Синхронизация локальных часов

Курносов Михаил Георгиевич

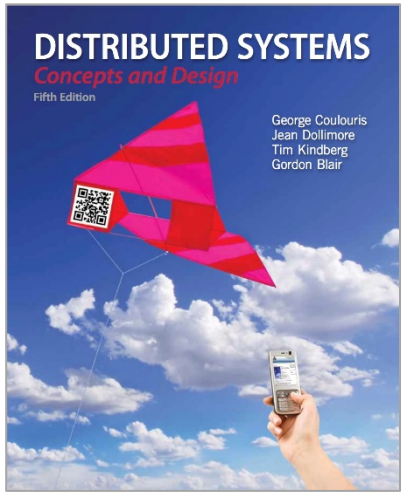
E-mail: mkurnosov@gmail.com
WWW: www.mkurnosov.net

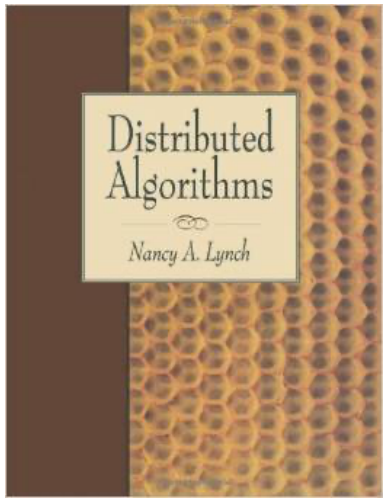
Курс «Распределенная обработка информации»
Сибирский государственный университет телекоммуникаций и информатики
Осенний семестр

Содержание

- Синхронизация локальных часов
- Логические часы
- Глобальное состояние
- Взаимное исключение
- Выборы
- Консенсус
- Репликации и согласованность

Литература





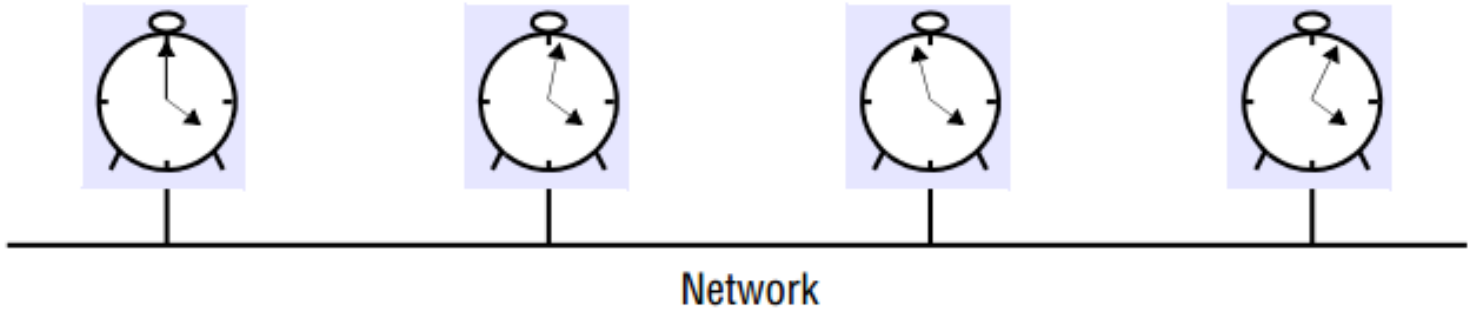
- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. **Distributed Systems: Concepts and Design** (5th Edition). – Addison-Wesley, 2011. – 1008 p.
- Жерар Тель. **Введение в распределенные алгоритмы**. - М.: МЦНМО, 2009.
- Эндрю Таненбаум, М. ван Стеен. **Распределенные системы. Принципы и парадигмы**. - СПб.: Питер, 2003.
- Nancy A. Lynch. **Distributed Algorithms**. – Morgan Kaufmann, 1996. – 904 p.

Модель распределенной системы

- Распределенная система состоит из n процессов $p_i, \quad i \in \{1,2, \dots, n\}$
- Каждый процесс однопоточный и выполняется на своем процессоре (узле)
- Процесс характеризуется своим состоянием – совокупность значений всех переменных процесса
- Процессы взаимодействуют путем передачи сообщений (send, recv)
  - ❑ Сообщения передаются не мгновенно (латентность, пропуская способность)
- В системе могут происходить как аппаратные (узлы, каналы связи), так и программные отказы
- Все процессы выполняют шаги *распределенного алгоритма* (distributed algorithm)

Локальные часы

- Каждый процесс распределенной системы имеет доступ к локальным часам своего вычислительного узла
- Процессы могут ассоциировать со своими событиями *временные метки* (timestamps)
- Показания локальных часов процессов могут быть существенно различными!



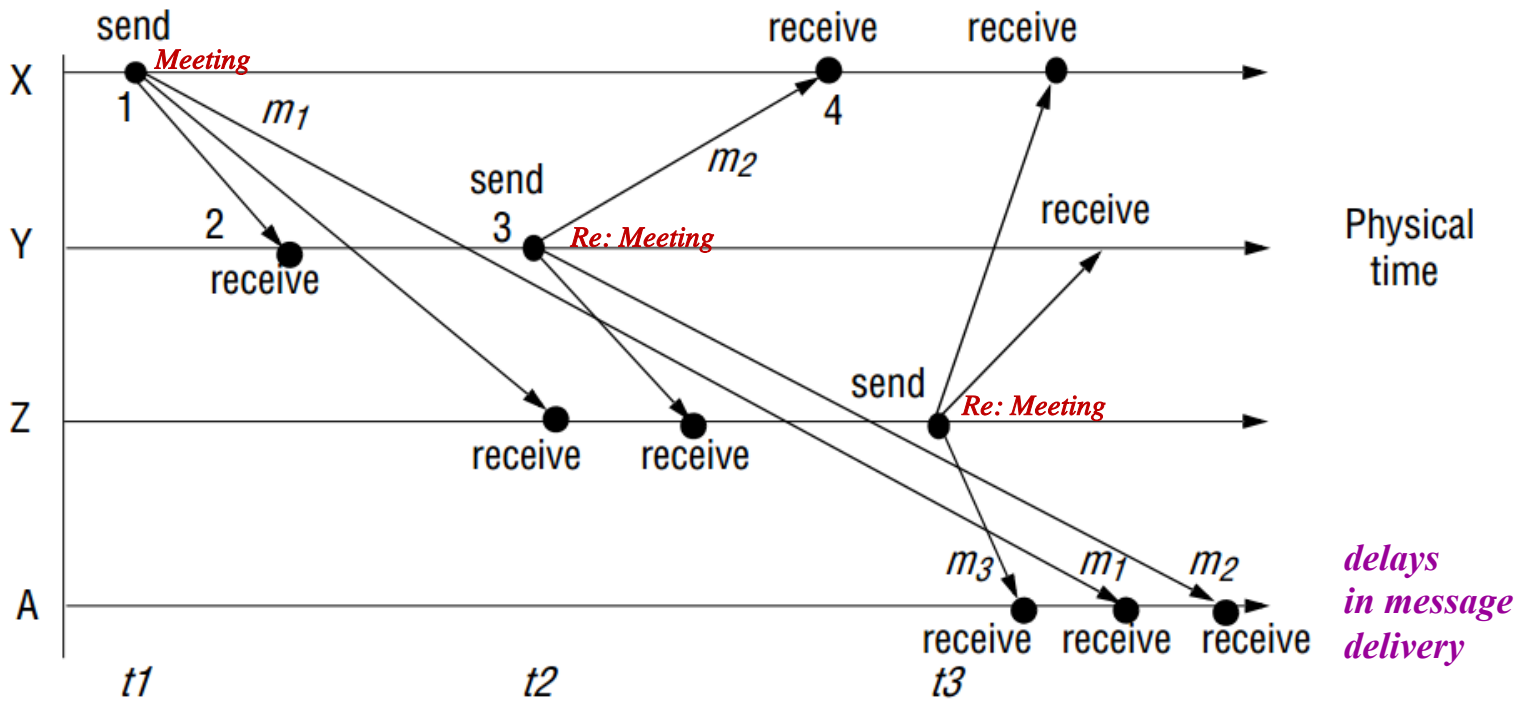
Модель взаимодействий

- **Синхронные распределенные системы** (synchronous distributed system)
  - Известны нижняя и верхняя границы времени выполнения шагов алгоритма
  - Передача и прием сообщения выполняется за известное конечное время
  - Скорость смещения показаний локальных часов (clock drift rate) ограничена и известна

  В таких системах можно использовать механизмы таймаутов для обнаружения отказов, оценивать границы времени выполнения алгоритма и др.

- **Асинхронные распределенные системы** (asynchronous distributed system)
  - Процессы и их отдельные шаги могут выполняться произвольное время
  - Передача и прием сообщений могут занимать произвольное время (заранее неизвестное)
  - Произвольное смещение показаний локальных часов процессов

Порядок событий (E-mail users X, Y, Z, A)



Inbox

Re: Meeting from Y
Re: Meeting from Z

Meeting from X
Re: Meeting from Z

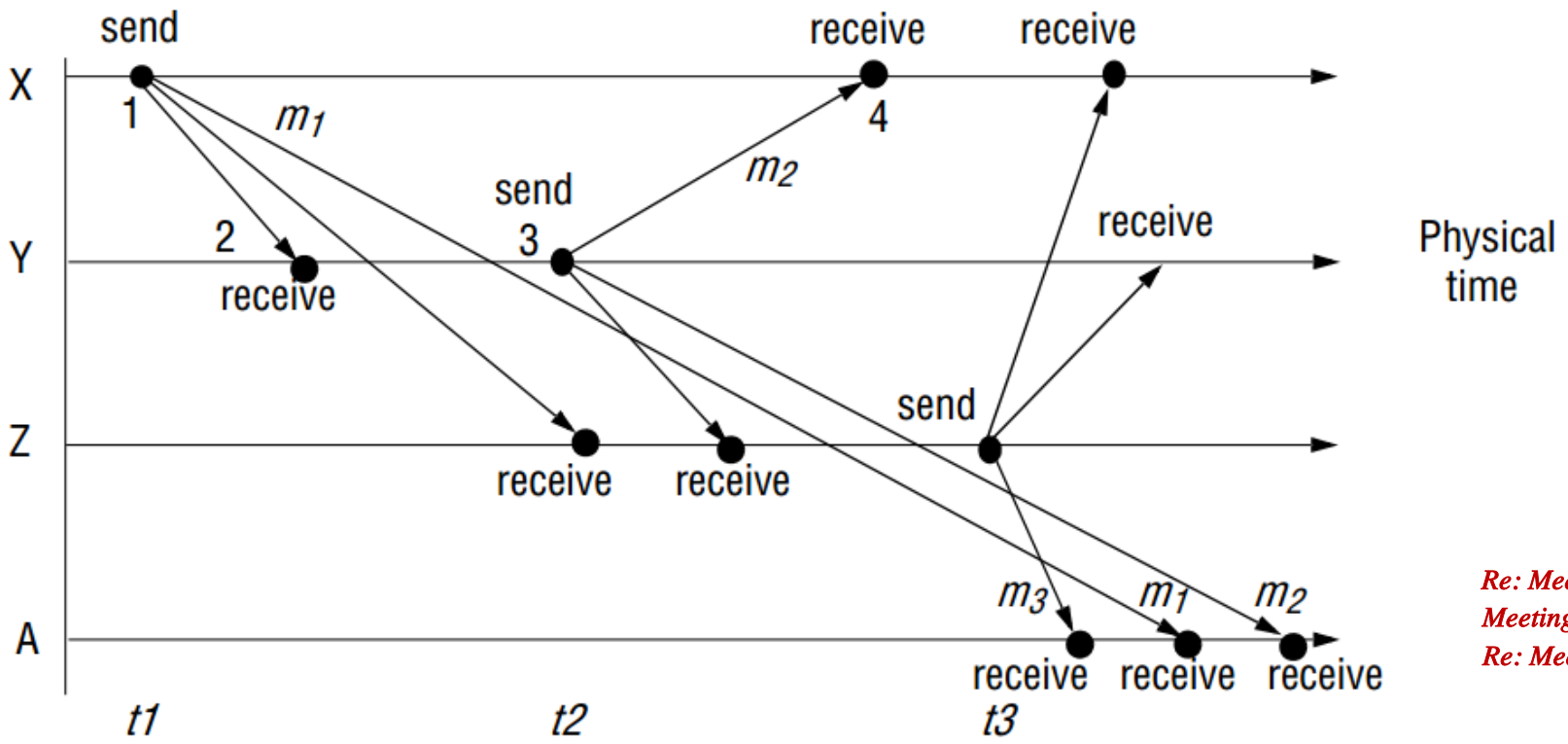
Meeting from X
Re: Meeting from Y

Re: Meeting from Z
Meeting from X
Re: Meeting from Y

- Некоторые процессы (A) получили сообщения в некорректном порядке
- Из-за задержек в сети сообщение m1 и m2 доставлены в A после m3

George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. **Distributed Systems: Concepts and Design** (5th Edition). – Addison-Wesley, 2011. – 1008 p.

Порядок событий (E-mail users X, Y, Z, A)



Sort by t

Re: Meeting from Z, t3
Meeting from X, t1
Re: Meeting from Y, t2

→

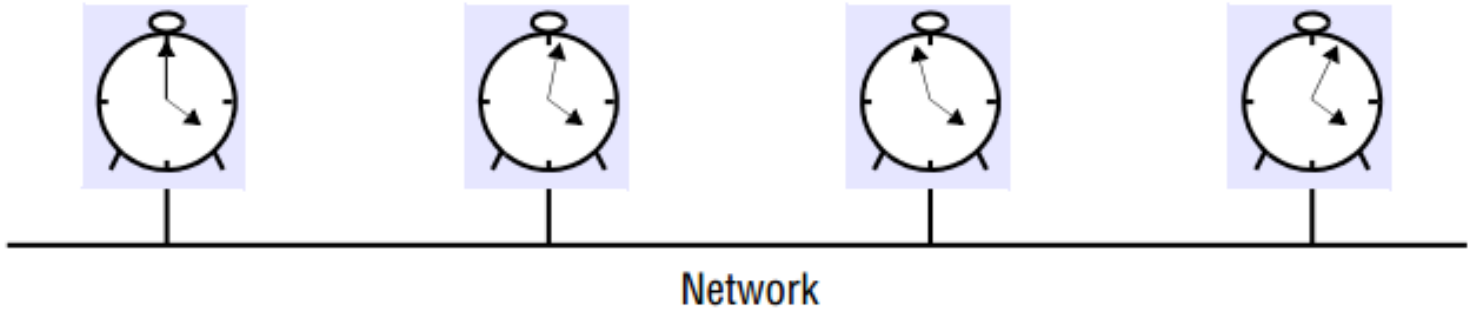
Meeting from X, t1
Re: Meeting from Y, t2
Re: Meeting from Z, t3

- Если локальные часы процессов синхронизированы, то каждое сообщение можно снабдить временной меткой: t1 < t2 < t3
- Принятые сообщения в A можно переупорядочить по временным меткам (восстановить исходный порядок сообщений)

Глобальные часы (global clock)

- **В какое время наступило определенное событие?**
  - ❑ Локальные часы процессов надо синхронизировать с надежным внешним источником времени
- **Приложения**
  - ❑ Обработка транзакций
  - ❑ Логирование и аудит
  - ❑ Профилировщики MPI-программ (Intel Trace Analyzer & Collector, MPE2, …)

Часы



- Показания локальных часов процессов могут быть различными
- Разность показаний локальных часов двух процессов называется *смещением* (clock skew)
- Локальные часы процессов могут иметь различные источники времени (RTC, TSC, HPET, …) и характеризуются разной точностью (clock drift)

```
$ cat /sys/devices/system/clocksource/clocksource0/available_clocksource
tsc hpet acpi_pm
```

Синхронизация показаний локальных часов

- **Внешняя синхронизация (External synchronization)**
- Локальные часы C_i процессов синхронизированы с внешним источником S с точностью D, если

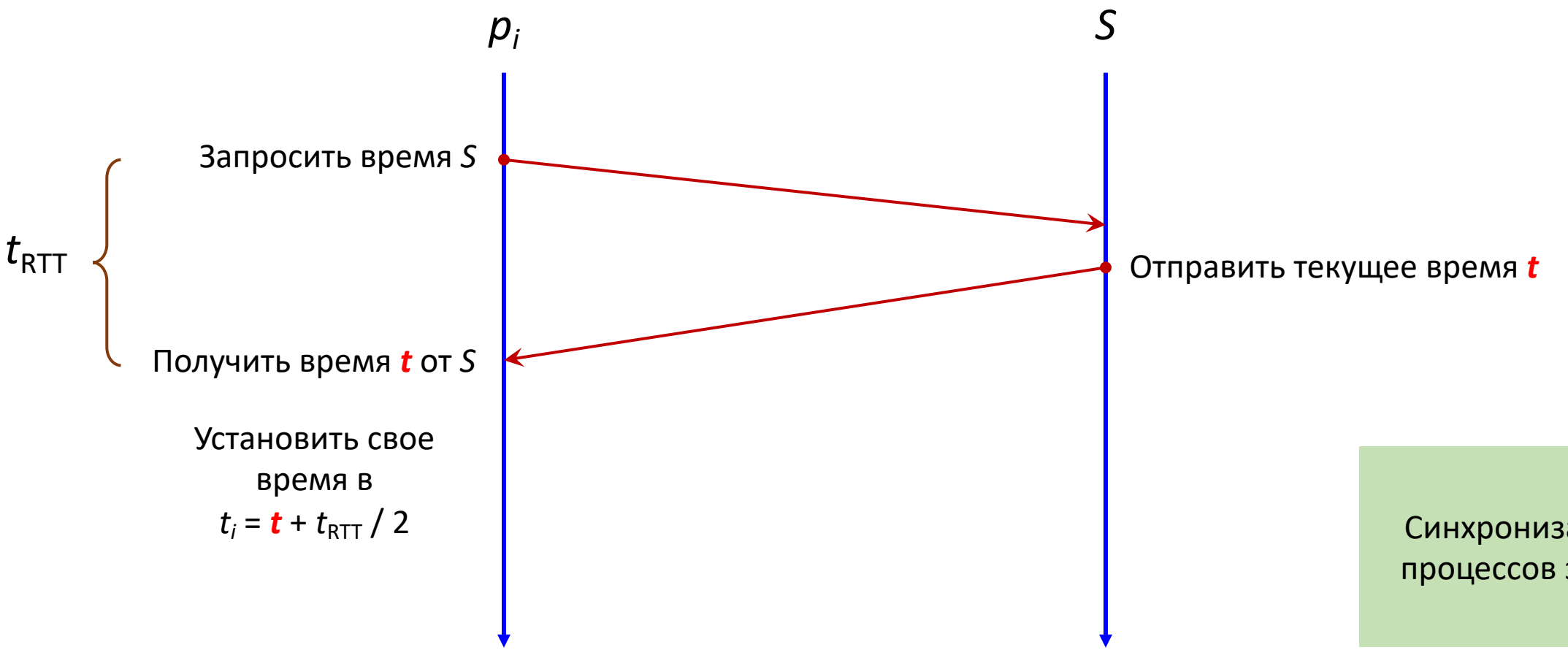
$$|S(t) - C_i(t)| < D, \qquad \forall t, \qquad i = 1,2, \dots, n$$

- **Внутренняя синхронизация (Internal synchronization)**
- Локальные часы процессов синхронизированы между собой с точностью D, если

$$\left|C_i(t) - C_j(t)\right| < D, \qquad \forall t, \qquad i, j = 1,2, \dots, n$$

Синхронизация показаний локальных часов

- **Алгоритм внешней синхронизации Кристиана**
- Cristian, F. (1989), “Probabilistic clock synchronization” // Distributed Computing (Springer) 3 (3): 146–158



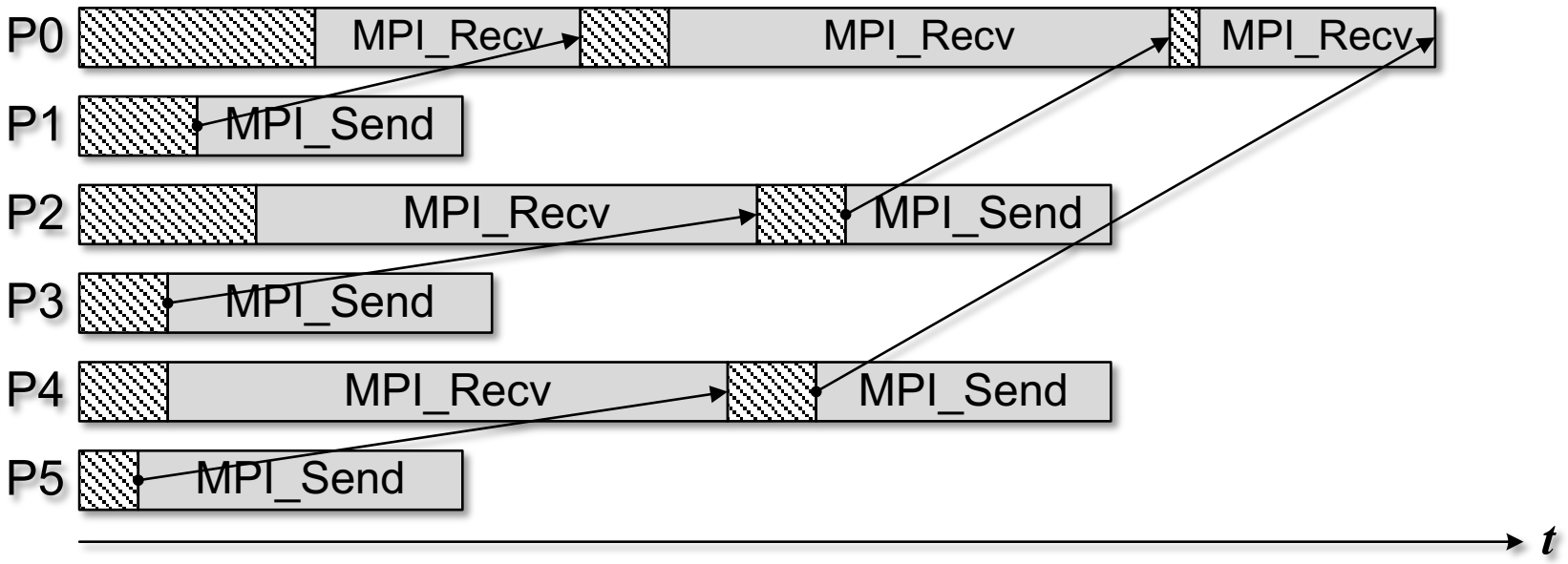
Синхронизация n процессов за $O(n)$

Синхронизация показаний локальных часов

- Алгоритм Беркли (Gusella, Zatti, 1989)
- NTP (алгоритм K. Marzullo, 1984)
- Алгоритмы синхронизации часов в MPI-профилировщике MPE2 (mpe2/src/logging/src/clog_sync.c):
  - Sequential $O(n)$
  - Binomial tree $O(\log n)$
  - Ring algorithm $O(1)$

Пример: профилирование MPI-программ

- В MPI-процессах происходят события (MPI_Send, MPI_Recv, MPI_Bcast, …)
- Профилировщик для каждого процесса ведет журнал событий: (timestamp, event)
- Как построить пространственно-временную диаграмму выполнения процессов (timeline) имея *n* журналов событий?

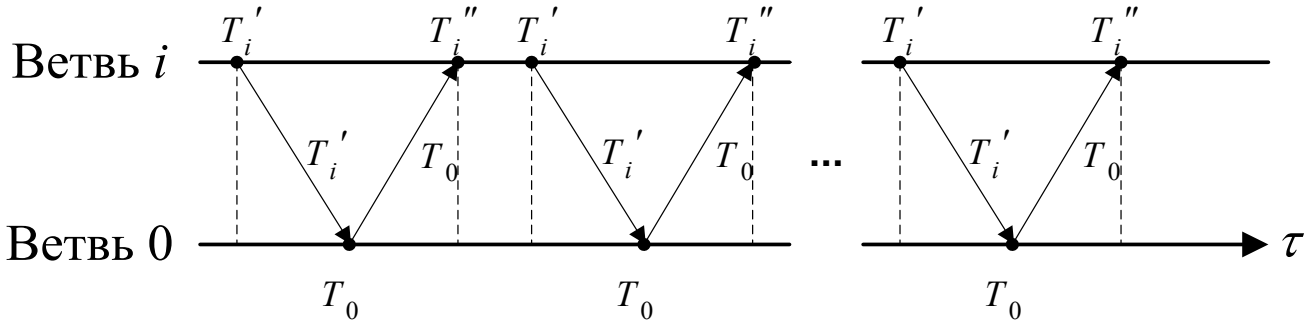


Пример: профилирование MPI-программ

- Каждая ветвь ***i*** вычисляет смещение o_i показаний своих часов относительно часов ветви 0
- Зная o_i и показания T_i своих часов ветвь ***i*** может вычислить показания T_0 глобальных часов

$$T_0 = T_i + o_i \qquad T_i = T_0 - o_i$$

- Вычисление o_i



$$T_{\text{RTT}} = T_i'' - T_i' \qquad o_i = T_0 - \frac{T_{\text{RTT}}}{2} - T_i'$$

- Синхронизируем показания локальных часов (формируем глобальные часы)
- Каждое событие в журнале процесса *i* снабжается временной меткой: ($T_i + o_i$, *event*)

Пример: измерение времени выполнения коллективных MPI-операций

- Требуется измерить время выполнения коллективной операции (MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Reduce, MPI_Allreduce, MPI_Barrier, …)

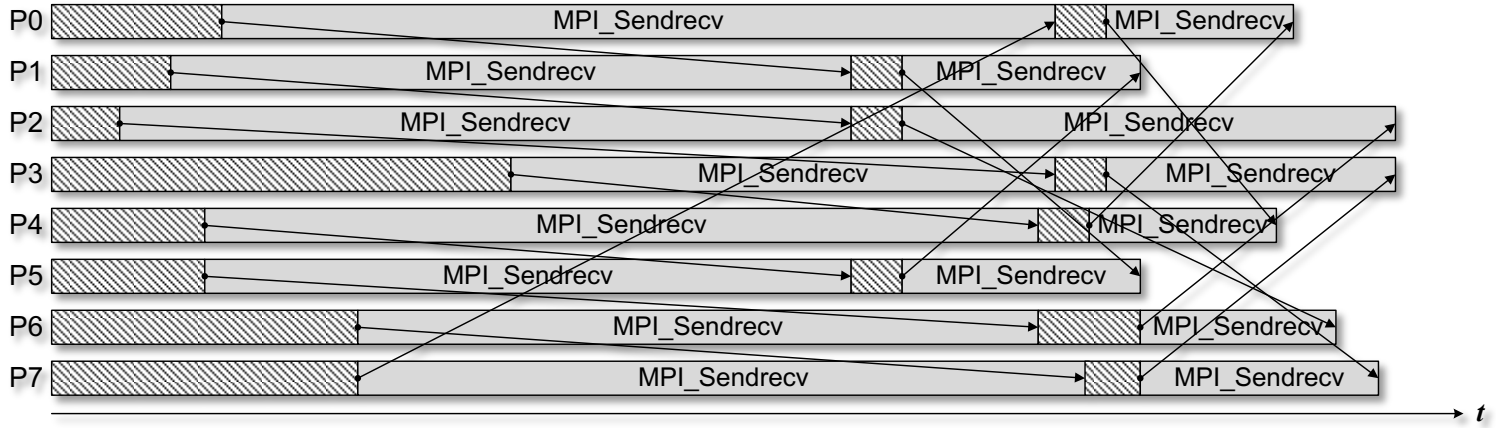


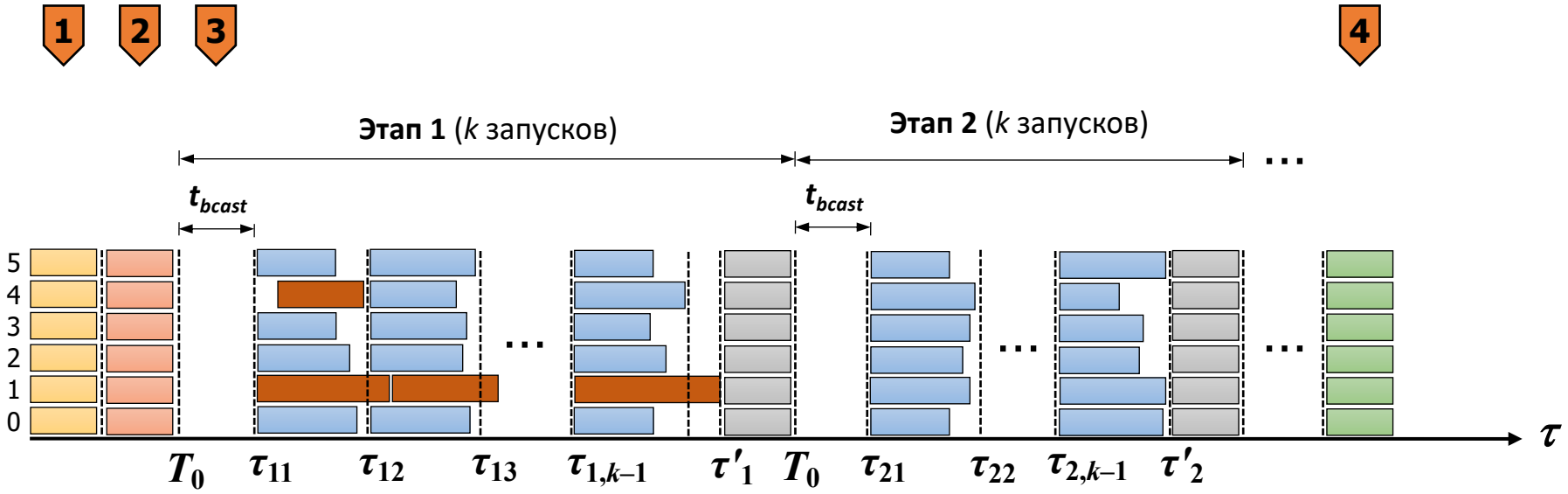
Диаграмма выполнения барьерной синхронизации “рассеивающим” алгоритмом (Dissemination barrier, MPICH, Open MPI)

- Ветви начинают выполнение операции в разные моменты времени (load imbalance, предварительная синхронизация функцией MPI_Barrier)
- Измеряется среднее время выполнения операции по результатам *k* запусков (ошибка измерений)

```
MPI_Bcast(buf, count, MPI_BYTE, root, comm) /* Init */
MPI_Barrier(comm)                           /* Sync */
t = MPI_Wtime()
for i = 1 to k do
    MPI_Bcast(buf, count, MPI_BYTE, root, comm)
end for
t = (MPI_Wtime() – t) / k
```


- T. Hoefler, T. Schneider and A. Lumsdaine. 6] T. Hoefler, T. Schneider and A. Lumsdaine. **Accurately Measuring Overhead, Communication Time and Progression of Blocking and Nonblocking Collective Operations at Massive Scale** // http://unixer.de/publications/img/hoefler-collmea.pdf
- Курносов М.Г. **MPIPerf: пакет оценки эффективности коммуникационных функций стандарта MPI** // ПаВТ-2012, http://www.mkurnosov.net/uploads/Main/kurnosov-pavt-2012.pdf

Пример: измерение времени выполнения коллективных MPI-операций



- SKaMPI
- NetGauge
- MPIPerf

1 Синхронизация показаний локальных часов ветвей – установление глобального времени

2 Оценка времени выполнения трансляционного обмена:

```
MPI_Bcast(&buf, 1, MPI_DOUBLE, mpiperf_master_rank, comm)
```

3 **Формирование расписания запусков операции по глобальным часам**

Измерение времени выполнения операции. Корректировка расписания. Проверка условий окончания измерений

4 Статистическая обработка результатов измерений (выбросы, доверительные интервалы). Формирование отчета