Embedded Software Engineering

3 Unit Course, Spring 2002

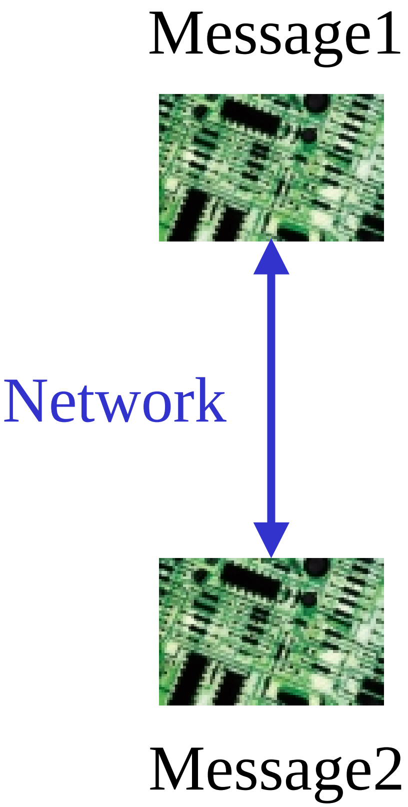
EECS Department, UC Berkeley

Chapter 3: RT Communication

Christoph Kirsch

www.eecs.berkeley.edu/~fresco/giotto/course-2002

Real-Time Communication

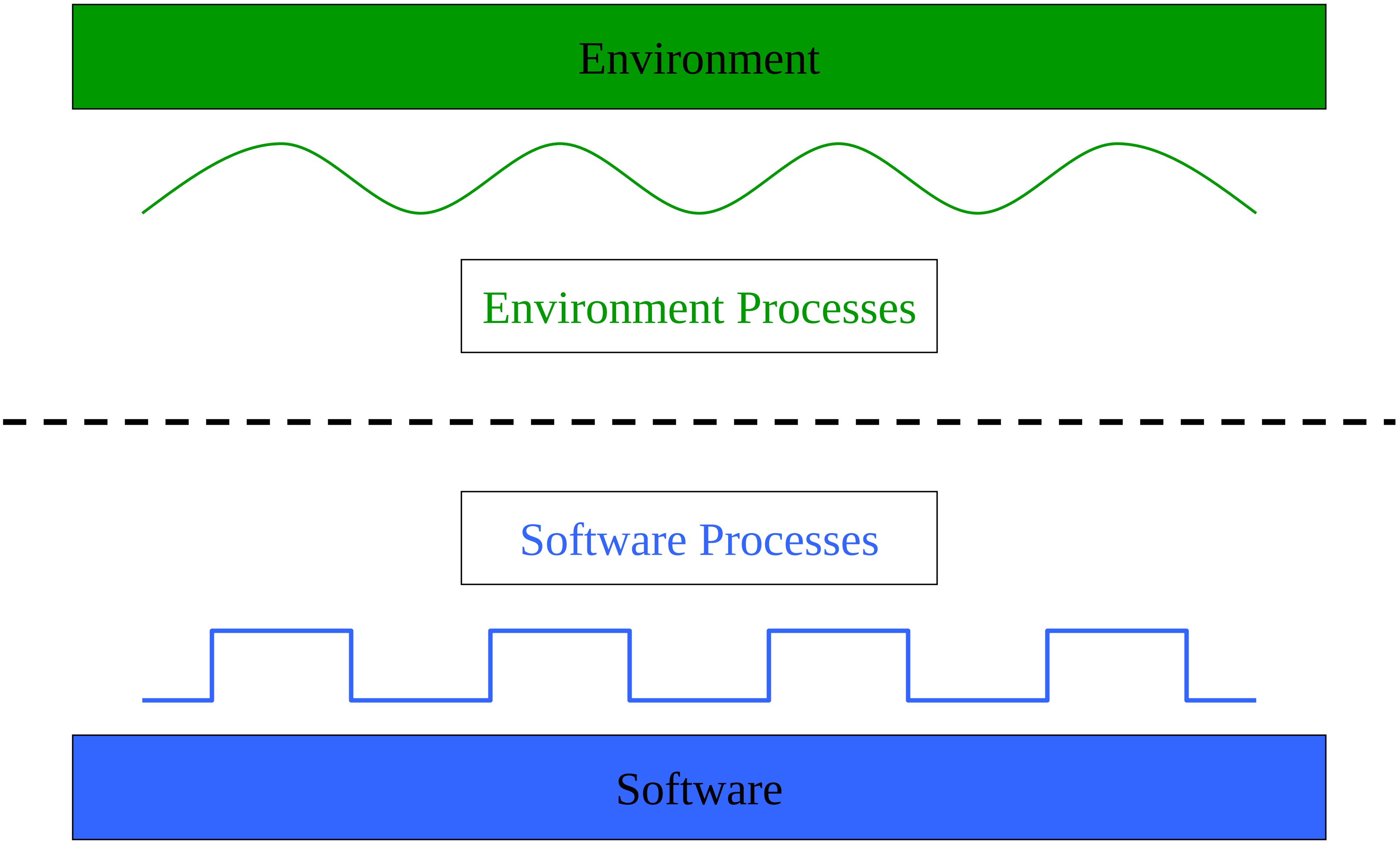


Message1

Network

Message2

Embedded Software



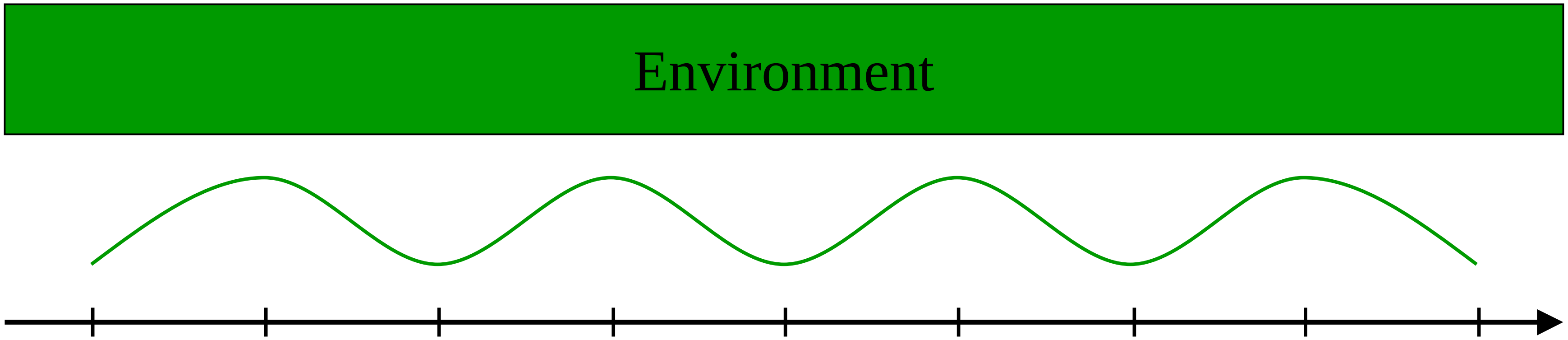
Environment

Environment Processes

Software Processes

Software

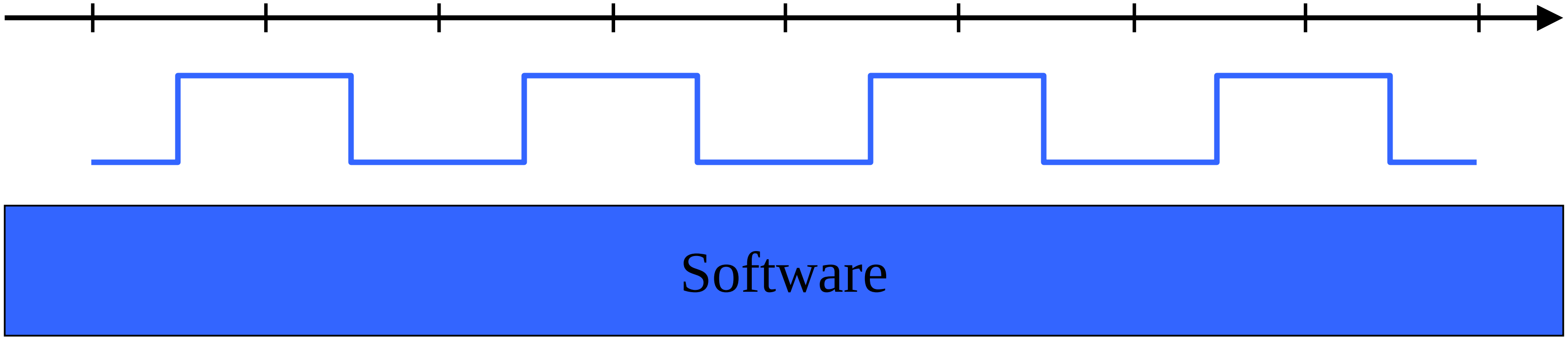
Platform Time is Platform Memory



Environment

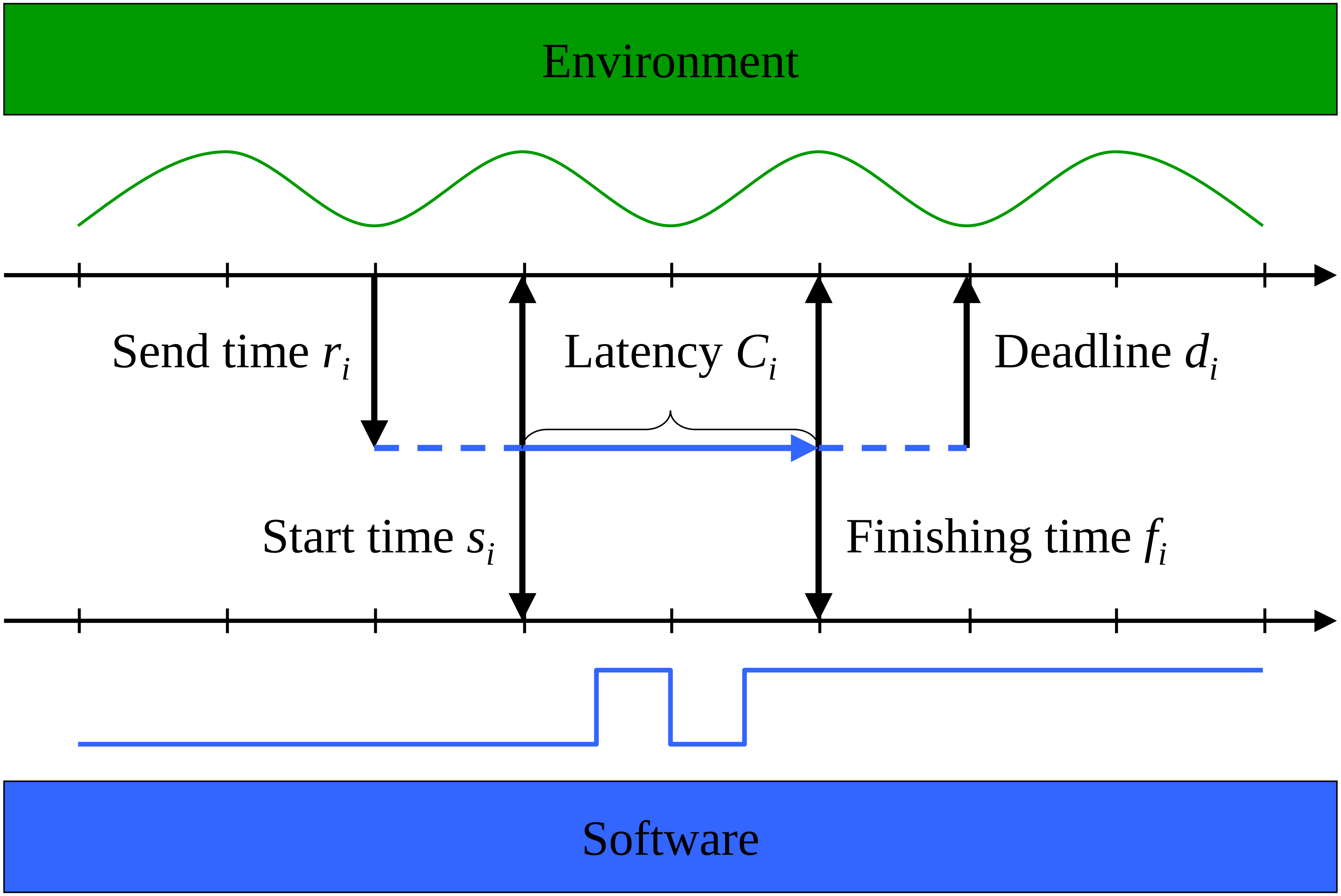
- Programming as if there is enough platform time

- Implementation checks whether there is enough of it



Software

A Message M_i



Environment

Send time r_i

Latency C_i

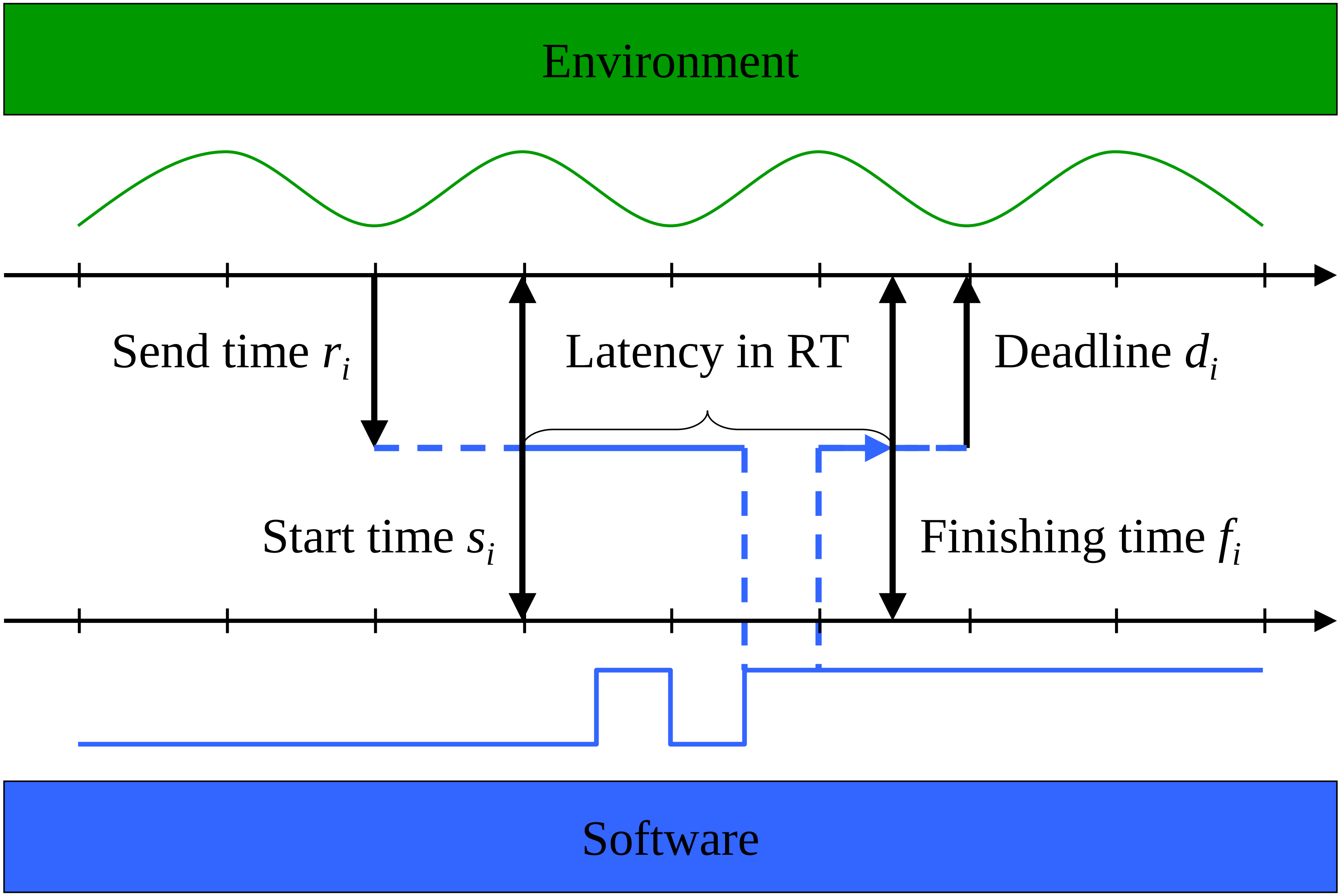
Deadline d_i

Start time s_i

Finishing time f_i

Software

Preemption



Environment

Send time r_i

Latency in RT

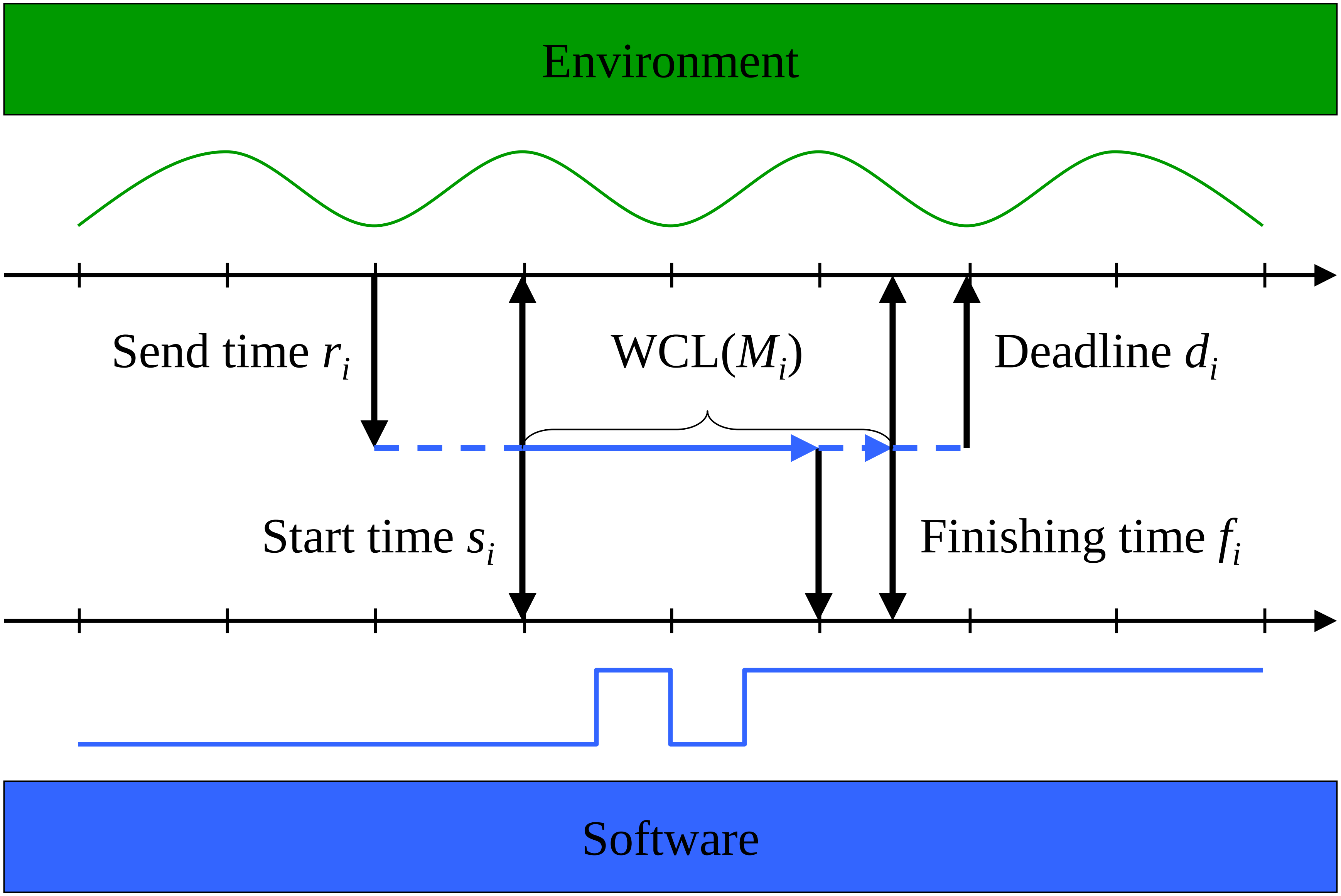
Deadline d_i

Start time s_i

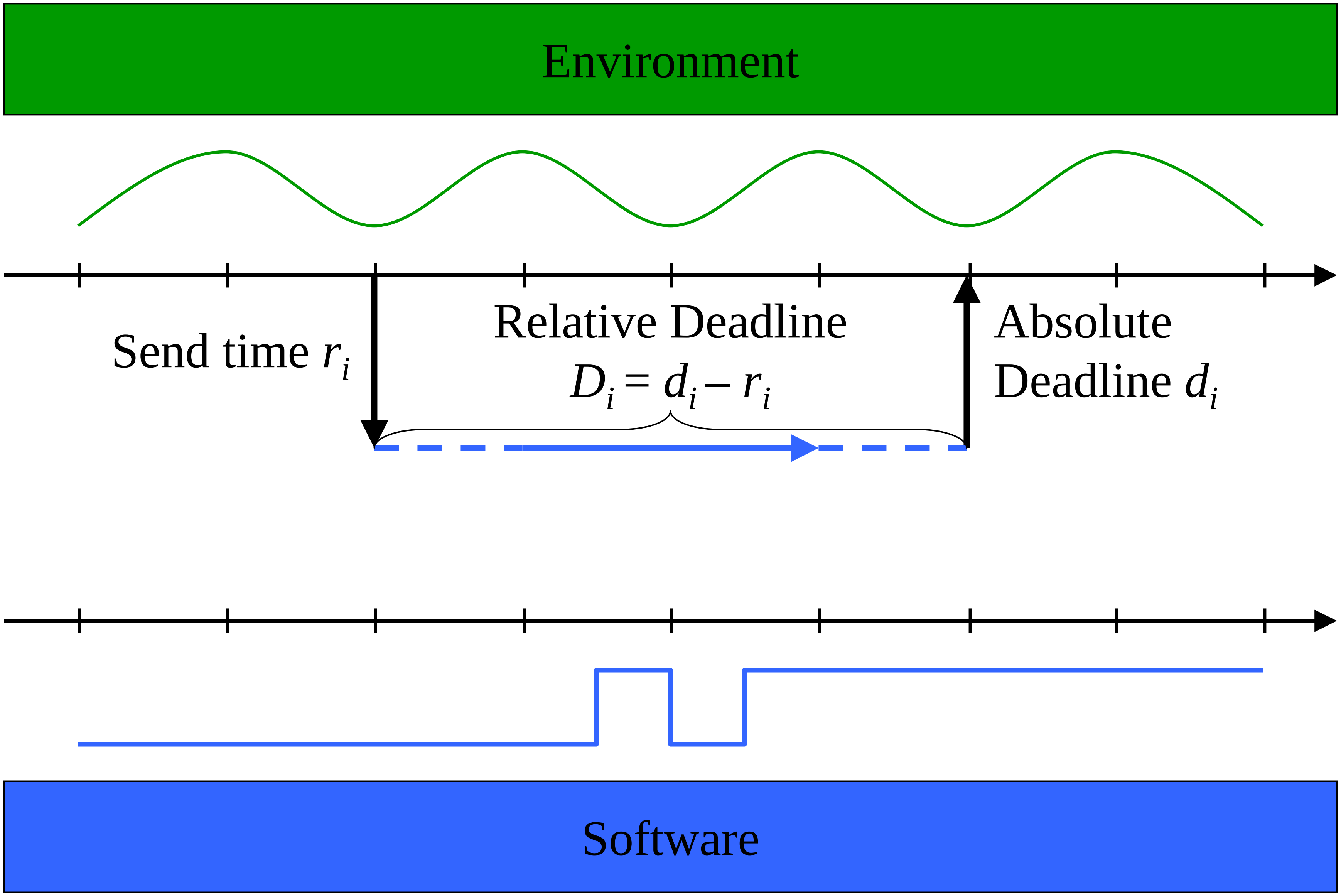
Finishing time f_i

Software

Worst-Case Latency: WCL(M_i)



Relative Deadline D_i



Environment

Send time r_i

Relative Deadline $D_i = d_i - r_i$

Absolute Deadline d_i

Software

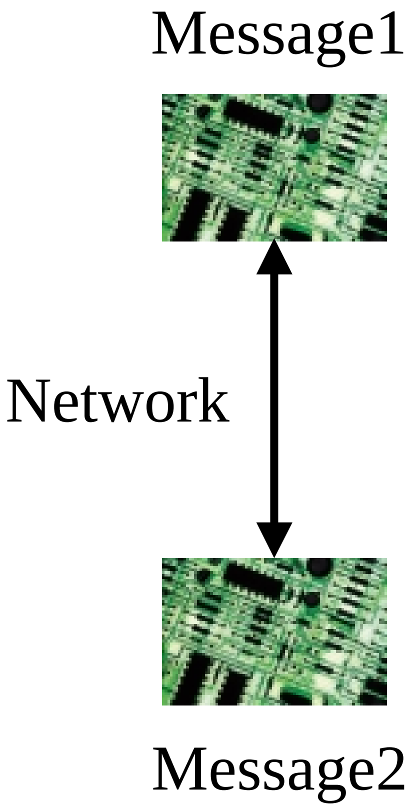
Triggering a Message M_i

- *Periodically*: A *periodic message* M_i is a message with a-priori known send times regularly activated at a constant rate P_i
    - The first send time r_i is called the *phase* ϕ_I
    - The send time of the n-th instance is given by $r_i + (n - 1)\, P_i$
    - P_i is called the *period* of M_i

- *Sporadically*: A *sporadic message* M_i is a message with a minimum (*interarrival*) time between any two send times

- *Aperiodically*: An *aperiodic message* M_i is a message without any constraints on the send times

Explicit Flow Control



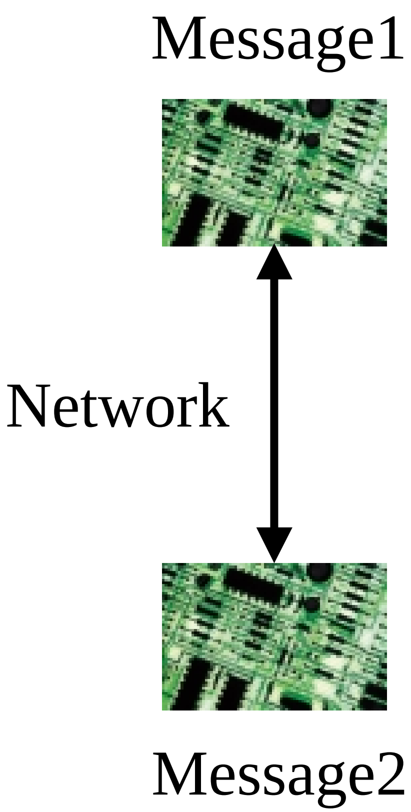
Message1

Network

Message2

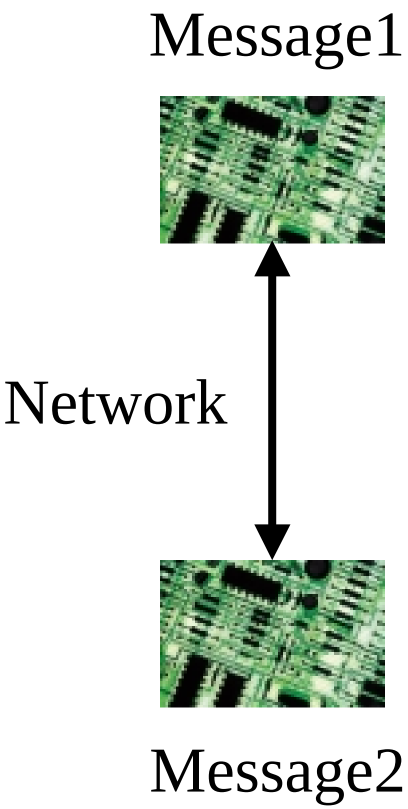
- Send time not known a priori
- Sender can detect errors

Implicit Flow Control



- Send time is known a priori
- Receiver can detect errors

Explicit Flow Control: Priority



Message1

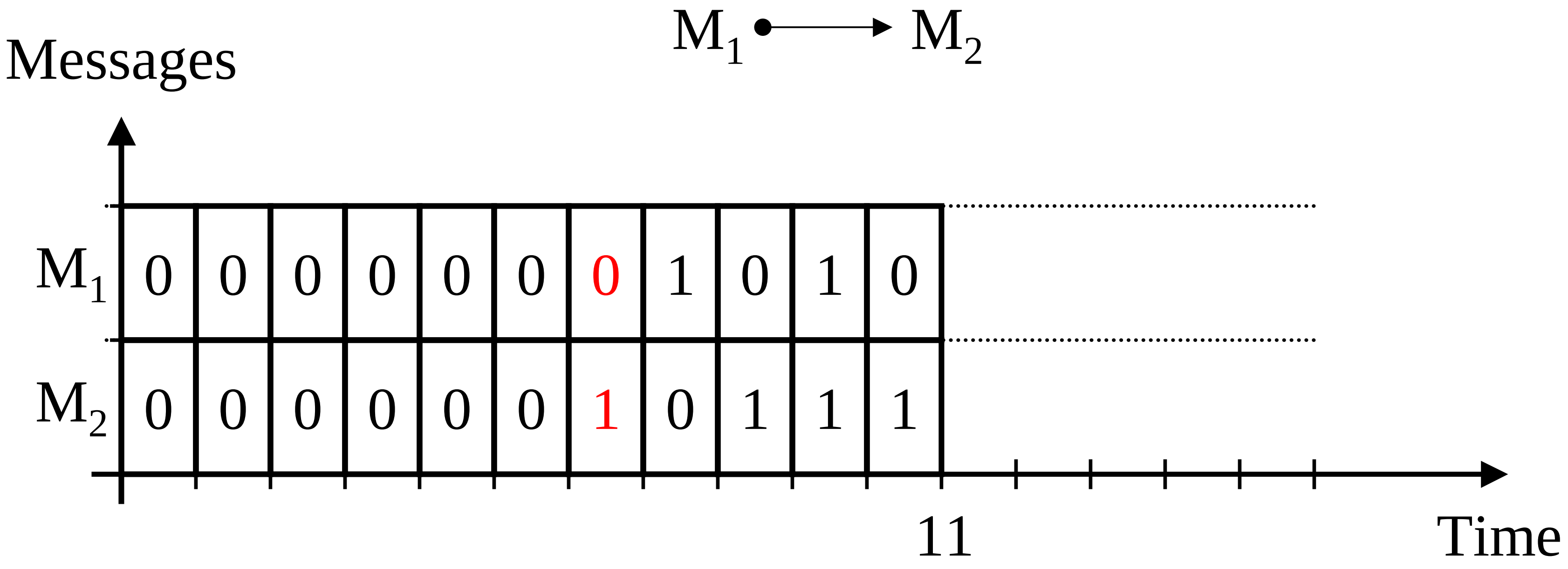
Network

Message2

Medium-Access Protocols:

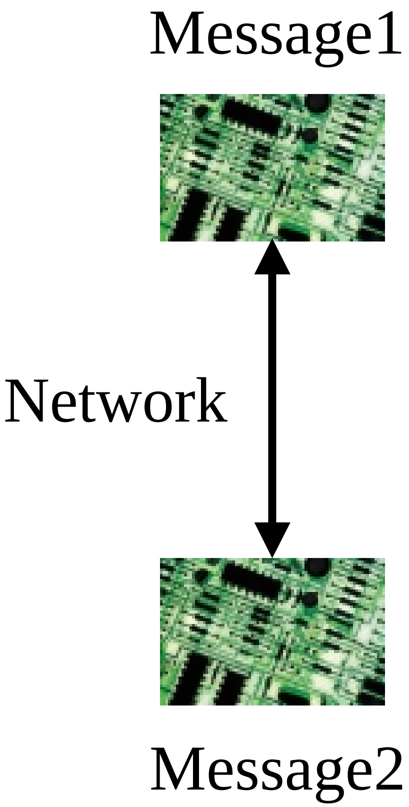
- CSMA/CD - LON, Echelon 1990
- CSMA/CA - CAN, Bosch 1990
- FTDMA - Byteflight, BMW 2000
- FTDMA - Flexray, Daimler/BMW 2001

Control Area Network



M_1	0	0	0	0	0	0	0	1	0	1	0
M_2	0	0	0	0	0	0	1	0	1	1	1

Implicit Flow Control: Time



Message1

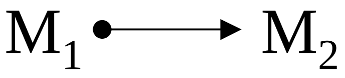
Network

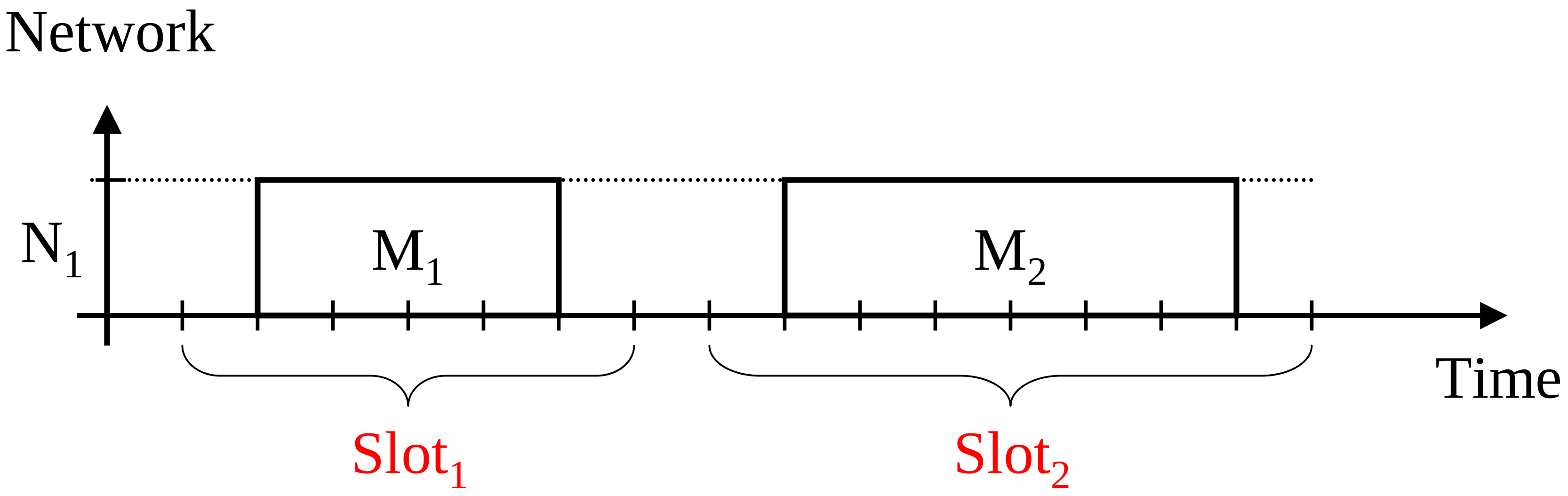
Message2

Medium-Access Protocols:

- TDMA - TTP, Kopetz 1993
- FTDMA - Flexray, Daimler/BMW 2001

Time-Triggered Protocol





Fault Tolerance

- A *value fault* causes an incorrect (physical or logical) value to be computed, transmitted, or received
- A *timing fault* causes a value to be computed, transmitted, or received at the wrong time (too early, too late, not at all)
- A *spatial proximity fault* is a fault where all matter in some specified volume is destroyed

- Definitions from Rushby, EMSOFT 2001

A Fault Model

- A *fault model* classifies the effects of faults in order to study the algorithms for fault tolerance

- A *manifest effect* of a fault can always be detected reliably: e.g., a fault that makes a host cease transmitting messages

- A *symmetric effect* of a fault is any effect that is the same for all observers: e.g., off-by-1 error

- An *arbitrary effect* of a fault is unconstrained:
  e.g., an asymmetric or *Byzantine* effect that is perceived differently by different observers

Fault Hypotheses and FCUs

- A *fault containment unit* (FCU) is an *independent* unit:
  faults propagate neither out of an FCU nor
  into an FCU ("common mode failure")

- A *fault mode* describes the kind of behavior
  a faulty FCU may exhibit

- A *fault hypothesis* describes the FCUs of an architecture
  as well as the fault modes to be tolerated and
  their maximum *number* and *arrival rate*

Redundancy

- Tolerating arbitrary fault modes does not require to justify *assumptions* about more specific fault modes
- Introducing redundancy *reduces* fault modes (e.g., by fail-silence)
- In general, it requires more redundancy to tolerate an arbitrary fault than a symmetric fault, which in turn requires more redundancy than a manifest fault
- E.g., there is a clock synchronization algorithm that tolerates a arbitrary faults, s symmetric faults, and m manifest faults that occur simultaneously, provided there are n FCUs with:

$$n > 3a + 2s + m$$

Protocol Services

- Fault-tolerant clock synchronization
- Fault-tolerant message transfer

- *Replication* requires agreement:
    - approximate (Problem: diverging state)
    - exact (Problem: interactive consistency/Byzantine agreement)

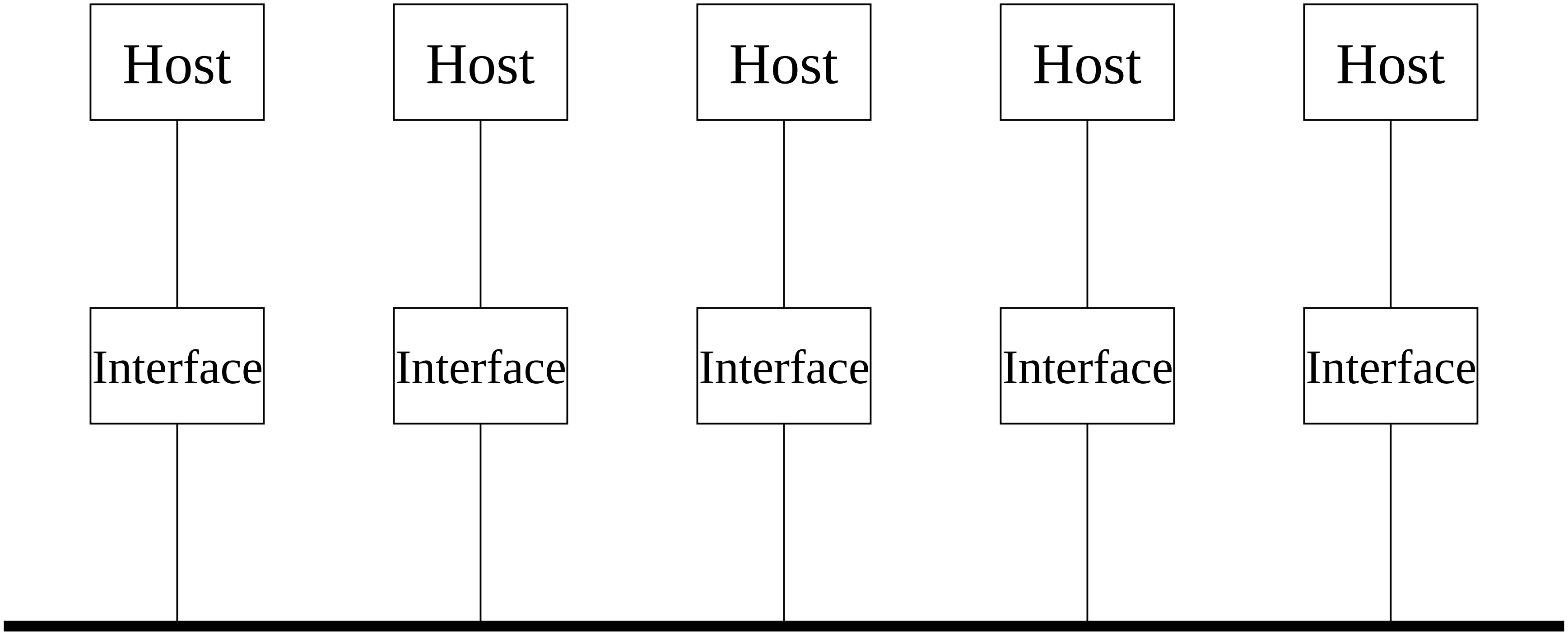
- *Interactive consistency* requires:
    - Agreement: All non-faulty receivers obtain the same message
    - Validity: If the transmitter is non-faulty, then non-faulty receivers obtain the message actually sent

Membership Service

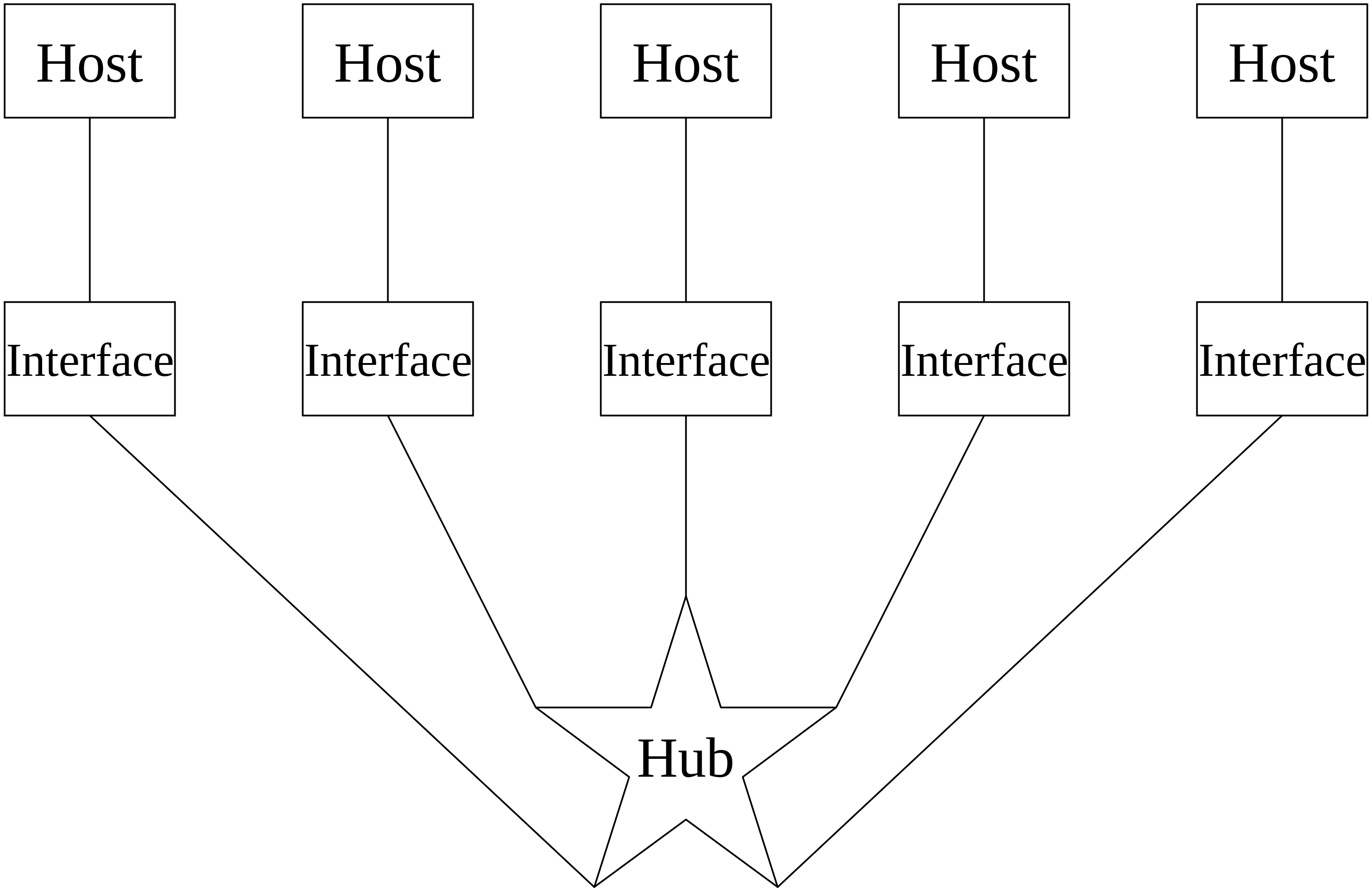
- Interactive consistency can be implemented by a *membership service* that must satisfy:
  - Agreement: The membership lists of all non-faulty nodes are the same
  - Validity: The membership lists of all non-faulty nodes contain all non-faulty nodes and at most one faulty node

- *Clique avoidance* weakens validity because non-faulty nodes may be excluded (that can later attempt to rejoin)

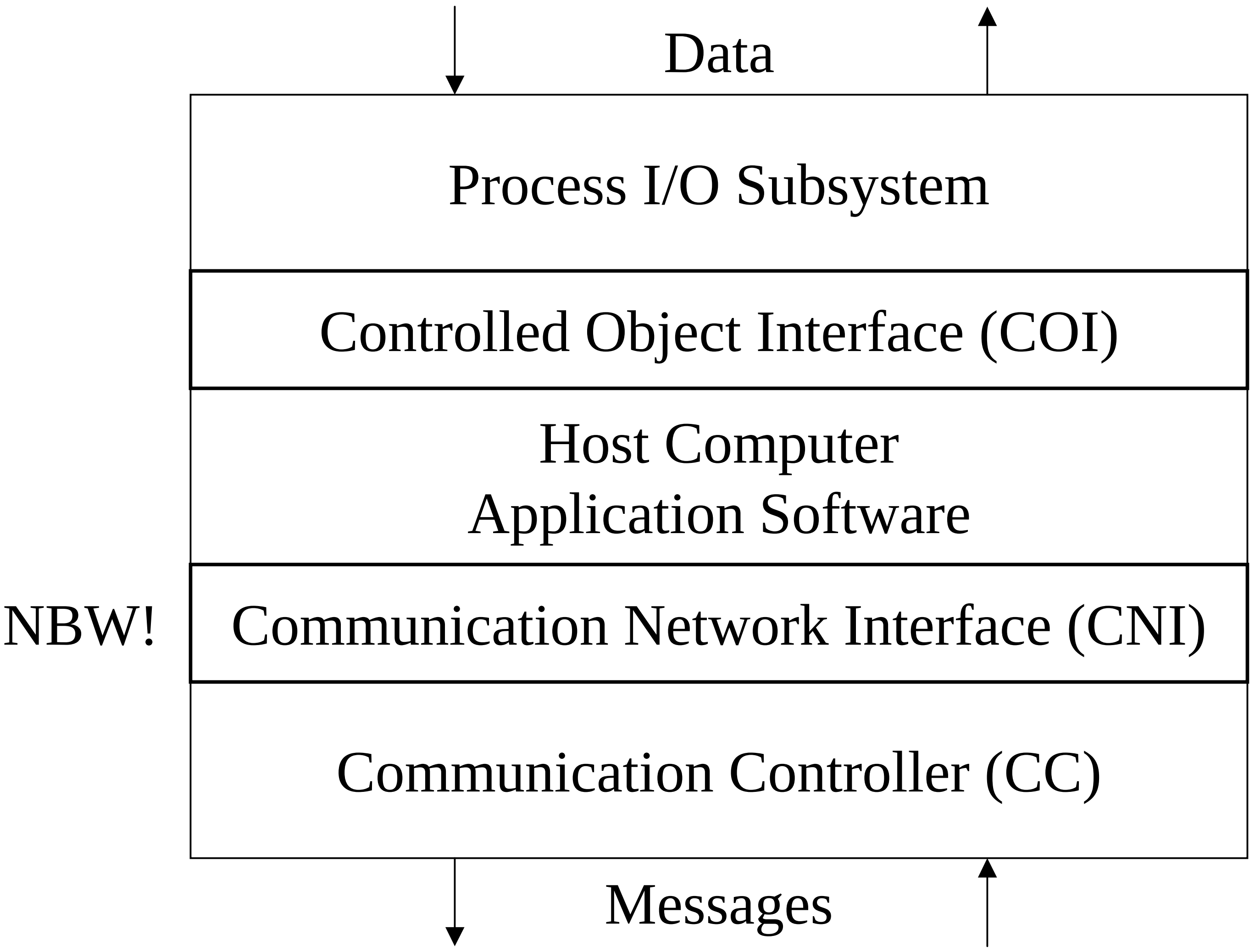
Physical Structure: A Bus



Physical Structure: A Star



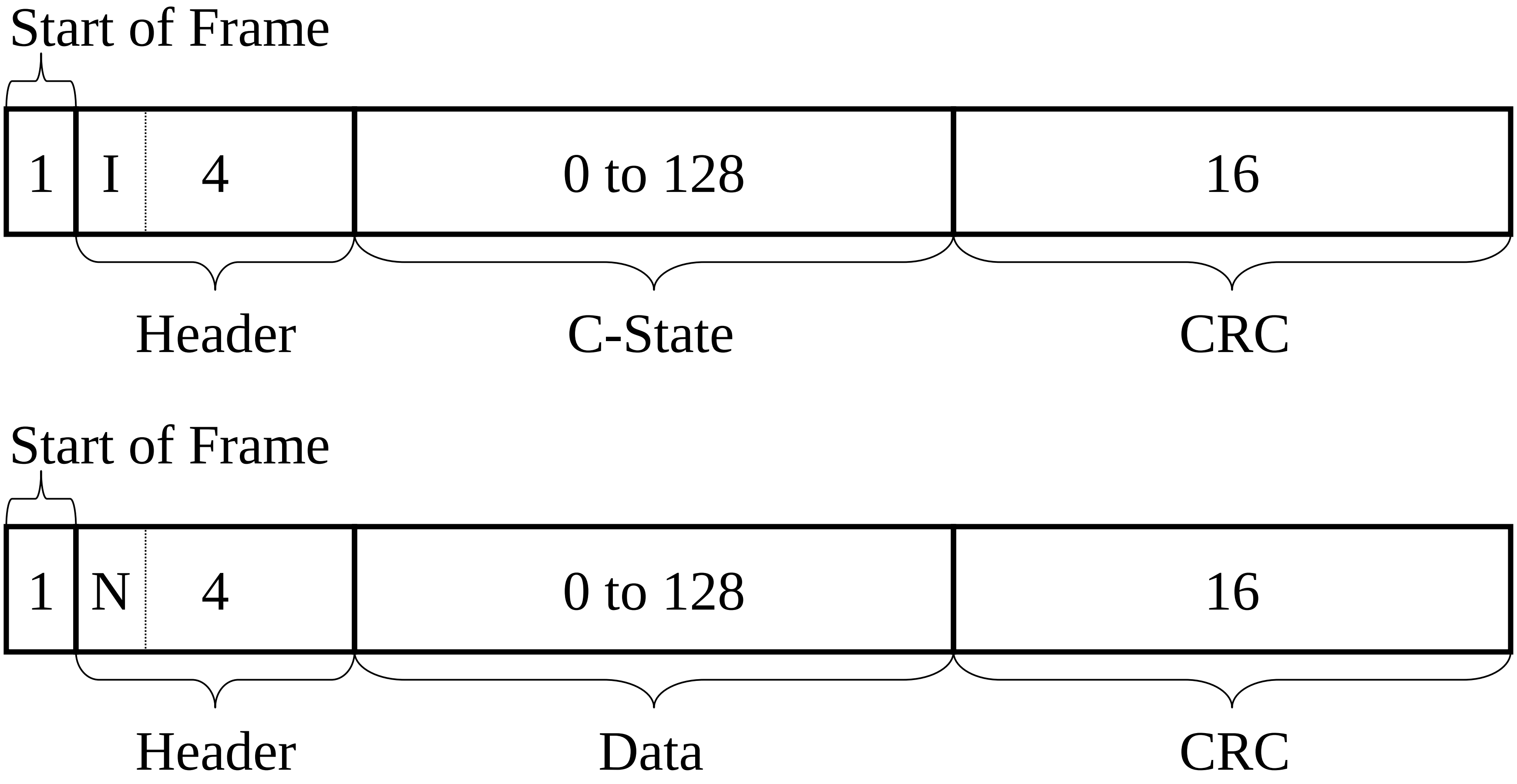
A Node: Event vs. Time-Triggered


Data
Process I/O Subsystem
Controlled Object Interface (COI)
Host Computer
Application Software
NBW!
Communication Network Interface (CNI)
Communication Controller (CC)
Messages

The Time Table (TTP: MEDL)

	Time	Address (CNI)	D	L	I	A
0						
1						
2						
3						

TTP: I- and N-Frames



Start of Frame

1 | I | 4 | 0 to 128 | 16

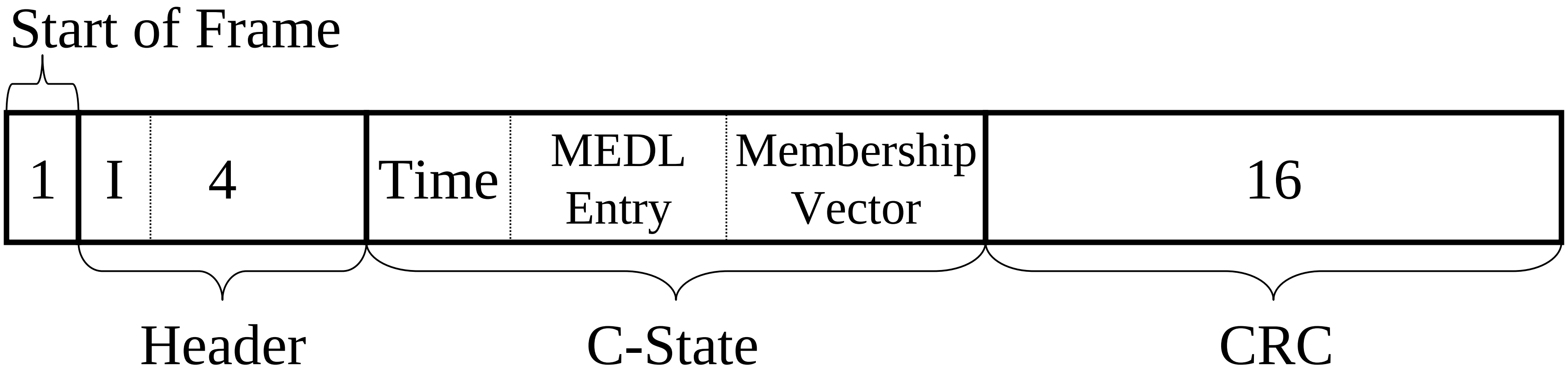
Header | C-State | CRC

Start of Frame

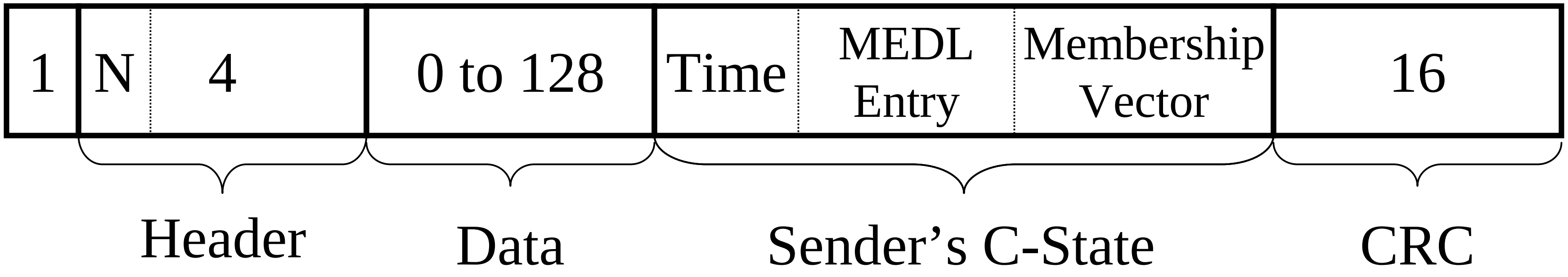
1 | N | 4 | 0 to 128 | 16

Header | Data | CRC

TTP: C-State in I-Frames



TTP: Sender’s C-State in N-Frames



1	N	4	0 to 128	Time	MEDL Entry	Membership Vector	16

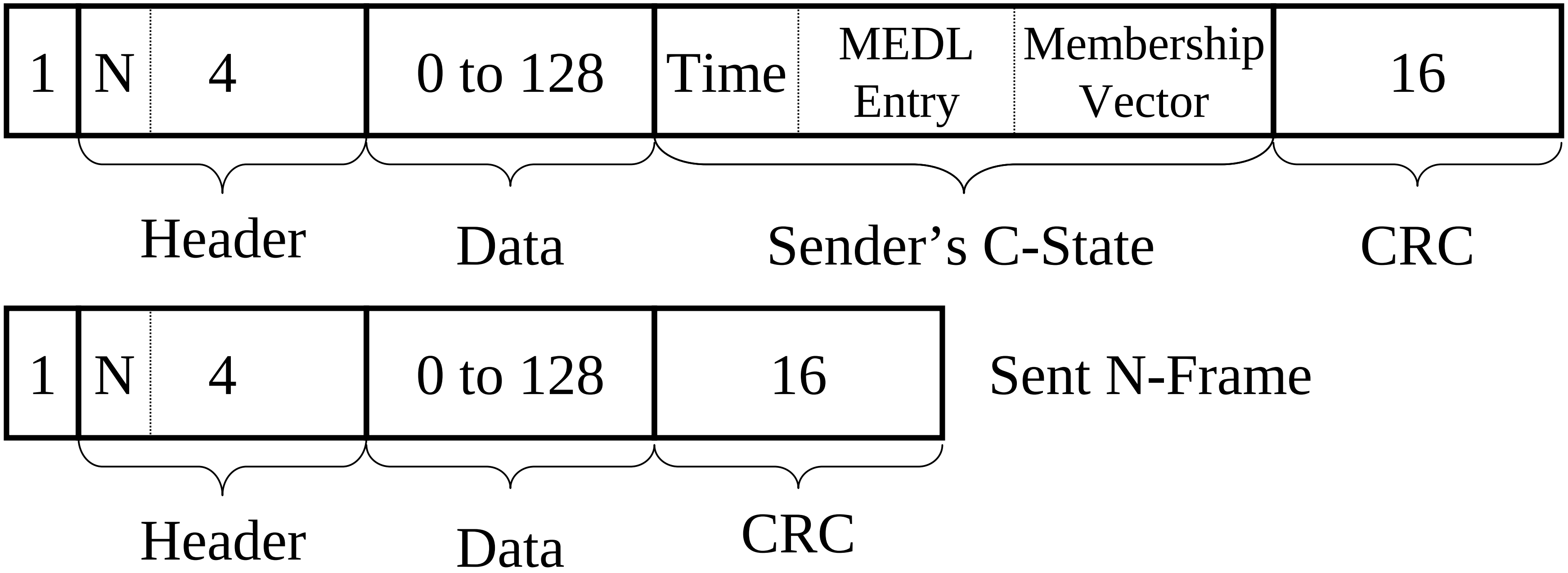
Header

Data

Sender’s C-State

CRC

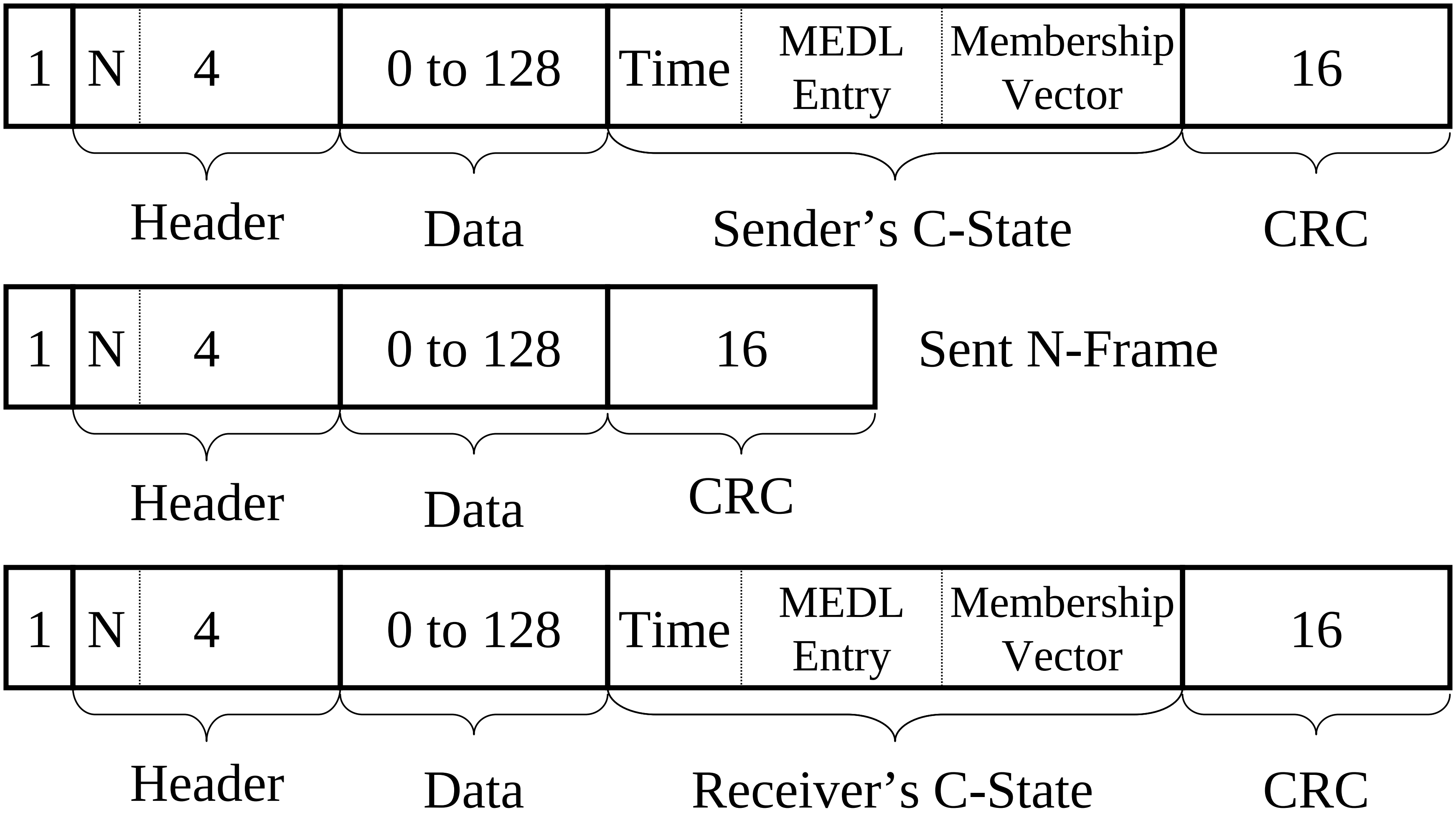
TTP: C-State in CRC of N-Frames



1	N	4	0 to 128	Time	MEDL Entry	Membership Vector	16
	Header		Data	Sender’s C-State			CRC

1	N	4	0 to 128	16	Sent N-Frame
	Header		Data	CRC	

TTP: Receiver’s C-State in N-Frames



1	N	4	0 to 128	Time	MEDL Entry	Membership Vector	16
	Header		Data	Sender’s C-State			CRC

1	N	4	0 to 128	16
	Header		Data	CRC

Sent N-Frame

1	N	4	0 to 128	Time	MEDL Entry	Membership Vector	16
	Header		Data	Receiver’s C-State			CRC