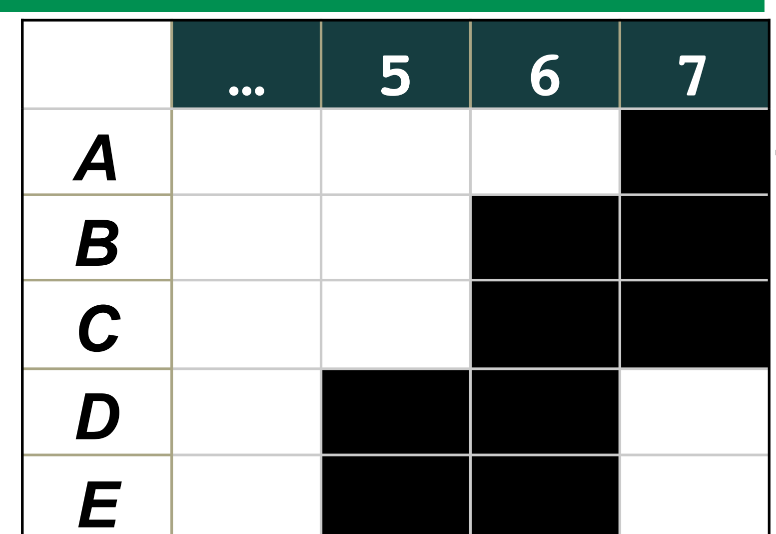
	0	1	2	. .
c_1	ME MD MC MB MA	BA DC EC	CA	



blocked in 3 slots

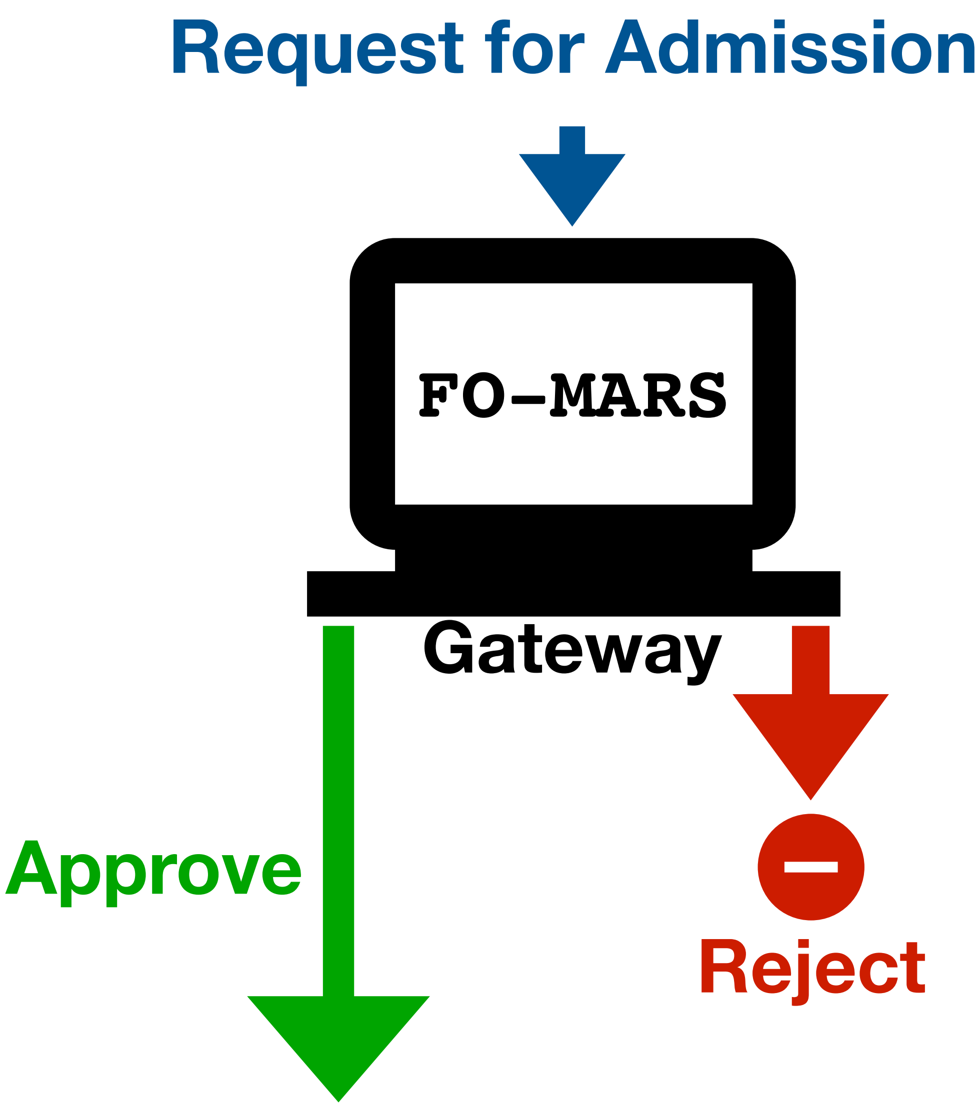
Slot Blocking with “Reverse Scheduling”

	...	5	6	7
c_1		ME MD	MB DC EC MC	MA BA CA



blocked in 1 slot

Flow-Ordered Mobility-Aware Scheduling Algorithm (FO-MARS)



Reserve bandwidth for:
new mobile node + existing nodes

Distribute the schedules for:
new mobile node + existing nodes

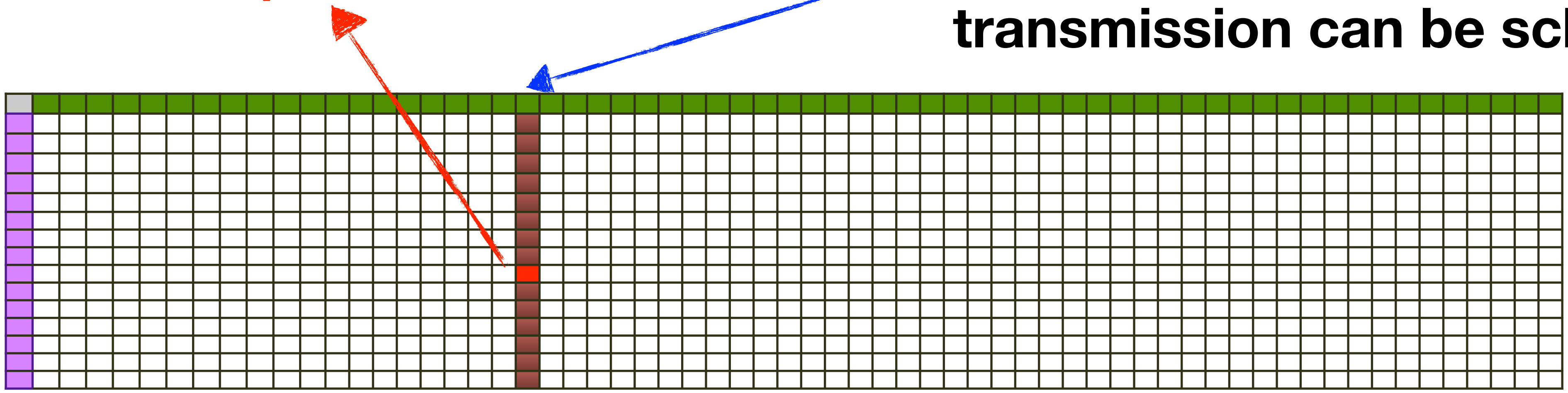
FO-MARS

- **Input:**
  - new mobile node's flows + existing flows
- **Output:**
  - Approve: The algorithm has scheduled all the flows
  - Reject: The algorithm cannot schedule all the flows
- **Operation Summary:**
  - Schedule flows in the order of their deadlines
  - Employ the techniques we proposed earlier

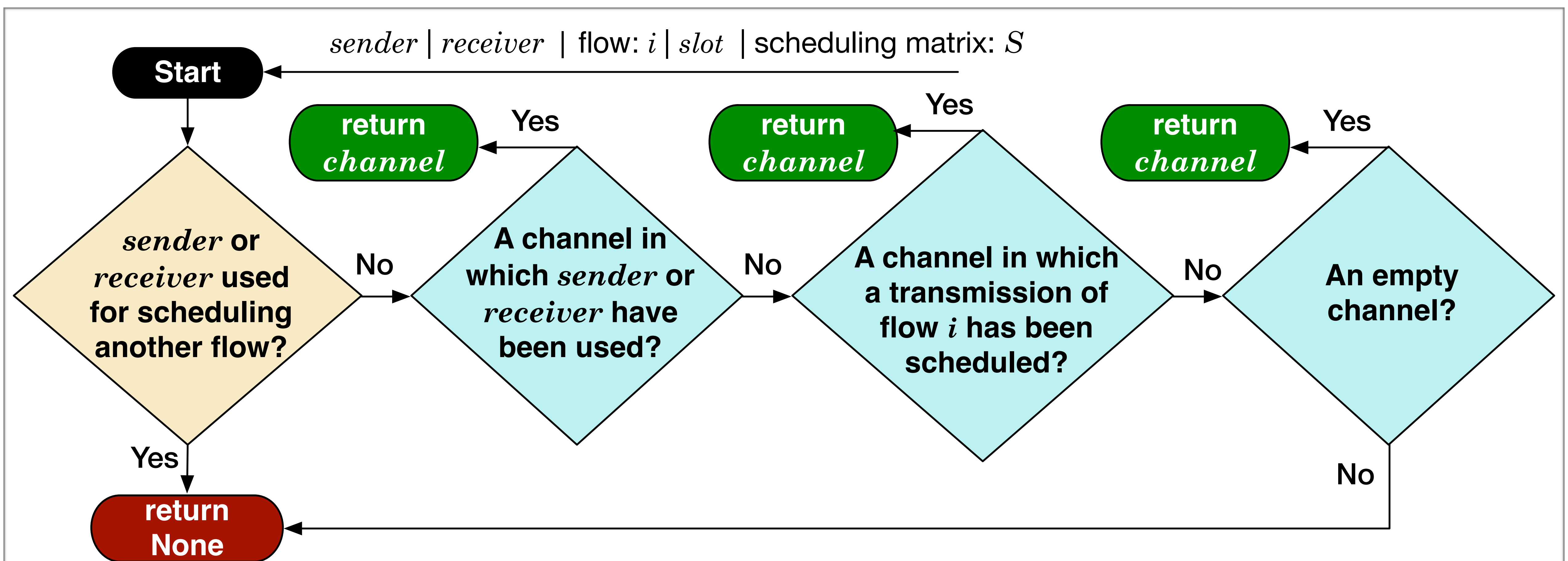
Algorithm `may-schedule()`

This is the best channel found
Based on the rules presented earlier

In time slot *s*, **what is the best channel** in which the given transmission can be scheduled?



Scheduling Matrix



Flow-Ordered Mobility-Aware Real-Time Scheduling (FO-MARS)

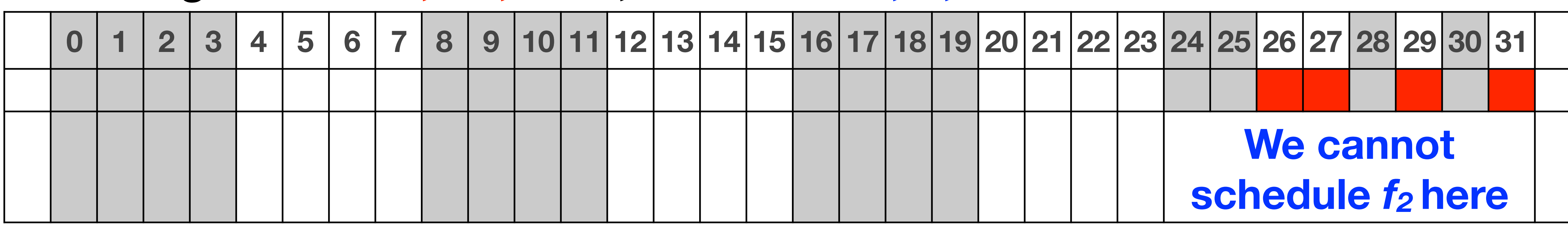
- **Schedules one flow at a time**
- **The order of flow scheduling is based on flows' deadlines**

NOTE: Scheduling a flow with longer period may reduce the schedulability chance of flows with shorter periods…

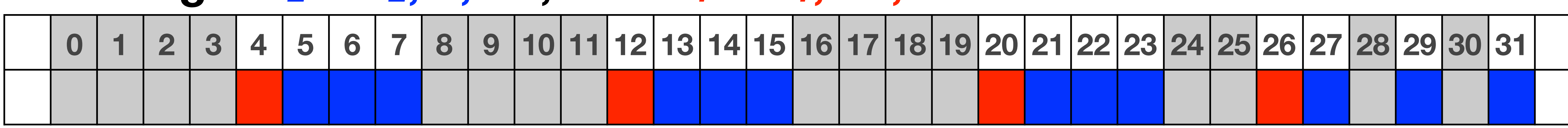
EXAMPLE:

- **f_1:<m_1, 32, 32> requires 4 transmissions**
- **f_2:<m_2, 8, 8> requires 4 transmissions**

Ordering 1: f_1:<m_1, 32, 32> , then f_2:<m_2, 8, 8>



Ordering 2: f_2:<m_2, 8, 8>, then f_1:<m_1, 32, 32>



Flow-Ordered Mobility-Aware Real-Time Scheduling (FO-MARS)

Two shortcomings of FO-MARS:

1. **When a request for bandwidth reservation is received, all the existing flows with shorter period must also be rescheduled**
   - **Significant control data dissemination**
   - **Long node join delay**

2. **A newly received schedule can be used at the beginning of the next hyper period**
   - **Long node join delay**

Flow-Ordered Mobility-Aware Real-Time Scheduling (FO-MARS)

We cannot switch to a new scheduling matrix at any time

✦ **Example:**

Existing Scheduling Matrix

…	s-1	s	s+1	s+2	…
				*	

↑ **Reception of new scheduling matrix** (at slot s+2)

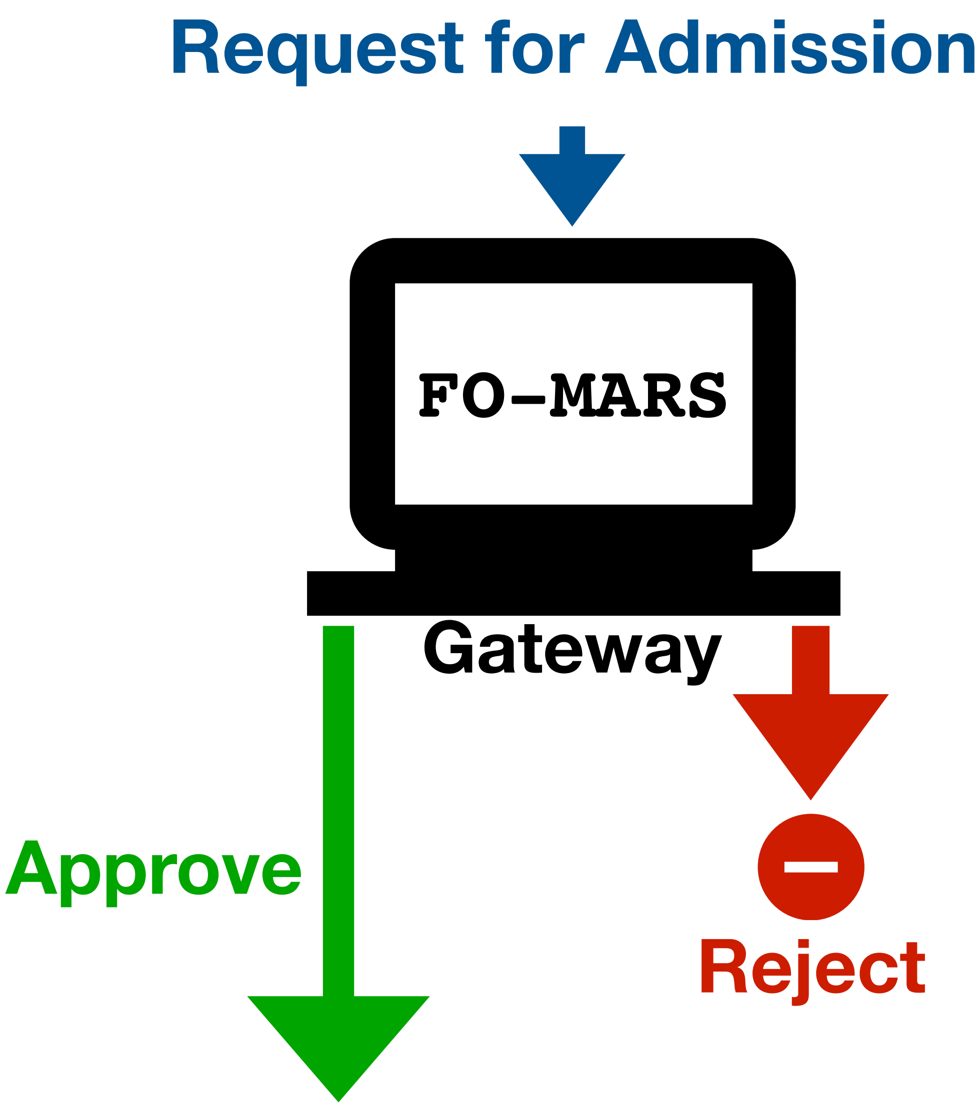
New Scheduling Matrix

…	s-1	s	s+1	s+2	…
		*			

↑ **Switch to the new scheduling matrix** (at slot s+2)

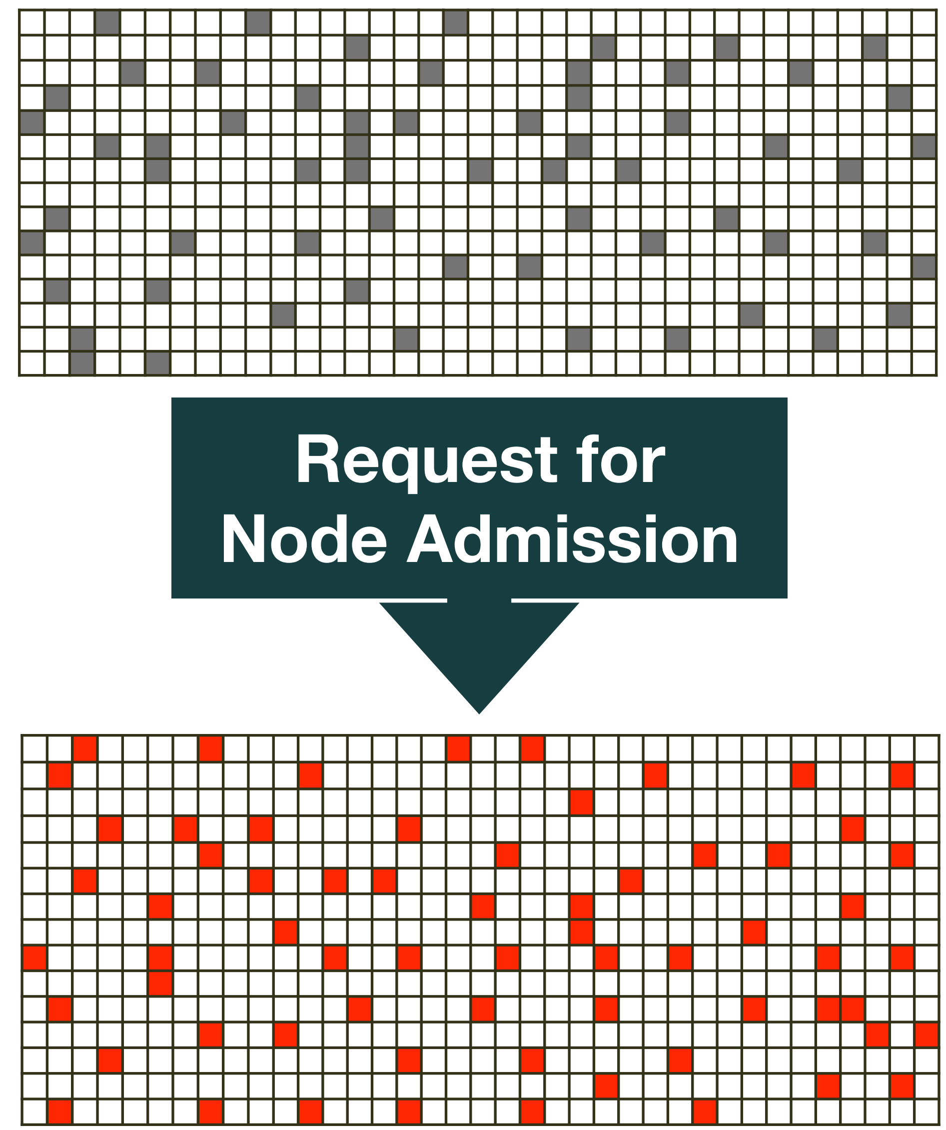
Transmission * never happens because that is scheduled for slot *s* in the new scheduling matrix

Flow-Ordered Mobility-Aware Scheduling Algorithm (FO-MARS)



Reserve bandwidth for:
new mobile node + existing nodes

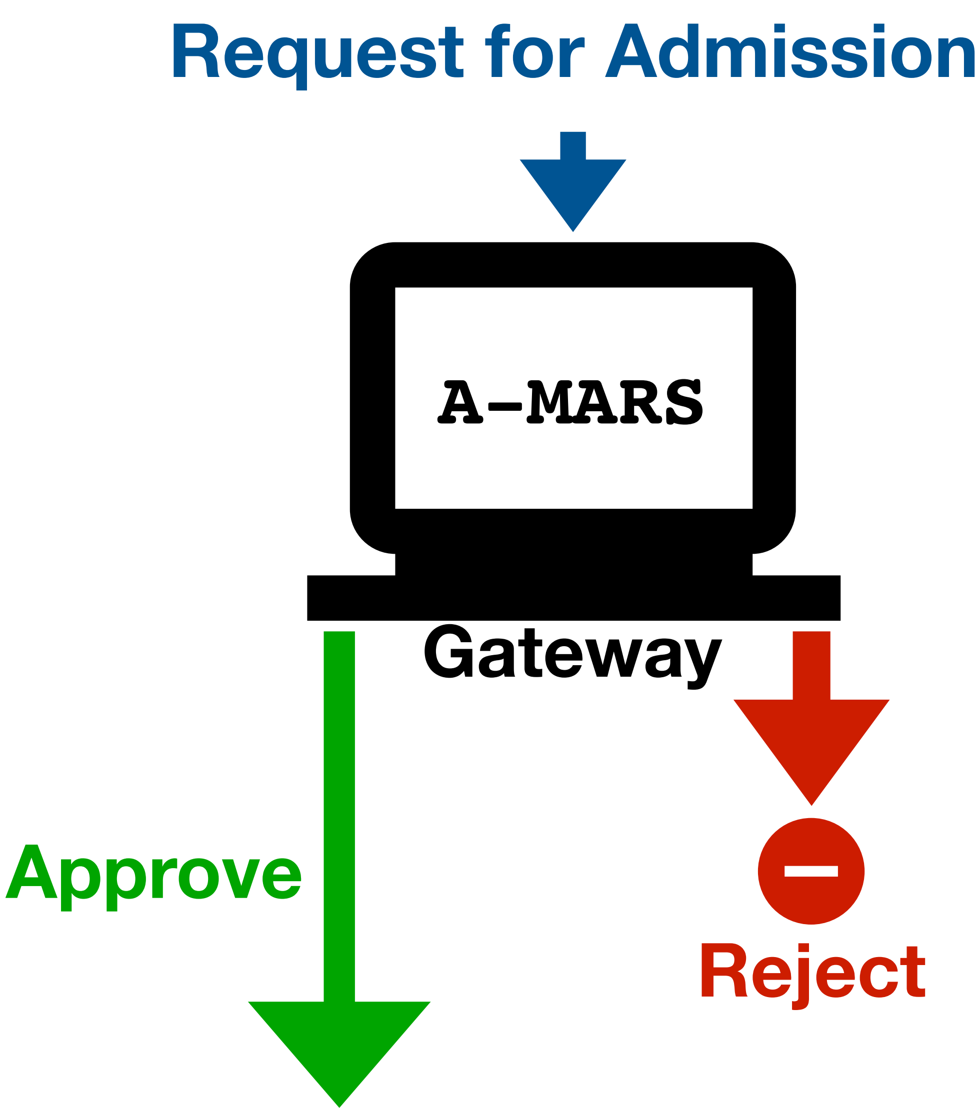
Distribute the schedules for:
new mobile node + existing nodes



All the red schedules should be disseminated

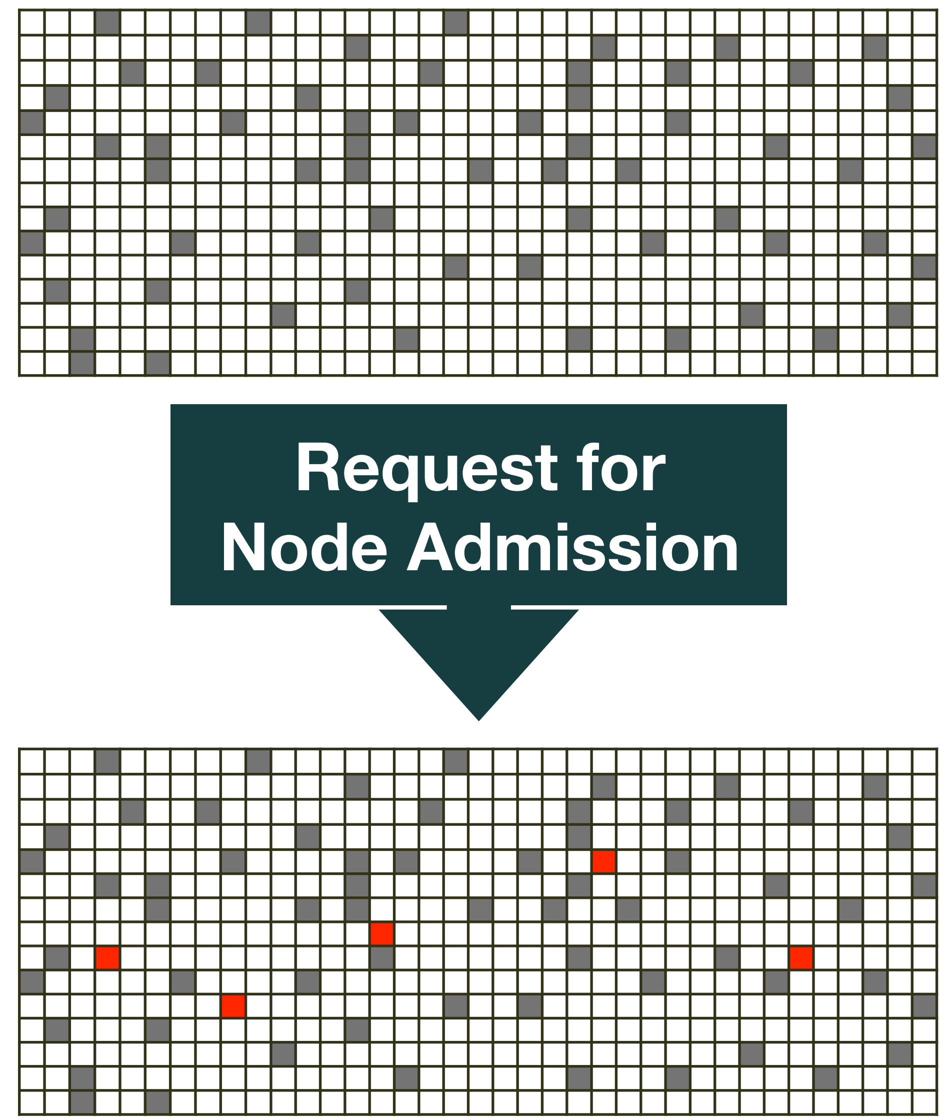
Problem — **LONG Node Admission Delay**

Additive Mobility-Aware Real-Time Scheduling (A-MARS)



Reserve bandwidth for:
new mobile node

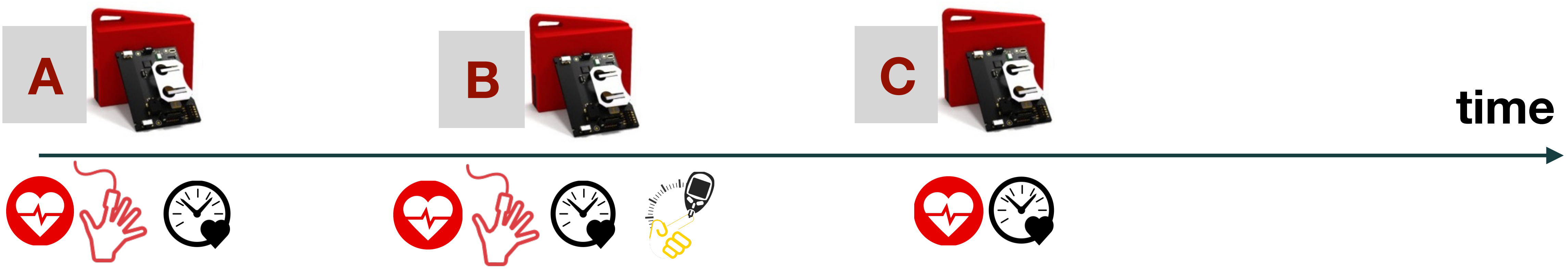
Distribute the schedules for:
new mobile node



Only the red schedules should be disseminated

Problem Solved **SHORT Node Admission Delay**

Challenge of Additive Scheduling



The flows of each node should be scheduled so that future mobile nodes can be scheduled as well

We need a smart bandwidth reservation algorithm that predicts the future to enhance the schedulability of future flows

Challenge of Additive Scheduling

- Assume $P_\gamma = 16$ and $P_\beta = 8$
- γ and β require 5 transmissions
- γ and β cannot be scheduled using the same time slots

Failed admission of flow β

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n^{th} Period of flow γ (Scheduling ok!)															
											γ	γ	γ	γ	γ
n^{th} Period of flow β (scheduling ok!)								$n+1^{th}$ Period of flow β (scheduling fail!)							
			β	β	β	β	β				γ	γ	γ	γ	γ

Successful admission of flow β

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n^{th} Period of flow γ (Scheduling ok!)															
						γ	γ						γ	γ	γ
n^{th} Period of flow β (scheduling ok!)								$n+1^{th}$ Period of flow β (scheduling ok!)							
	β	β	β	β	β	γ	γ	β	β	β	β	β	γ	γ	γ

How to know which slots should be used for scheduling a flow?

Flow Classes and Slot Prioritization

- Assume the network services a set of flow classes γ, β, α,…
- All the flows in a flow class have similar period and deadline
- **We prepare a prioritized list of slots for scheduling each flow class**

For flow class γ :

	Flow Class	Period/ Deadline
Heart Rate	γ	P_γ, D_γ
Pulse Oximetry	β	P_β , D_β
Blood Pressure	α	P_α, D_α

$D_\alpha < D_\beta < D_\gamma$

How using a slot for scheduling γ would affect the schedulability of β and α

Prepare a prioritized list of slots for scheduling a flow γ

Slot Prioritization

- **We propose the notion of Potential Utilization (PU) to measure the effect of choosing each slot**

$$\text{demand} = \frac{\text{percentage of flows belonging to class } \beta \times \text{\#required transmissions}}{\text{available slots}}$$

Time Slots	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	n^{th} Period of flow β								$n+1^{th}$ Period of flow β							
PU for class β	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31

(0.5 X 5)/8

Choose slot 7 and update PU

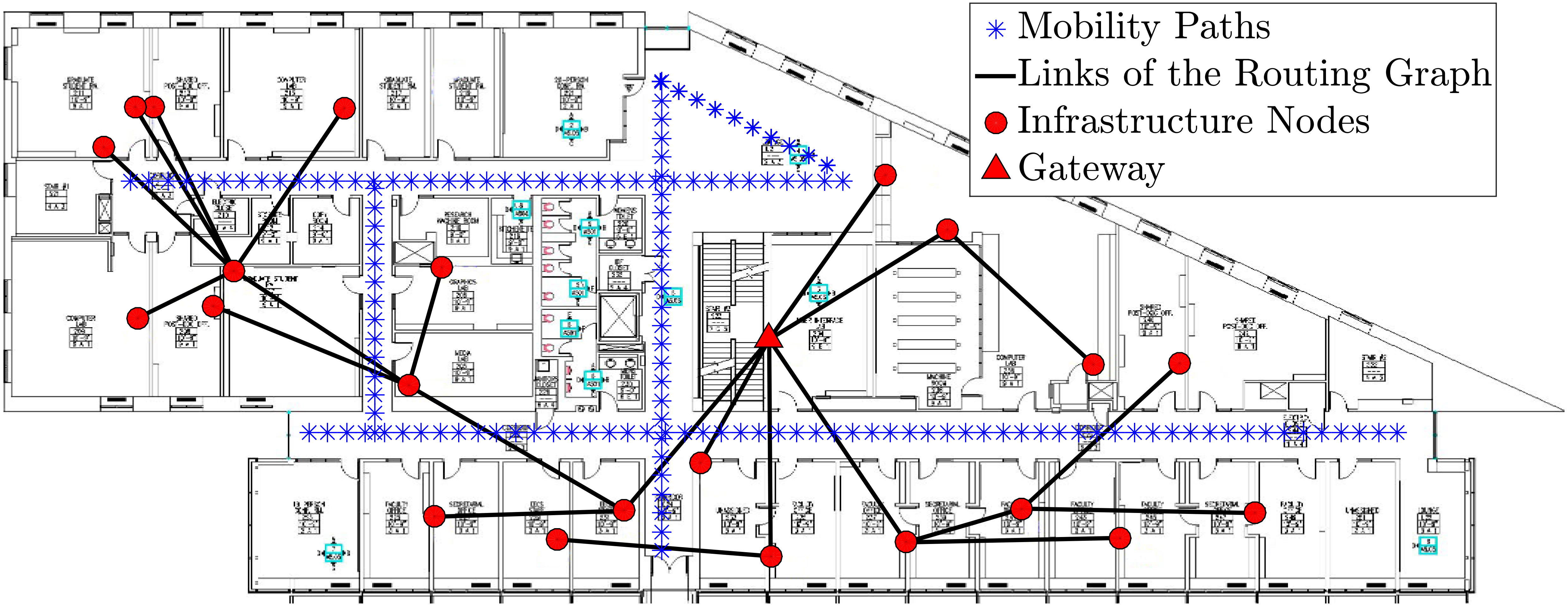
Time Slots	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	n^{th} Period of flow β								$n+1^{th}$ Period of flow β							
PU for class β	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31

choosing from these slots: increase PU by 0.39

choosing from these slots: increase PU by 0.31

Performance Evaluation

Setup

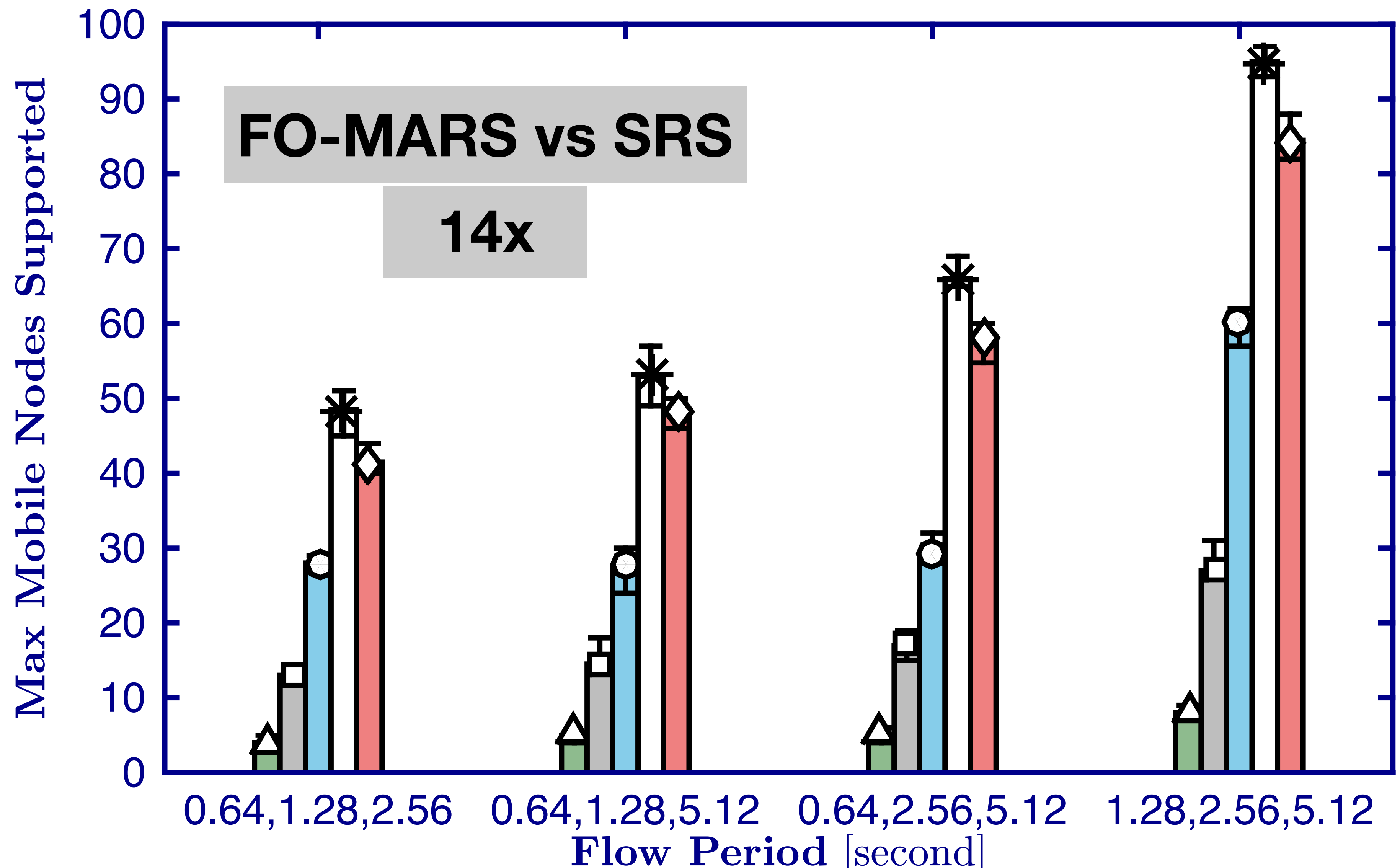


- Trace-driven simulator using exhaustive floor plan measurements
- Realistic and repeatable experimentation

Scalability: How Many Mobile Nodes can be Supported?

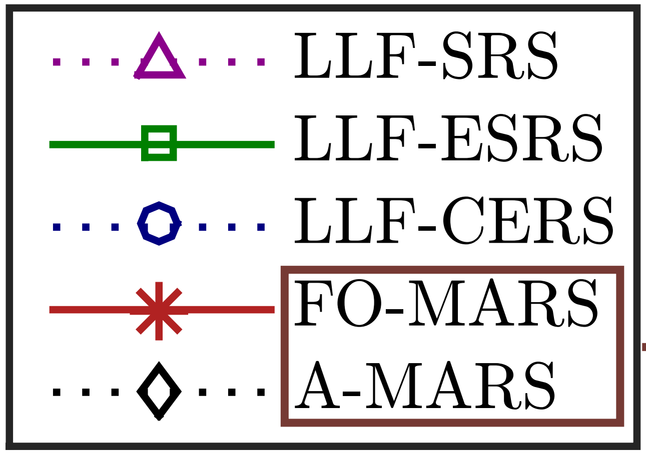
- LLF-SRS → SRS: Designed for Static Networks
- LLF-ESRS → ESRS: SRS + Flow Coordination
- LLF-CERS → CERS: SRS + Flow Coordination + Schedule Combination
- FO-MARS
- A-MARS

FO-MARS, A-MARS → **Proposed Algorithms**

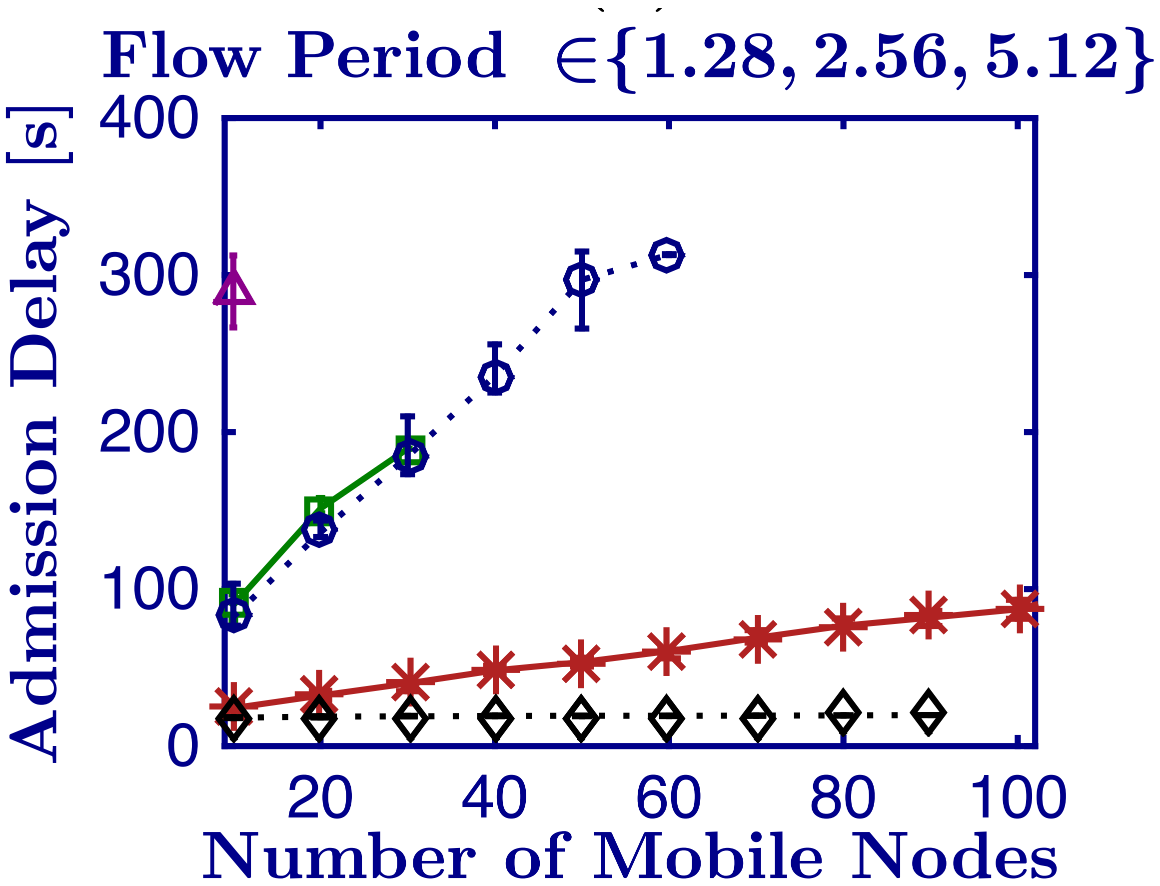


Mobile nodes generate reports with different rates

Admission Delay

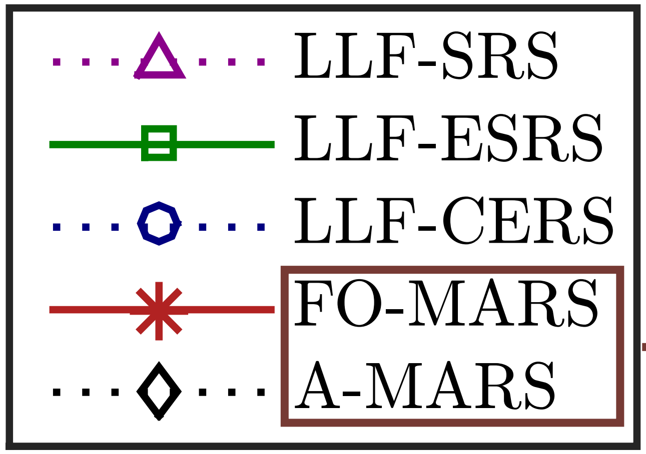


- SRS: Designed for Static Networks
- ESRS: SRS + Flow Coordination
- CERS: SRS + Flow Coordination + Schedule Combination
- Proposed Algorithms



A-MARS'S Admission delay < 20 seconds

Network Lifetime

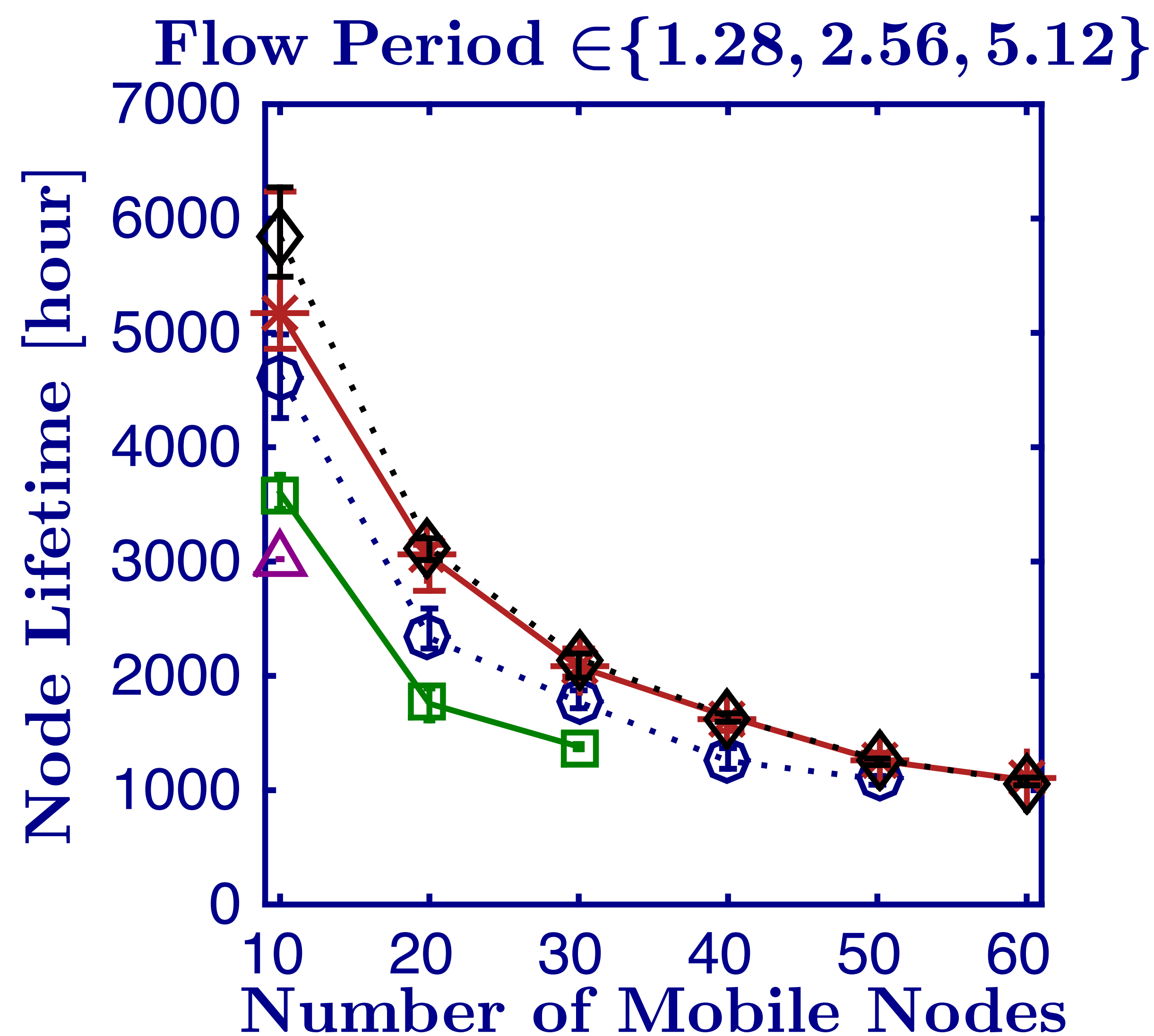


SRS: Designed for Static Networks

ESRS: SRS + Flow Coordination

CERS: SRS + Flow Coordination + Schedule Combination

Proposed Algorithms



Higher network lifetime achieved with FO-MARS and A-MARS

Conclusion

- Real-time wireless networks can be used in mission-critical applications such as industrial process control and medical monitoring
- Existing real-time networks do not efficiently handle network dynamics such as mobility and flow addition/removal
- We proposed scheduling techniques for efficient bandwidth reservation for mobile nodes
- We proposed an additive scheduling algorithm for effective handling of flow admission and removal
- Compared to the algorithms designed for stationary real-time networks, our proposed network admits a significantly higher number of mobile nodes, archives short admission delay, and handles network dynamics efficiently

Acknowledgement

Behnam Dezfouli
University of Iowa
Iowa City, IA, USA

Marjan Radi
University of Iowa
Iowa City, IA, USA

Octav Chipara
University of Iowa
Iowa City, IA, USA

This work was supported by the National Science Foundation and the Roy J. Carver Charitable Trust