SZÉCHENYI EGYETEM
UNIVERSITY OF GYŐR

Doctoral School of Multidisciplinary Engineering Sciences (MMTDI)

Performance Analysis of ICT Systems

Lecture 2

Novel Results in Parallel DES and Related Areas

https://www.tilb.sze.hu/cgi-bin/tilb.cgi?0=m&1=targyak&2=NGD_MDA64_1

Dr. Gábor Lencse
Professor
Dept. of Telecommunications, University of Győr
lencse@sze.hu

Agenda

- Reminder: the operation of the event-driven discrete-even simulation
- Parallel Discrete-Event Simulation (PDES)
- Traffic-Flow Analysis (TFA)
- Combination and interworking of TFA and DES
- Further solutions for accelerating the performance analysis of ICT systems

Recall: Operation of the event-driven DES

- Future Event Set (FES)
  - Stores the events to occur in the future

Timestamp	Event
0.000000	Transmission of frame #0 begins
0.000100	The head of frame #0 arrives to the receiver
0.000512	Transmission of frame #0 ends
0.000612	The tail of frame #0 arrives to the receiver

Recall: Algorithm of the event-driven DES

```
Initialization: put staring event(s)
 into the FES;
REPEAT
  Remove the event with the lowest
   timestamp from the FES;
  NOW := the timestamp of the removed
   event;
  Process the event and schedule new
   events, if necessary;
UNTIL (NOW > limit)OR(no more events)OR
 (we must stop for some reason)
```

Limits of sequential DES

- In the case of a detailed model of a complex system, problems may arise from
  - The storage capacity needed (memory size)
  - The computations required (execution time)
- Possible solutions include
  - Parallelization
  - Application of another, simulation like method (TFA)
  - The combination of the two above

Parallel Discrete-Event Simulation

- The model of the system is divided into segments
- The segments are assigned to processors, which execute them
- All segments must have their own FES
  - A centralized FES would be a bottleneck
- The virtual times of the segments must be synchronized to maintain causality
- The achievable speed-up depends on
  - The partitioning of the model into segments
  - The applied synchronization method
  - The communication channels among the processors

Synchronization methods for PDES

- Conservative (causality is always ensured)
- Optimistic (causality errors are detected and a roll-back is performed)
- Statistical synchronization method (only statistics describing the message flow are exchanged)
  - It has two variants

Conservative PDES

- An event may be processed only in the case, if it is ensured that no event with a smaller timestamp may arrive (from another segment)
- There are good guarantees exist with certain classes of systems, due to the inherent properties of the model (terms: lookahead, null messages)
- It does not provide speed-up in the general case: only a single CPU can work at a time.

Optimistic PDES

- Allows causality errors to happen
  - The segments can be executed “independently”
  - *straggler*: an event with a lower timestamp than the current virtual time
- When a causality error happens, a *roll-back* is performed: it undoes everything happened since the timestamp of the straggler, including
  - Resetting all state variables (state saving is needed!)
  - Invalidating the messages* sent out (using anti-messages) and scheduled locally. But they may already be processed!
- It does not scale up well: as the number of segments grows, rollbacks become a problem

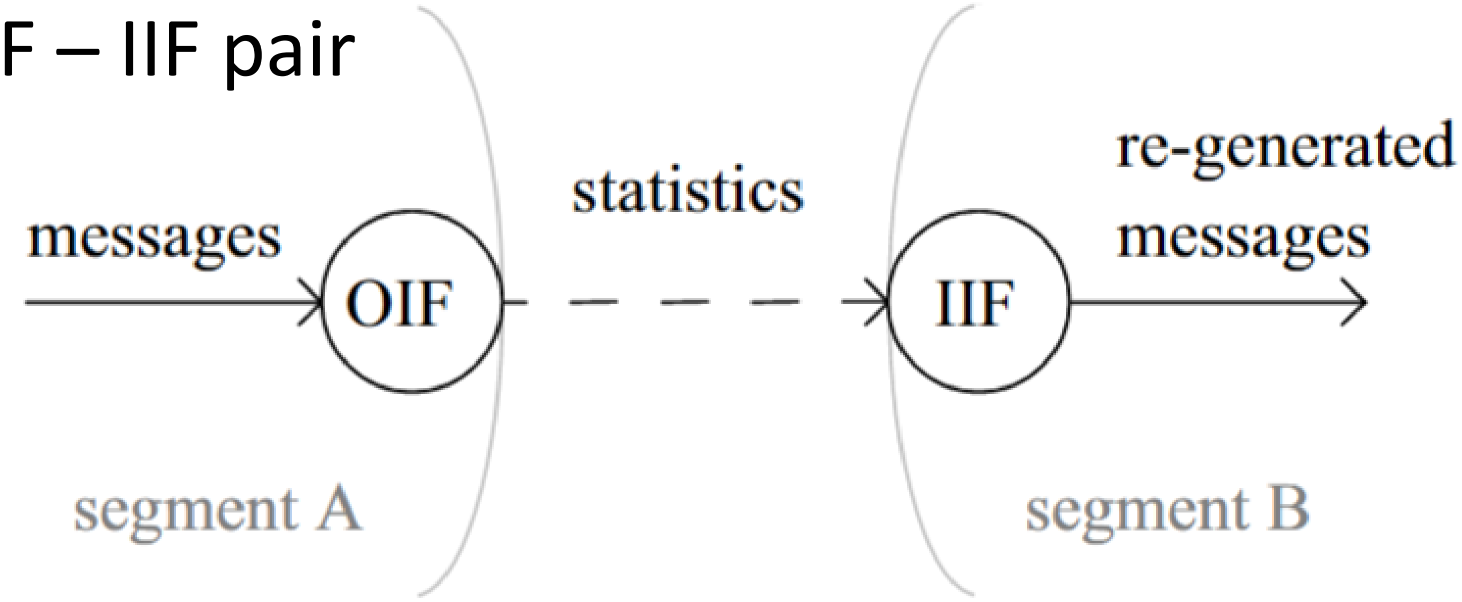
*A “message” is a synonym of an “event”.

Original Statistical Synchronization Method

- Segments are equipped with one or more input and output interfaces.
- The messages generated in a given segment and to be processed in a different segment, are not transmitted there, but the output interfaces (OIF) collect statistical data of them.
- The input interfaces (IIF) generate messages for the segment according to the statistical characteristics of the messages collected by the proper output interfaces.

Original Statistical Synchronization Method

- An OIF – IIF pair



- This method was invented by György Pongor
- It is suitable for the analysis of systems in *steady state* (opposite of transient state)

(Pongor, 1992)

The SSM-T Method

- SSM + *loose synchronization*
- Loose synchronization between segment A and segment B is defined formally as follows:
  - Let t_1, t_2, … t_i, … t_n be synchronization points of time.
  - Let t_A and t_B denote the LVTs (local virtual times) of segment A and segment B, respectively.
  - Segment A and segment B are loosely synchronized, if:

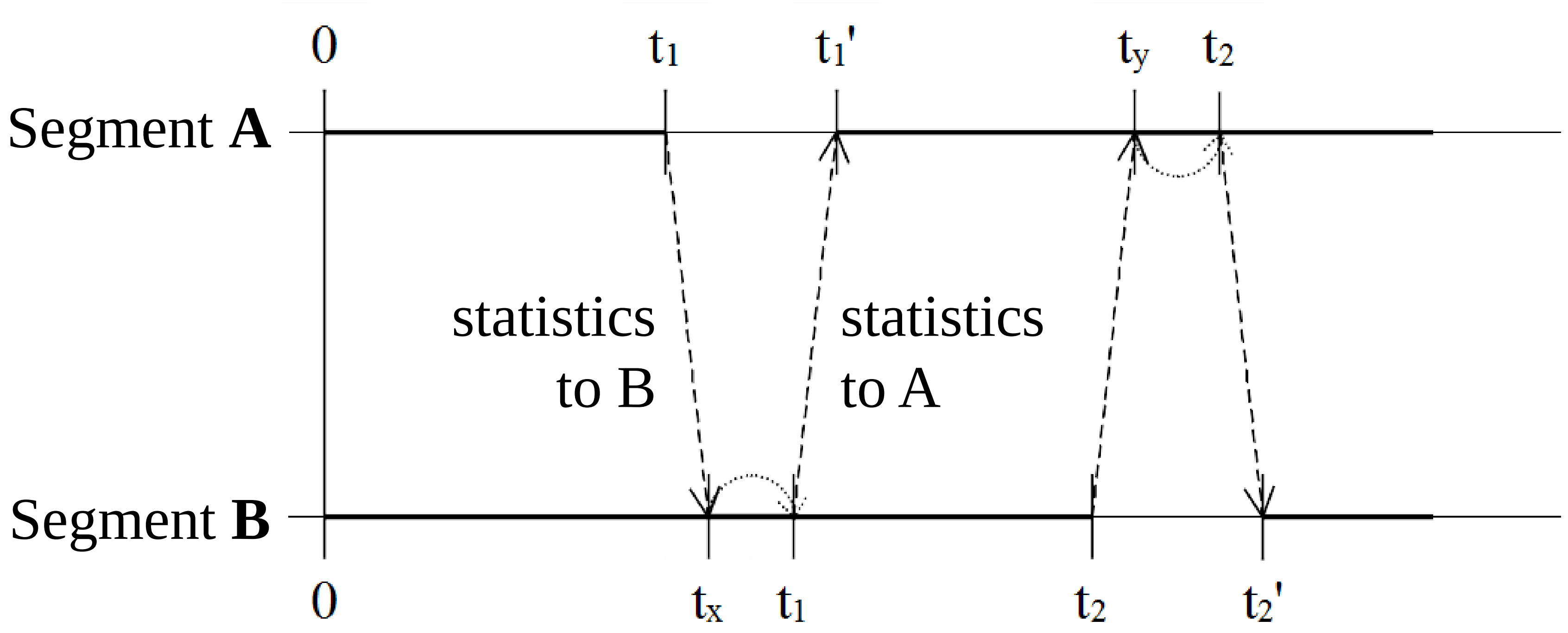
$$((t_A < t_i) \Rightarrow (t_B \le t_i)) \wedge ((t_A > t_i) \Rightarrow (t_B \ge t_i)),\ \ i = 1, 2, \ldots , n.$$

Informally, loose synchronization of two segments means that none of the segments may leave a synchronization point of time until the other segment reached it.

(Lencse, 1998a)

Illustration of the operation of SSM-T

How segment A and segment B exchange their statistics?



There is loose synchronization between segment A and segment B.

Applicability criteria of SSM-T

a) The simulated system can be divided into segments so that not the individual messages but only their statistical characteristics are important between the segments.

b) A *small error* in the approximation of the statistical characteristics of the message flow causes *small error* in the output of the simulation that *depends only on the measure of the approximation error*.

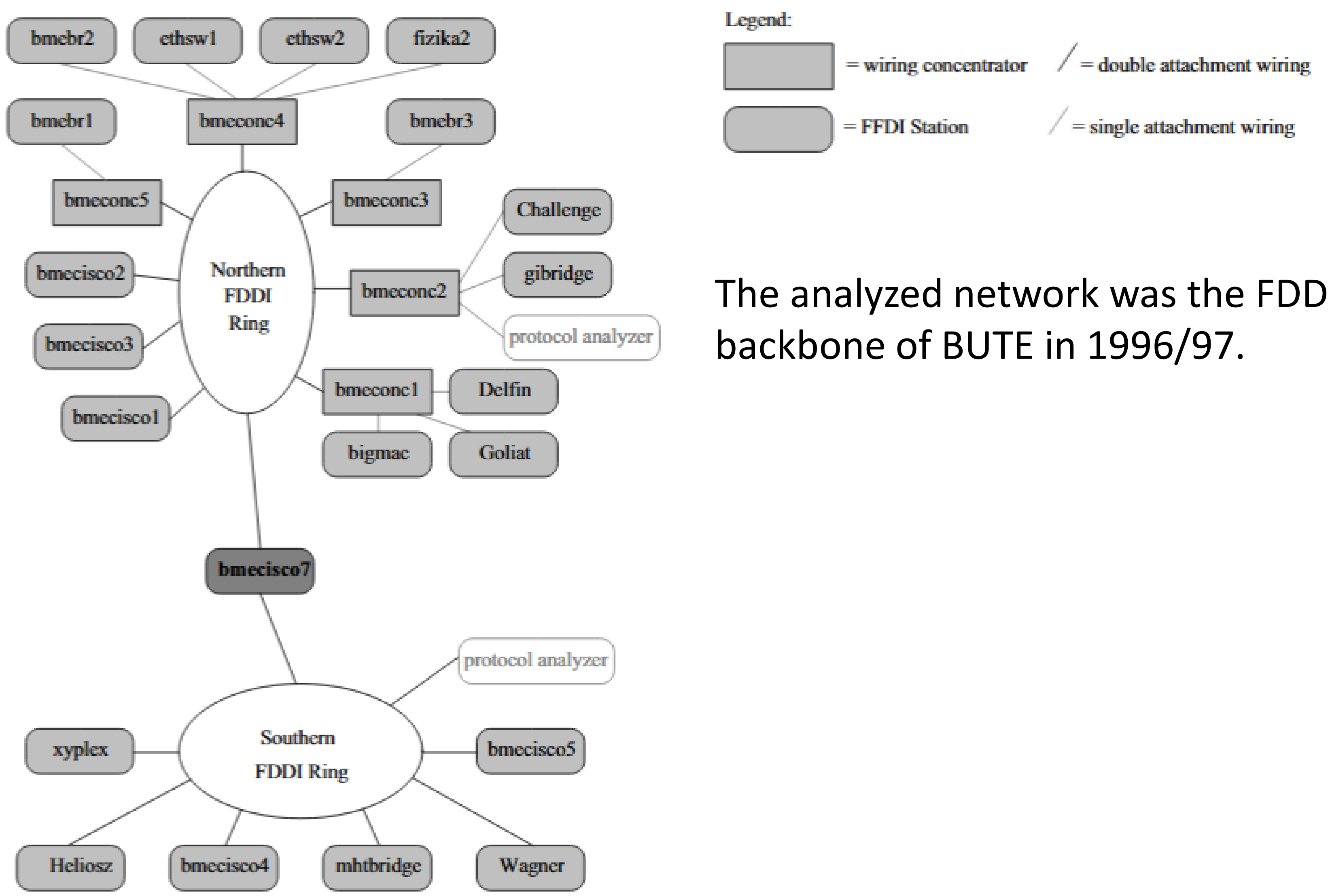
c) The parameters of the model may change during the simulation, but the changes in the statistical characteristics of the message flows between the segments are *rare enough*.

Applicability criteria of SSM-T

d) A change in the statistical characteristics of the message flow is only propagated to other segments, when the statistics package is actually sent over by the OIF. The fourth condition is that this delay causes error in the output of the simulation only during the delay and/or at most during an additional time interval with a length proportional to the delay.

Formalization and formal proof can be found in (Lencse 1999a)

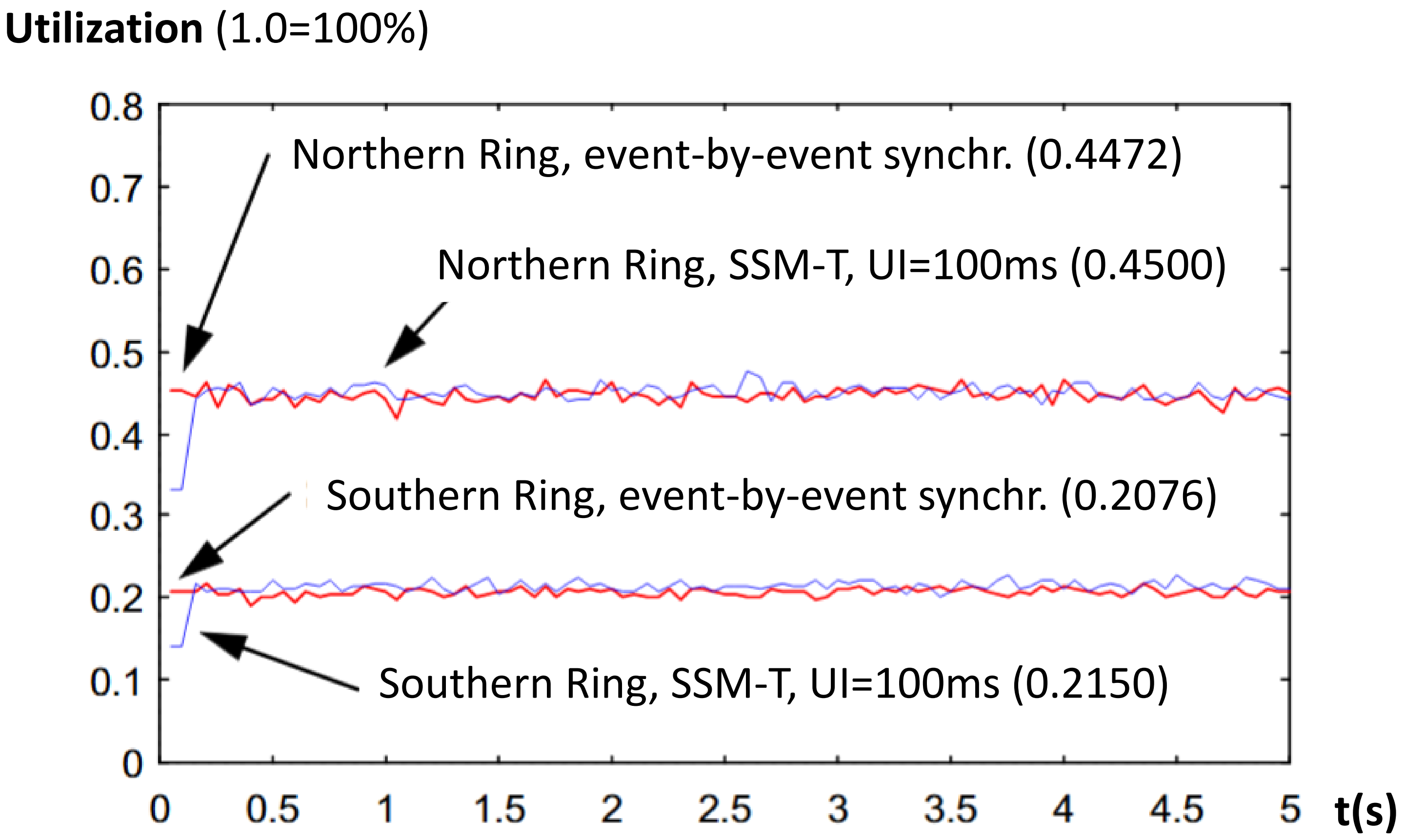
Topology of the example network for SSM-T



The analyzed network was the FDDI backbone of BUTE in 1996/97.

Results received with SSM-T

- Utilization of the FDDI rings as a function of time with and without SSM-T



Speed-up achieved with SSM-T

- A speed-up of 1.91 or 1.86 (depending of the update interval parameter of SSM-T) was achieved in the simulation of two interconnected FDDI rings by two networked computers.
- In the case, when the speed-up was 1.91, the output error was 0.63% for one ring and 3.56% for the other ring (compared to the single processor simulation).

(Lencse, 1998a)

Traffic-Flow Analysis (TFA)

- TFA shows the steady state traffic conditions of the analyzed network.
- TFA is a combination of simulation and numerical methods.
- The traffic of the applications is represented by statistics (aggregated traffic model).
- TFA may use various traffic models, which comply with the applicability criteria, currently one traffic model has been defined.

(Lencse 2001)

Operation Steps of TFA

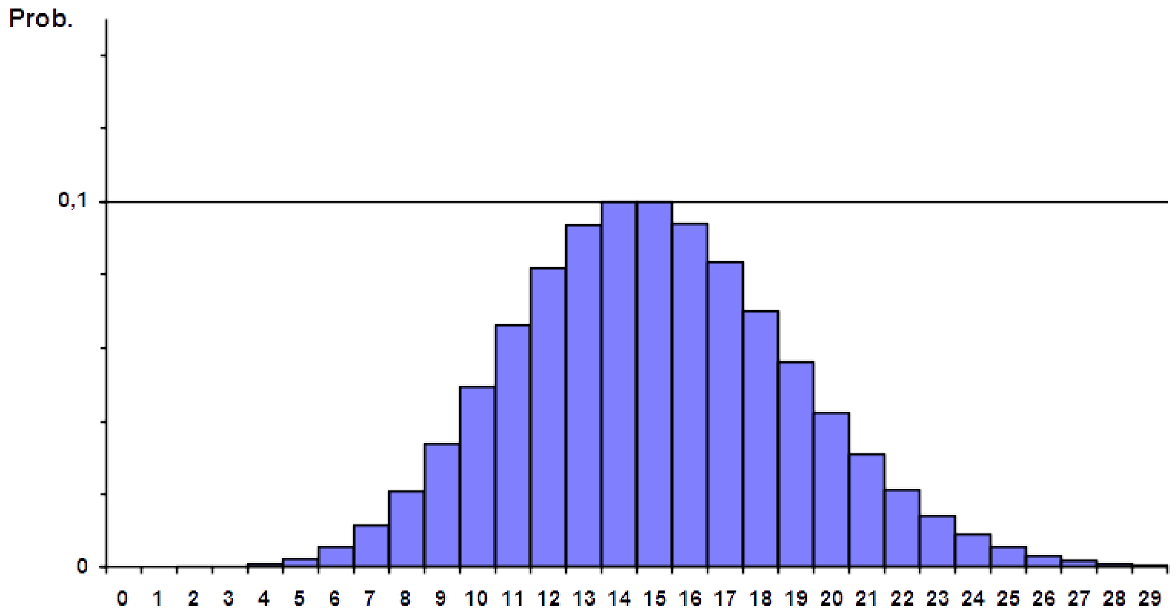
1. Determining the spatial distribution of the traffic
   – The statistics are carried in the network according to the original routing algorithm (simulation), and the elements of the network (nodes, lines) record the statistics transferred by them.
2. Determining the time distribution of the traffic
   – First, the elements sum up the recorded statistics (numerical method), and then the effect of their finite capacities to the time distribution of the traffic is taken into consideration.

TFA in more details

- Traffic model:
  - packet-throughput histogram for the nodes
  - bit-throughput histogram for the lines
- Summation is done by convolution
- The finite capacities are taken into consideration using an iterative algorithm
- Results
  - packet/bit-throughput histogram
  - delay distribution

Correction for the finite capacity – illustration

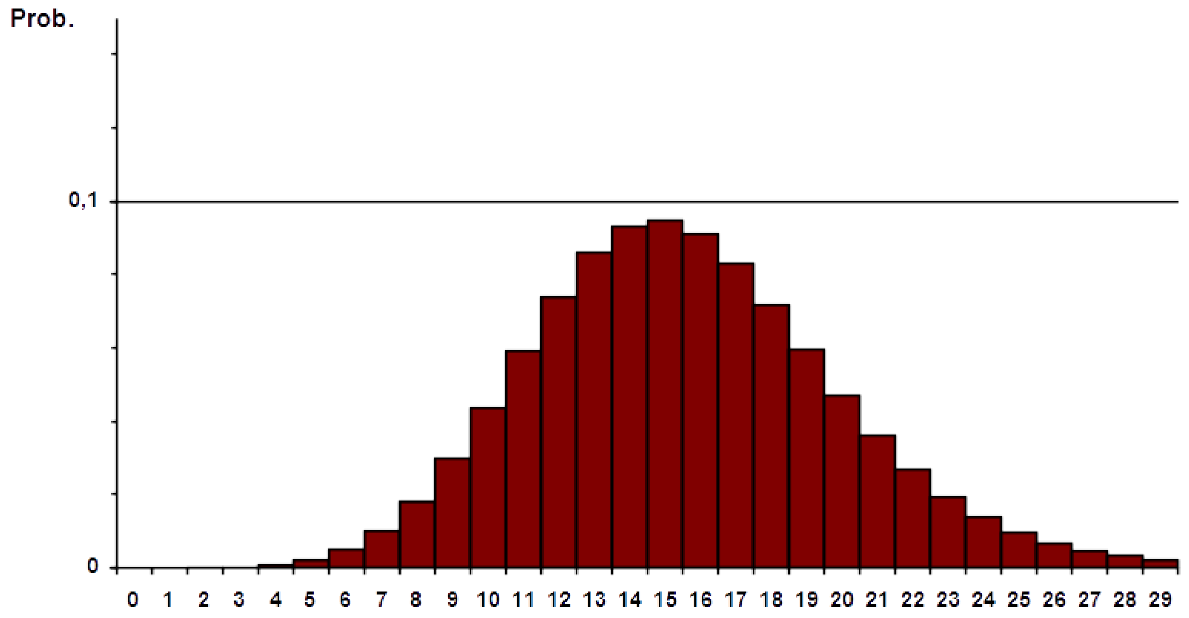
- Let $p_T[k]$ denote the probability that **exactly *k* packets arrive to the node in *T* time**.



$p_T[k]$ – the original PDF

Correction for the finite capacity – illustration

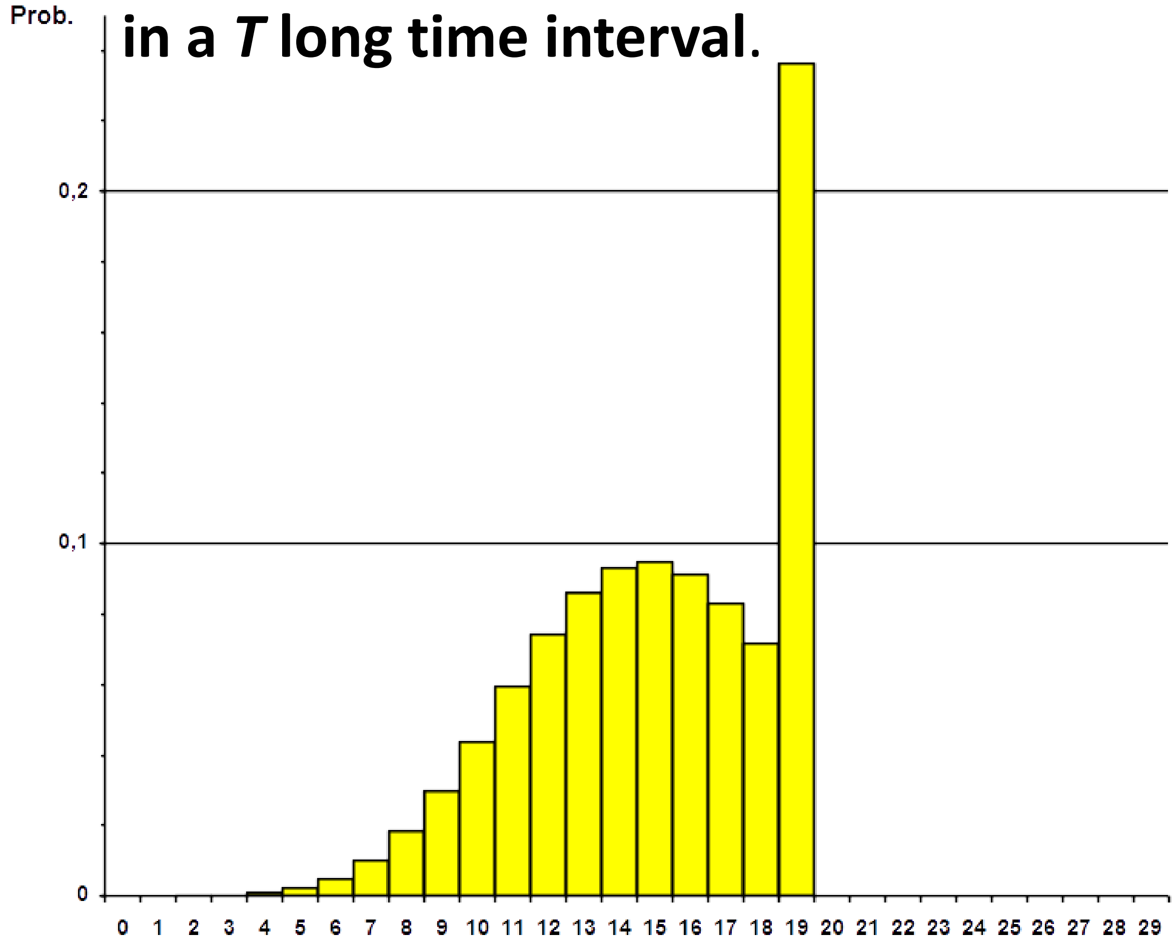
- Let $\boldsymbol{p_T^*[k]}$ denote the probability that **exactly *k* packets require service in a *T* long time interval**. (fresh arrivals + queueing ones)



$\boldsymbol{p_T^*[k]}$ – result of convolutions

Correction for the finite capacity – illustration

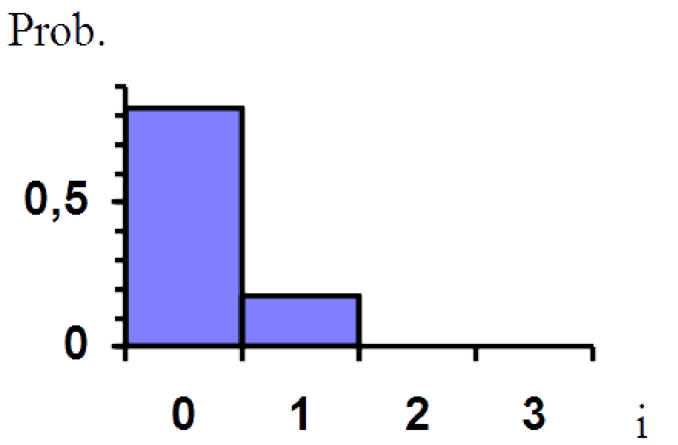
- Let ***$P_T[k]$*** denote the probability that **exactly *k* packets are serviced in a *T* long time interval**.



$P_T[k]$ – result of capacity thresholding

Correction for the finite capacity – illustration

- Let ***D_T[i]*** denote the probability that the **packets are delayed by *i* number of *T* long time intervals**.



D_T[i] – delay distribution

(Lencse 2001)

Combination of TFA and DES – 1.

- Typical situation: a network has a critical part
  - The network contains a high number of elements
  - Its critical part contains much less elements
- The vast majority of the events occur in the non-critical part, and they influence the behavior of the critical part only through their statistical characteristics

(Lencse 2004)

Combination of TFA and DES – 2.

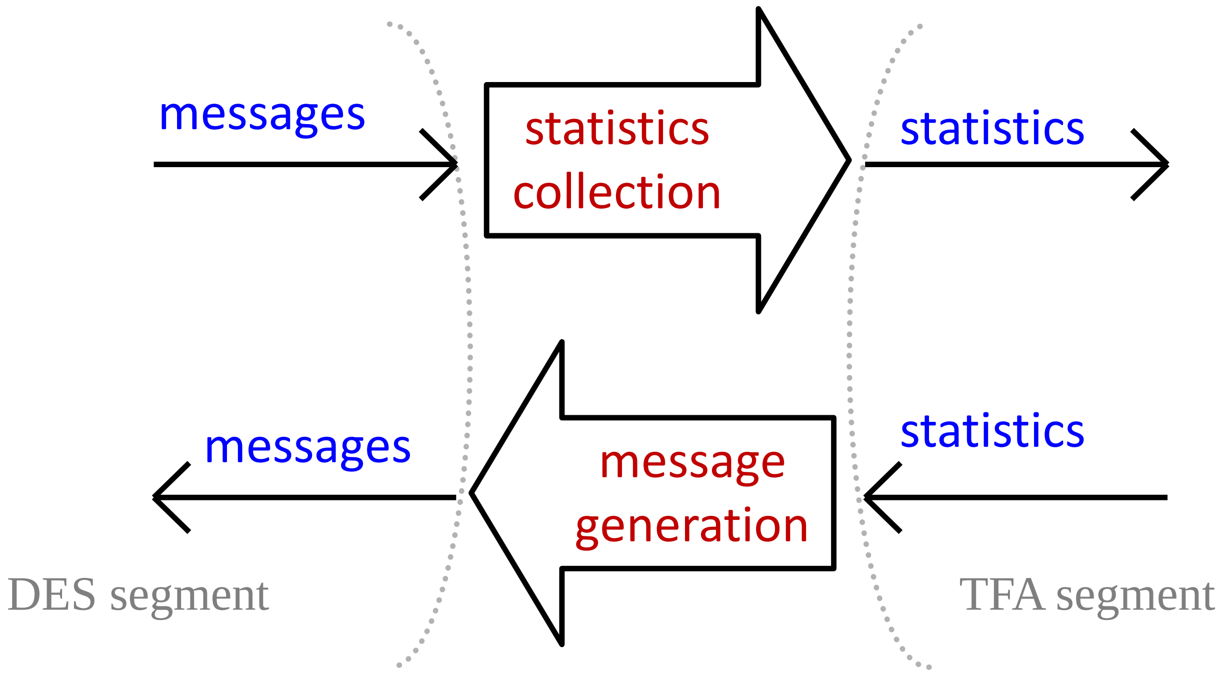
- Why is it worth combining TFA and DES?
  - *Using DES for the whole system would result in far too many events, that is, unacceptably long execution time.*
  - *Applying TFA for the whole system would not model the behavior of the critical part faithfully enough.*
- How can they be combined?
  - DES is applied for the critical part of the network only
  - TFA is used for the rest (and much larger) part of the network

Problems to solve

- The TFA and the DES parts need to exchange traffic information, but their traffic representations are different:
  - TFA uses statistics for the description of the traffic
  - DES uses individual messages for each data packet
  - *Conversion between the two kinds of traffic representations is needed!*
- The virtual time usage of TFA and DES is different:
  - TFA: gives a snapshot – virtual time is just a parameter
  - DES: virtual time plays a key role – see the algorithm of the event-driven DES
  - *This difference must be handled!*

Bidirectional conversion of traffic

- The basic idea:



  – Earlier results of SSM research regarding statistics collection (Lencse 1998b) may be reused

Interworking of TFA and DES

- Under interworking of TFA and DES we mean the following cooperation (it happens one or more times that):
  1. DES sets up the virtual time of TFA
  2. DES provides input parameters to TFA
  3. DES calls TFA for the evaluation of the part of the system that was trusted for TFA
  4. TFA runs
  5. TFA returns parameters to DES

Communication between TFA and DES

A. The DES part and the TFA part are two processes and communicating with each other by inter-process communications (using PVM/MPI or named pipes, etc.).

B. TFA is a function/procedure within the DES program and is called sometimes, but it does not use the virtual time of the DES engine for its internal purposes.

C. TFA is a set of functions within the DES program and is called sometimes, and it uses the virtual time of the DES engine (and some of its services) for its internal purposes.

- Because of its advantages let us consider "C"

Interworking of TFA and DES – communication “C”

1. DES sets up the virtual time of TFA by scheduling its starting event to the appropriate virtual time. (It is put into the FES.)
2. DES provides input parameters to TFA by sending messages to the module(s) responsible for TFA.
3. DES not at all calls TFA, it runs when its virtual time comes.
4. When TFA runs, it may use the event scheduling system, but it needs some discussion (because of the possible interleaving of the TFA internal events with the DES events). – Discussed in details in (Lencse 2004)
5. TFA returns parameters to DES by sending messages to the appropriate module(s) in the DES segment

Parallelization of Combined TFA and DES

- What can be executed in parallel?
- How can it be done?
- What speed-up can we expect?

PCDT: Parallel Combined DES and TFA

(Lencse 2005)

What can be Executed in Parallel?

- First ideas:
  - Execute the DES and the TFA segments in parallel
  - Execute multiple DES segments in parallel
  - Execute multiple TFA segments in parallel

- They are to be refined…

Parallel Combined DES and TFA on 2 CPUs

- The original combination of the DES and TFA uses bidirectional data exchange
- The aim of the combination is usually to get proper load for the DES part from the TFA part (that is we need the TFA $\rightarrow$ DES traffic)
- The parallel combined DES and TFA method (PCDT) uses prediction for the DES segment $\rightarrow$ TFA segment traffic

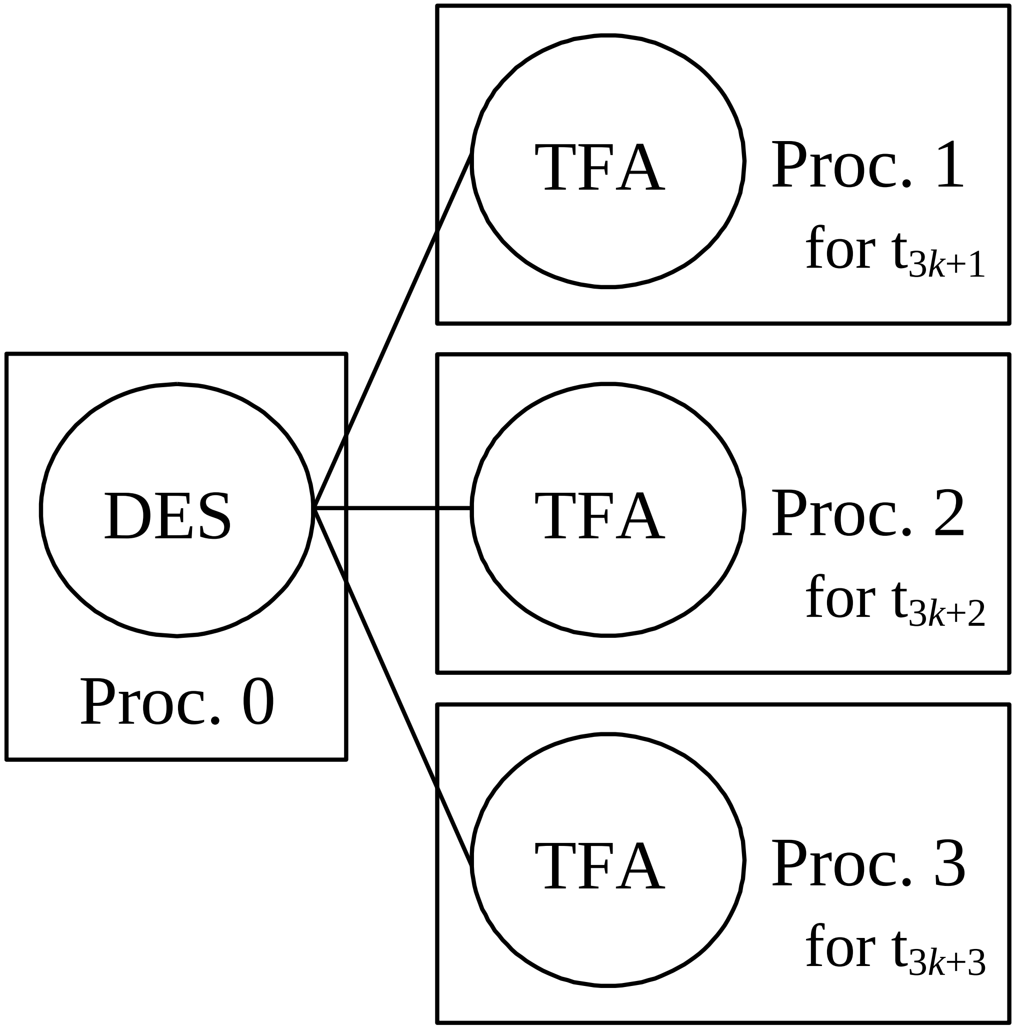
PCDT on 2 CPUs

- In the beginning
  - at t_0 : DES $\rightarrow$ TFA : “TFA request” with timestamp t_1 and with other parameters
  - from t_0 to t_1 : DES and TFA segments run independently
- In the general step
  - at t_i : TFA $\rightarrow$ DES “TFA result” (traffic info)
  - at t_i : DES $\rightarrow$ TFA : “TFA request” with timestamp t_{i+1} and with other parameters
  - from t_i to t_{i+1} : DES and TFA segments run independently

The basic idea of PCDT on n+1 CPUs

- Let us imagine:
  - If for example:
    - size_of(TFA segment)=100*size_of(DES segment)
    - efficiency(TFA segment)=10*efficiency(DES segment)
  - Then
    - computation(DES segm.)=10*computation(TFA segm.)
- Idea:
  - Let us use multiple processors for multiple TFA segments that are all examining the **same network** but **for different t_i virtual times**!

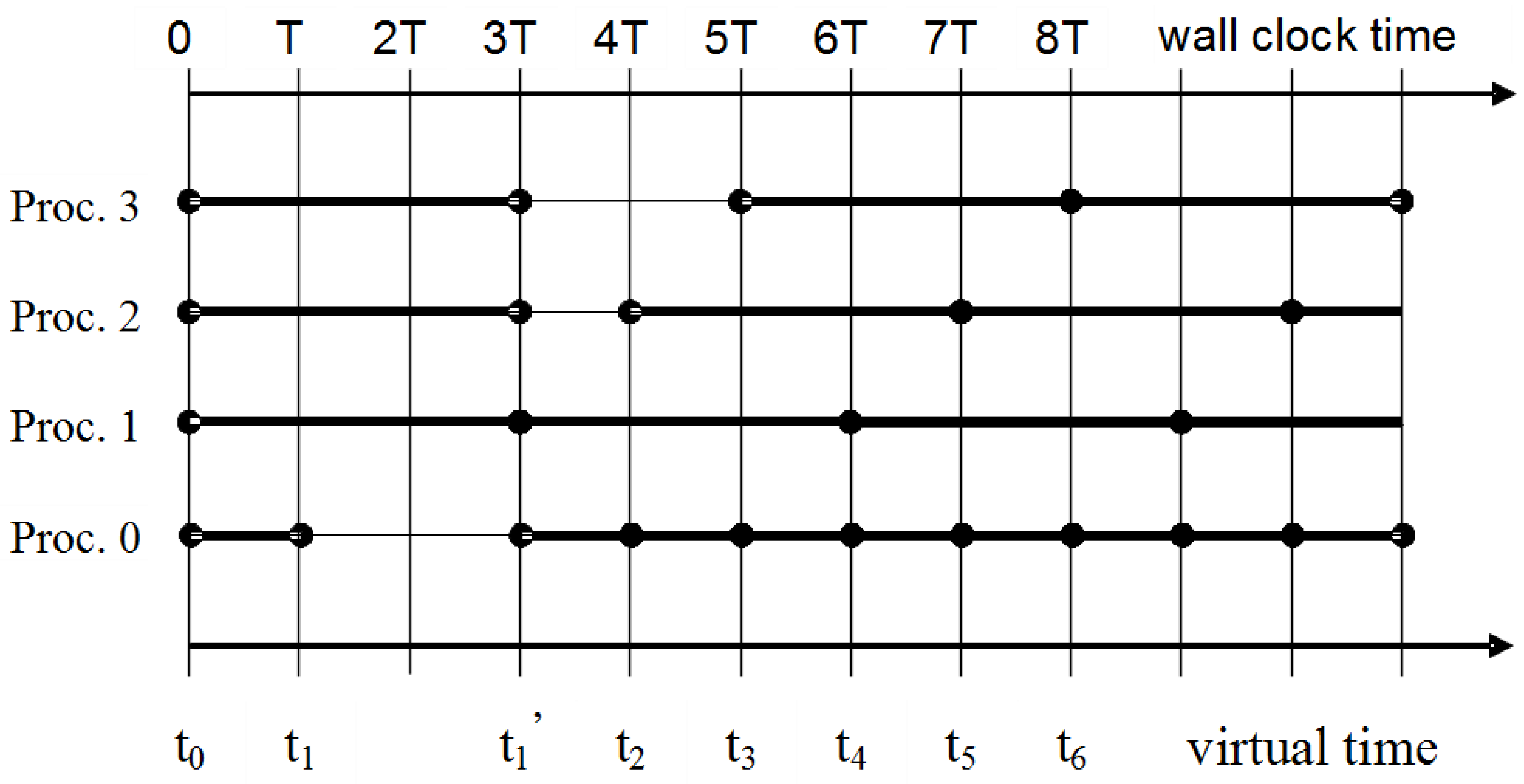
Example: PCDT Using 3+1 Processors



Note:

The same TFA segment is executed parallel for different t_i points of virtual time!

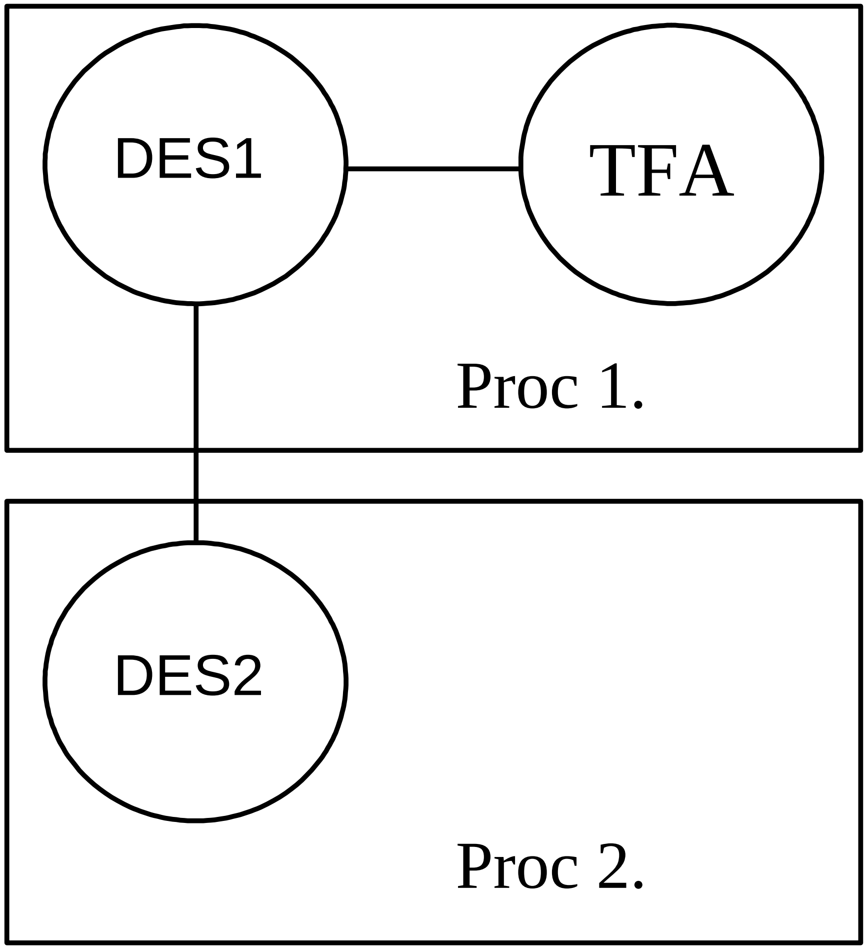
Operation of PCDT Using 3+1 CPUs



Applicability Criteria of PCDT Using n+1 CPUs

- The results of TFA produced at t_i characterize the TFA$\rightarrow$DES traffic well until t_{i+1}. (same as in the non-parallel case)
- The DES$\rightarrow$TFA traffic is either negligible, or predictable for the t_{i+n} point of virtual time at t_i point of virtual time. (new criterion)

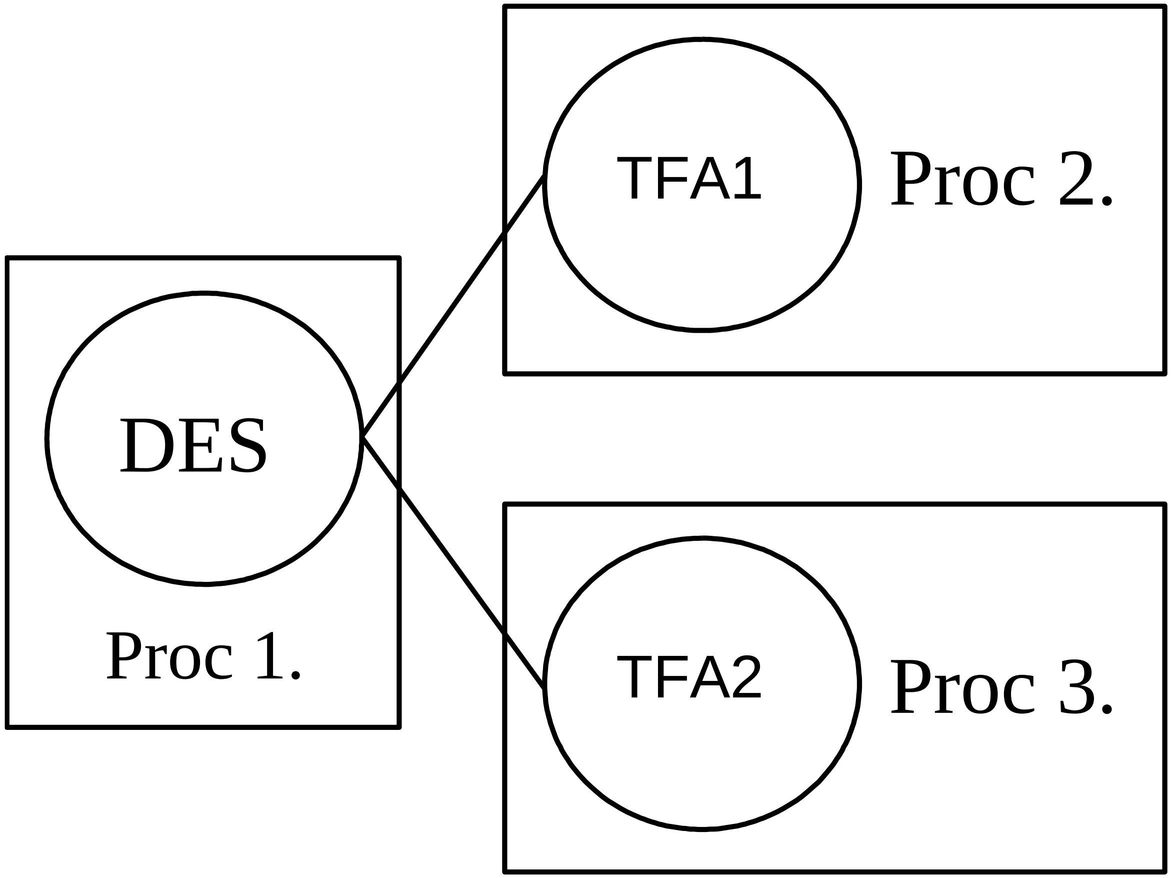
Multiple DES Segments are Executed in Parallel



Note:

The DES part of the combined TFA and DES is further divided and executed by two processors (using a PDES method)

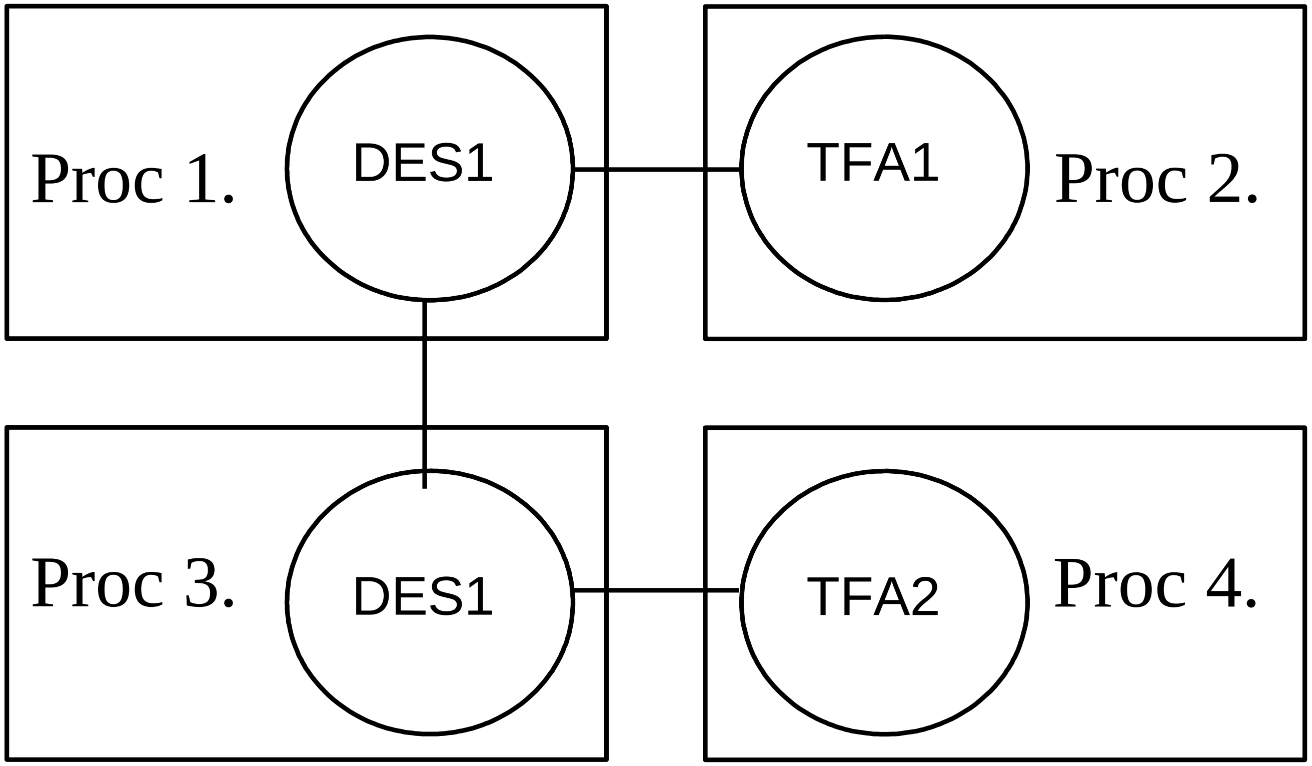
Multiple **Independent** TFA segments are Executed in Parallel



Note:

There is **no cross traffic** between TFA1 and TFA2

Parallel execution of multiple DES and multiple **independent** TFA segments



Note:

There is **no cross traffic** between TFA1 and TFA2

References – 1

- Fujimoto, R. M. 1990. "Parallel Discrete Event Simulation", *Communications of the ACM*, vol. **33** no. 10, pp. 31-53
- Lencse, G. 1997. "Efficient Simulation of Large Systems - Transient Behaviour and Accuracy", *Proceedings of the 1997 European Simulation Symposium (ESS'97),* Passau, Germany, Oct. 19-23, SCS Europe, pp. 660-665.
- Lencse, G. 1998a. "Efficient Parallel Simulation with the Statistical Synchronization Method", *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation (CNDS'98)*, San Diego, CA. Jan. 11-14, SCS International, pp. 3-8.
- Lencse, G. 1998b. "Statistics Collection for the Statistical Synchronisation Method", *Proceedings of the 1998 European Simulation Symposium (ESS'98)*, Nottingham, UK, Oct. 26-28, SCS Europe, pp. 46-51.
- Lencse, G. 1999a. "Applicability Criteria of the Statistical Synchronization Method", *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation (CNDS'99)*, San Francisco, CA. Jan. 17-20, SCS International, pp. 159-164.

References – 2

- Lencse, G. 1999b. "Design Criterion for the Statistics Exchange Control Algorithm used in the Statistical Synchronization Method", *Proceedings of the Advanced Simulation Technologies Conference (ASTC 1999)* part of the 32nd Annual Simulation Symposium, April 11-15, 1999, San Diego, CA, USA, pp. 138-144.
- Lencse Gábor 2000. "*Kommunikációs rendszerek hatékony szimulációjának egyes kérdései*", PhD dissertation in Hungarian, Department of Telecommunications, Budapest University of Technology and Economics, public defense: May 9, 2001.
- Lencse, G. 2001. "Traffic-Flow Analysis for Fast Performance Estimation of Communication Systems", *Journal of Computing and Information Technology*, vol. **9**, no. 1, pp. 15-27.
- Lencse, G. 2002. "Parallel Simulation with OMNeT++ using the Statistical Synchronization Method", *Proceedings of the 2nd International OMNeT++ Workshop*, Jan. 8-9, 2002, Technical University Berlin, Berlin, Germany, pp. 24-32.

References – 3

- Lencse, G. 2004. "Combination and Interworking of Traffic-Flow Analysis and Event-Driven Discrete Event Simulation", *Proceedings of the 2004 European Simulation and Modelling Conference (ESM® '2004),* Paris, France, Oct. 25-27, EUROSIS-ETI, pp. 89-93.
- Lencse, G. 2005. "Speeding up the Performance Analysis of Communication Systems", *Proceedings of the 2005 European Simulation and Modelling Conference (ESM® '2005),* Porto, Portugal, Oct. 24-26, EUROSIS-ETI, pp. 329-333.
- Pongor, Gy. 1992. "Statistical Synchronization: a Different Approach of Parallel Discrete Event Simulation", *Proceedings of the 1992 European Simulation Symposium (ESS'92)* , Nov. 5-8, 1992, The Blockhaus, Dresden, Germany, SCS Europe, pp. 125-129.

My papers can be downloaded from: http://www.hit.bme.hu/~lencse/publications/

Thank you for your attention!

Questions?

Dr. Gábor Lencse

Professor

Dept. of Telecommunications, University of Győr

lencse@sze.hu