

# Lecture 5: Intro to Parallel Programming

CMSE 822: Parallel Computing  
Prof. Sean M. Couch





# Things are tough....

## Have some puppy video!



Amalie “Millie” Couch

Named for Amalie Emmy Noether





# Be sure to post PCA questions!

Response rate for PCA4 pretty low...

- Expectations: *Everyone* either POSTS a new question for UPVOTES someone else's question.



# Today

## Intro to programming in parallel

- More time to work on Homework 3!
- No programming needed for HW3
- Goal is to answer all exercises (if not complete the write-ups) by end of class!



# PCA Question

## “Fault tolerance”

Stephen White 9:28 PM  
PCA4: With distributed memory programming, is there any risk for message corruption? With networking protocols you have error detection like checksums, is there any equivalent here?

3 likes

Yes! See <https://www.christian-engelmann.info/publications/fiala12detection2.pdf> and <https://sites.google.com/site/ftxsworkshop/>

Generally hope for hardware solutions to this



# PCA Question

## MPI Dynamics Process Management



**Matthew Zeilbeck** 12:04 AM

PCA4: In the book, it says that in the original MPI, there is no way to create new processes and have them be part of the parallel run. Does this mean that it just couldn't create new processes dynamically or on-the-fly? Otherwise if it couldn't do multiple processes at all, even at compile time or whatever the correct terminology is, then MPI would be useless. (edited)

Yes, at runtime. For MPI-2 and newer, dynamic process management is implemented: <https://wgropp.cs.illinois.edu/bib/papers/pdata/1995/sanantonio.pdf>



# PCA Question

## Semaphores and race conditions



Nathan Haut 7:13 PM

PCA 4 Question. The book discusses the use of semaphores as a method for preventing multiple threads from changing a shared variable at the same time. Is it possible that a race condition for the semaphore can occur, therefore defeating the function of the semaphore. For example:

semaphore = 1

thread 1 reads semaphore, sees = 1, so initiates request to change semaphore to 0

thread 2 reads semaphore before it gets changed by thread 1, sees semaphore = 0

thread 1 changes semaphore to 0

thread 2 sets semaphore to 0

They both believe they have independent access to the critical section of code. (edited)



If implemented correctly, no. Hardware should enforce *atomic* update of semaphore (only one thread at a time).



# PCA Question

## Fortran Array-of-structures

Diane Wang 5:09 PM  
PCA4: Could Array-Of-Structures (AOS) and Structure-Of-Arrays (SOA) be used in FORTRAN? If so, how?

1

Yes! Fairly easy with object-orientation in Modern Fortran. Use user-defined `types` to do this. See, e.g., <https://stackoverflow.com/questions/38461099/how-to-implement-structures-of-arrays-instead-of-arrays-of-structures-in-fortran>

and <https://software.intel.com/content/www/us/en/develop/documentation/fortran-compiler-developer-guide-and-reference/top/language-reference/a-to-z-reference/t-to-z/type-statement-derived-types.html>



# PCA Question

## Deadlocks and serialization

$$\begin{cases} y_i \leftarrow y_i + x_{i-1} & i > 0 \\ y_i \text{ unchanged} & i = 0 \end{cases}$$

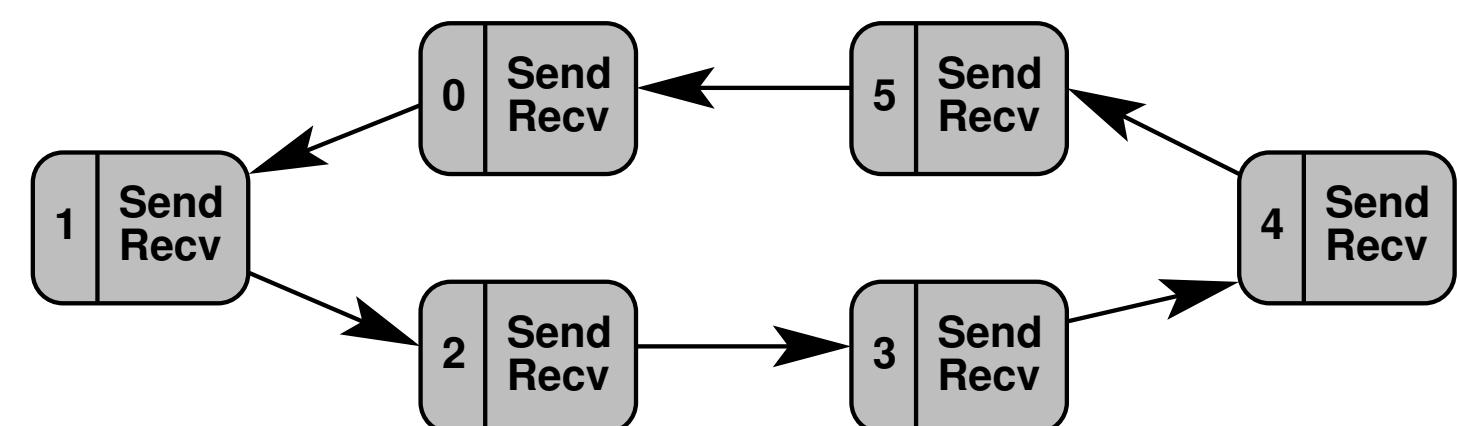
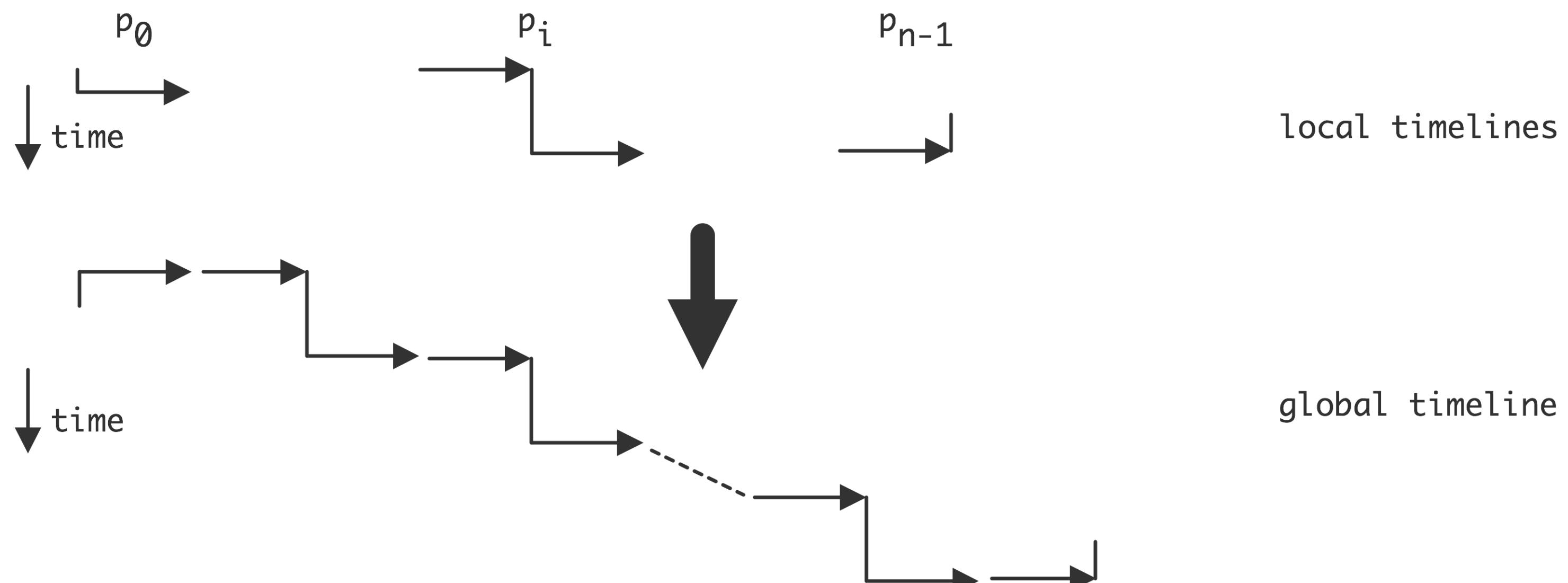


Figure 2.11: Local and resulting global view of an algorithm for sending data to the right



# PCA Question

## Deadlocks and serialization

$$\begin{cases} y_i \leftarrow y_i + x_{i-1} & i > 0 \\ y_i \text{ unchanged} & i = 0 \end{cases}$$

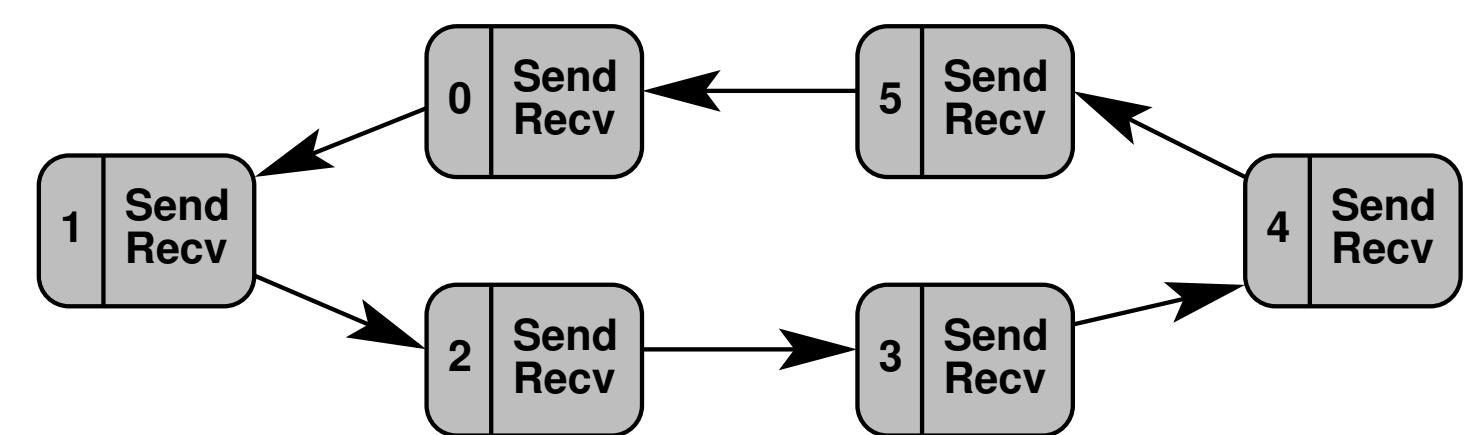
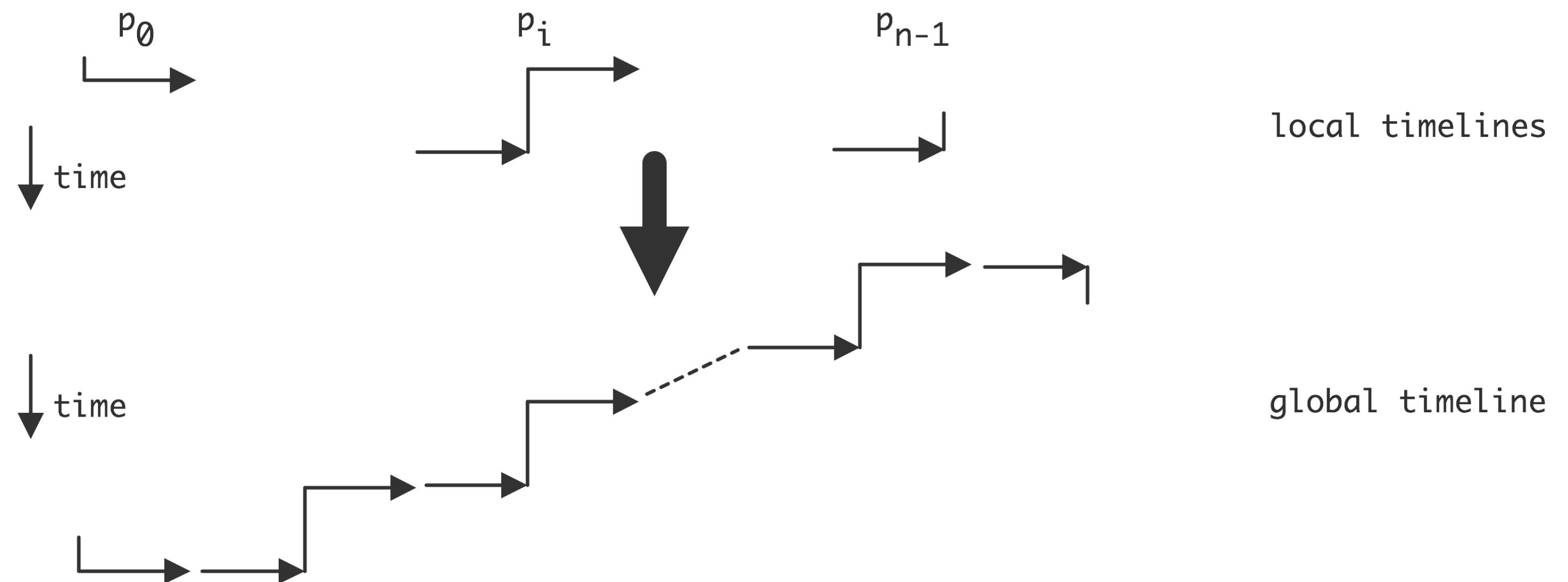


Figure 2.12: Local and resulting global view of an algorithm for sending data to the right



# PCA Question

## Deadlocks and serialization

$$\begin{cases} y_i \leftarrow y_i + x_{i-1} & i > 0 \\ y_i \text{ unchanged} & i = 0 \end{cases}$$

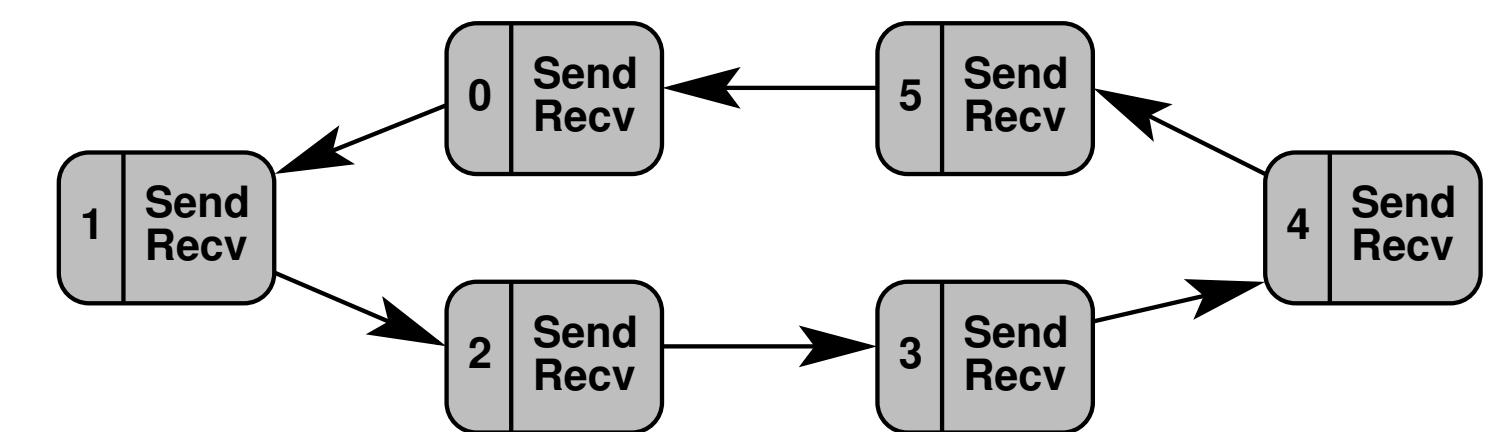
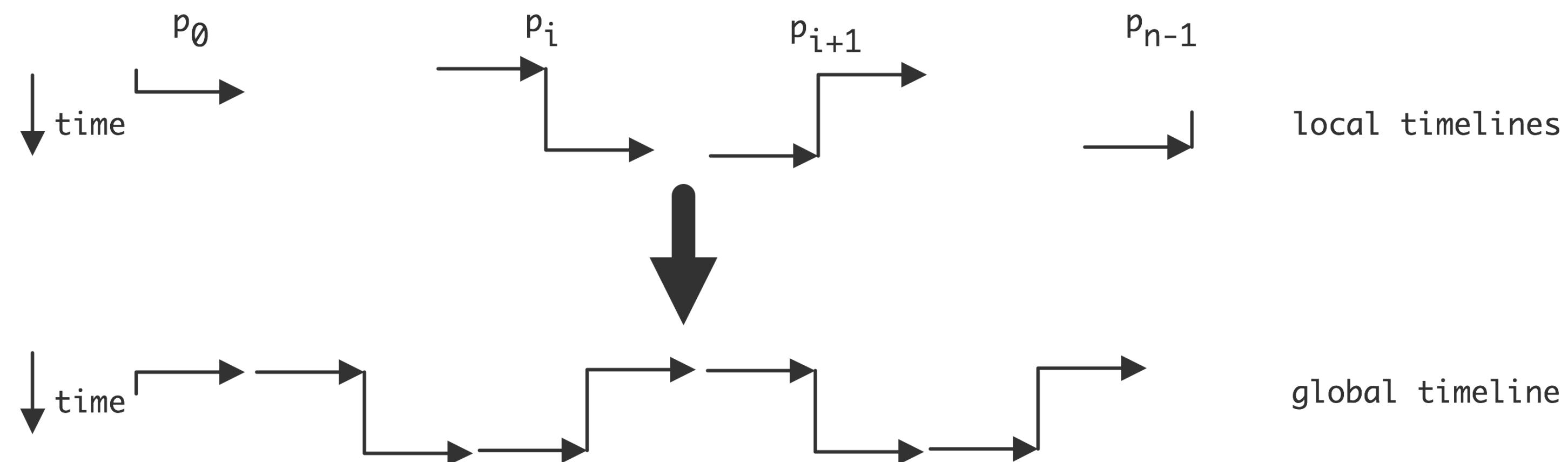


Figure 2.13: Local and resulting global view of an algorithm for sending data to the right

# **Group work on HW3**