FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

T569 –SISTEMAS DE TEMPO REAL

Aula 3

Prof. Marcelo Sousa

Agenda

- Segurança e Confiabilidade
- Tolerância a Falhas por Software
- Tipos de Tarefas de Tempo Real
- Restrições de Tempo

Segurança e Confiabilidade

- Sistema Seguro
  - É um sistema que nunca causa danos, mesmo em caso de fala

- Sistema Confiável:
  - É um sistema que permanece em funcionamento por longos períodos de tempo sem apresentar uma falha

Segurança e Confiabilidade

- Em sistemas tradicionais
  - É comum observar sistemas onde essas características são independentes.
  - Exemplos:
    - Sistemas de processamento de texto.
      - Pouco confiável
      - Seguro
    - Arma de fogo
      - Confiável
      - Pouco seguro

Segurança e Confiabilidade

- *Fail-Safe State*
  - Estado no qual o sistema deve se encontrar em caso de falha do sistema e nenhum dano deve ser gerado como resultado.
  - É necessária preocupação com a transição entre os estados anteriores à entrada no *fail-safe state*.

Segurança e Confiabilidade

- *Fail-Safe State*
  - Sistema de Semáforos não confiável e inseguro
    - Falha várias vezes durante um dia de operação, não sincronizando os semáforos.
  - Sistema de Semáforos Seguro, mas ainda não confiável
    - Criação de um *fail-safe state*, onde todas as luzes ficam amarelas em caso de falha.

Segurança e Confiabilidade

- Sistema *Safety-Critical*
  - Sistema que, em caso de falha, pode causar danos severos.
  - Exemplo:
    - Sistema de navegação de um avião
    - Em caso de falha do computador do avião, não há *fail-safe state*, pois não é possível desligar as turbinas em caso de falha, por exemplo.

Segurança e Confiabilidade

- Sistema *Safety-Critical*
  - Em sistemas onde não é possível a existência de *fail-safe state*, a única forma de garantir a segurança é aumentando a confiabilidade.
  - Exemplo:
    - Em sistemas de controle de voo (*fly-by-wire aircraft*) a maioria dos controles é realizado por um computador. Logo, falha no computador é inaceitável
    - Taxa de Falhas: 1 a cada 10^9 horas de voo (milhões de anos de voo contínuo)

Segurança e Confiabilidade

- Alta Confiabilidade – Software
  - *Error Avoidance*
    - Utilização de boas práticas de programação
    - Utilização de metodologias de desenvolvimento
    - Adoção de padrões de programação
  - *Error Detection and Removal*
    - Erros ainda podem ser encontrados
    - Realização de testes extensivos e intensivos
    - Realização de revisões constantes
    - Todos os erros identificados devem ser corrigidos

Segurança e Confiabilidade

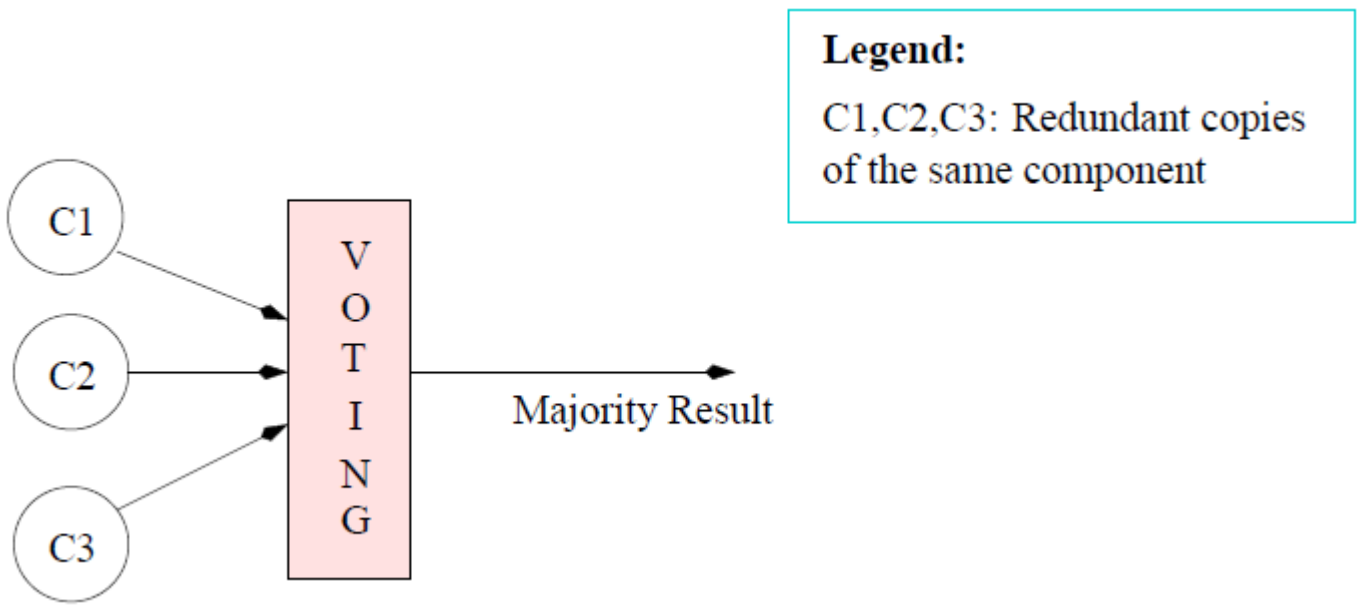
- Alta Confiabilidade – Software
  - *Fault-Tolerance*
    - Não interessa o quanto as duas técnicas anteriores sejam utilizadas, é praticamente impossível existir um sistema que não possua falhas (*error-free*)
    - Mesmo que o erro haja no sistema, este deve ser tolerado e sua computação deve ser realizada corretamente.

Segurança e Confiabilidade

- Alta Confiabilidade – Hardware
  - *Built In Self Test (BIST)*
    - O sistema analisa periodicamente os componentes do sistema.
    - Em caso de detecção de falha este componente é removido do sistema e outro é componente redundante é inserido no sistema.

Segurança e Confiabilidade

- Alta Confiabilidade – Hardware
  - *Triple Modular Redundancy (TMR)*



Tolerância a Falhas por Software

- *N-Version Programming*
  - Adaptação da Técnica TMR
  - Desenvolvimento de times independentes para a o mesmo conjunto de funcionalidades
  - Tarefas rodam concorrentemente e seus resultados são comparados através de votação
  - Utiliza o princípio que equipes diferentes podem cometer erros diferentes que são eliminados quando postos em votação.

Tolerância a Falhas por Software

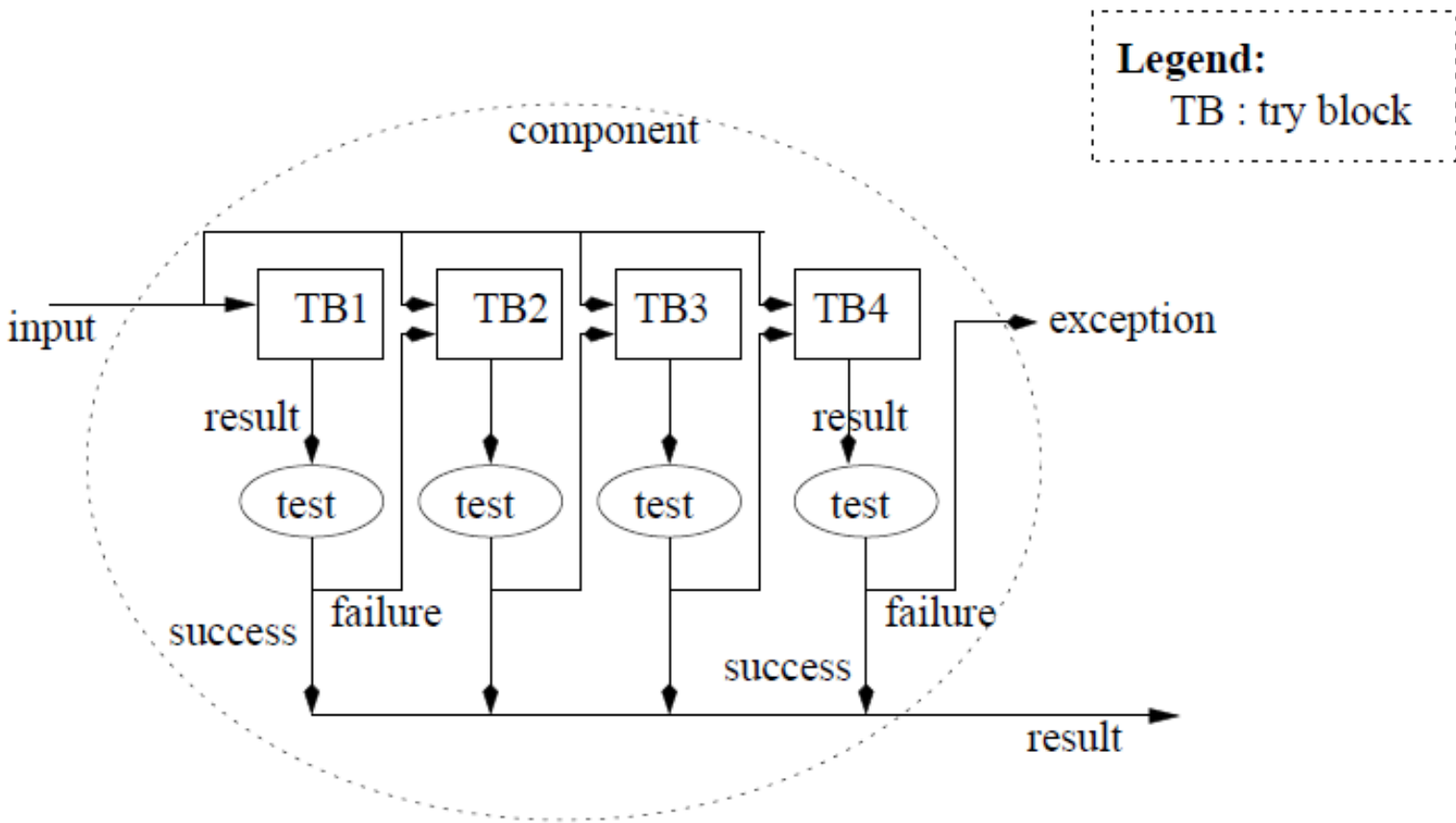
- *N-Version Programming*
  - Problemas:
    - Erros comuns podem surgir, pois problemas de difícil resolução para uma equipe usualmente são difíceis para outra equipe.
    - Pouco eficiente em sistemas de complexidade elevada
    - **Correlação Estatística das Falhas:** Os componentes do sistema podem falhar pelos mesmos motivos

Tolerância a Falhas por Software

- *Recovery Blocks*
  - Utilização de blocos redundantes (*try blocks*)
  - Cada *try block* é desenvolvido intencionalmente com um algoritmo diferente
  - Quando há falha no resultado de alguns dos *try blocks* outro bloco é habilitado para ser executado

Tolerância a Falhas por Software

- *Recovery Blocks*



Tolerância a Falhas por Software

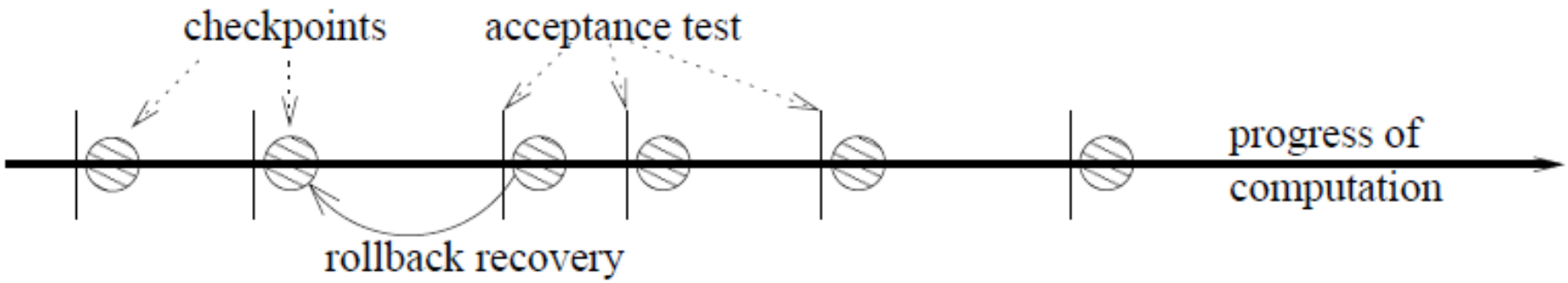
- *Recovery Blocks*
  - Problemas:
    - **Correlação Estatística das Falhas:** Os componentes do sistema podem falhar pelos mesmos motivos
    - Apenas pode ser utilizada se o *deadline* de uma tarefa for muito maior que seu tempo computacional

Tolerância a Falhas por Software

- *Checkpointing and Rollback Recovery*
  - Testes de coerência dos estados são realizados periodicamente.
  - Em caso de sucesso, todos os estados do sistema são armazenados em forma de *backup* (*checkpoints*)
  - Em caso de falha, o sistema retorna ao *checkpoint* anterior (*rollback recovery)*
  - Após o *rollback recovery*, o processo computacional é retomado iniciado novamente

Tolerância a Falhas por Software

- *Checkpointing and Rollback Recovery*



Tarefas de Tempo Real

- Definição: é uma tarefa dentro do sistema no qual expressões quantitativas de tempo são utilizadas para descrever seu comportamento.

- Classificações:
  - *Hard real-time*
  - *Soft real-time*
  - *Firm real-time*

Tipos de Tarefas de Tempo Real

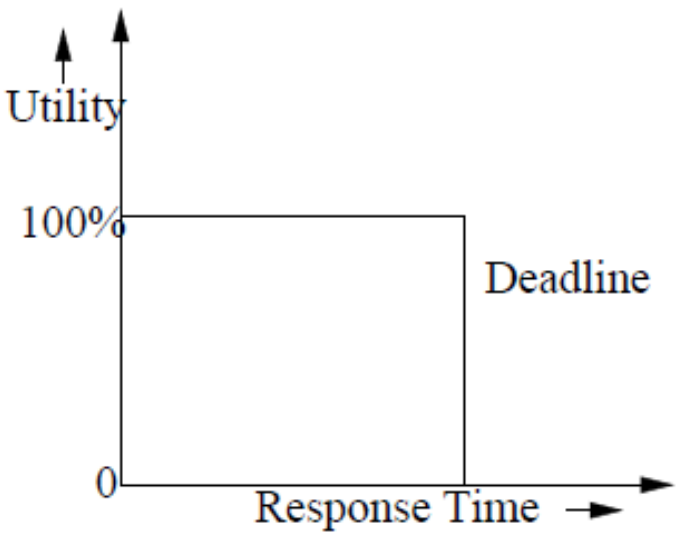
- *Hard Real-Time Tasks*
  - É uma tarefa que DEVE produzir seus resultados até um certo limite de tempo;
  - O sistema é considerado falho caso nenhuma tarefa deste tipo seja atendida;
  - Sistemas que possuem *Hard Real-Time Tasks*, em sua maioria, são *Safety-Critical*
  - *Deadline* : de poucos *μs* a poucos *ms*

Tipos de Tarefas de Tempo Real

- *Hard Real-Time Tasks*
  - Exemplos:
    - Robô
      - Detecção e reação a objetos são *hard-real time tasks*
    - Sistema Anti-Míssel
      - Detecção de mísseis, movimentação do canhão e disparo são *hard-real time tasks*
  - OBS: Não é necessário que uma tarefa seja atendida o mais rápido possível, apenas que ela seja atendida em seu tempo especificado sem o ultrapassar.

Tipos de Tarefas de Tempo Real

- *Firm Real-Time Tasks*
  - Este tipo de tarefa é associado a um *deadline*
  - Não implica em falha do sistema em caso de não atendimento do *deadline*
  - Seu resultado é descartado caso exceda o *deadline*



Tipos de Tarefas de Tempo Real

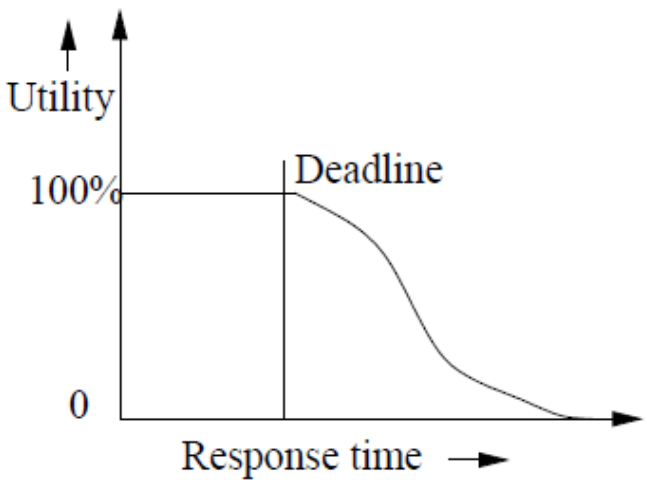
- *Firm Real-Time Tasks*
  - Exemplos:
    - Vídeo Conferência
      - A transmissão de pacotes é *firm real time task*
    - Sistema de rastreamento de inimigos baseado em satélites
      - A transmissão de pacotes até a base central é *firm real time task*
  - *Deadline* : de poucos *ms* a centenas de *ms*

Tipos de Tarefas de Tempo Real

- *Soft Real-Time Tasks*
  - Também possui restrições de tempo
  - Tempo necessário para a resposta é expressa através de valores aproximados
  - Valores tardios no atendimento de uma tarefa demonstram degradação do sistema e não falha

Tipos de Tarefas de Tempo Real

- *Soft Real-Time Tasks*
  - Exemplos:
    - Web Browser
      - Resposta a uma requisição de URL é *soft real time task*
    - Reserva de assentos de aviões
      - Tempo de resposta à requisição de reserva é *soft real time task*



Tipos de Tarefas de Tempo Real

- *Non-Real-Time Tasks*
  - Não está associada a nenhum tipo de restrição de tempo (ou quase).
  - No entanto, antigamente várias tarefas que são *soft real time* eram *non-real time*
  - *Non-real-time task*
    - *Deadline: Alguns minutos, horas ou mesmo dias*
  - *Soft-real-time task*
    - *Dealine:Poucos segundos*

Restrições de Tempo

- O perfeito funcionamento (*correctness*) depende de dois fatores:
  - Lógico
    - Computação correta
  - Temporal
    - Atendimento dentro do tempo estabelecido

Restrições de Tempo

- Eventos em Sistemas de Tempo Real
  - Eventos de Estímulo
    - Gerados pelo ambiente agindo no sistema
  - Eventos de Resposta
    - Gerados pelo sistema agindo no ambiente

Restrições de Tempo

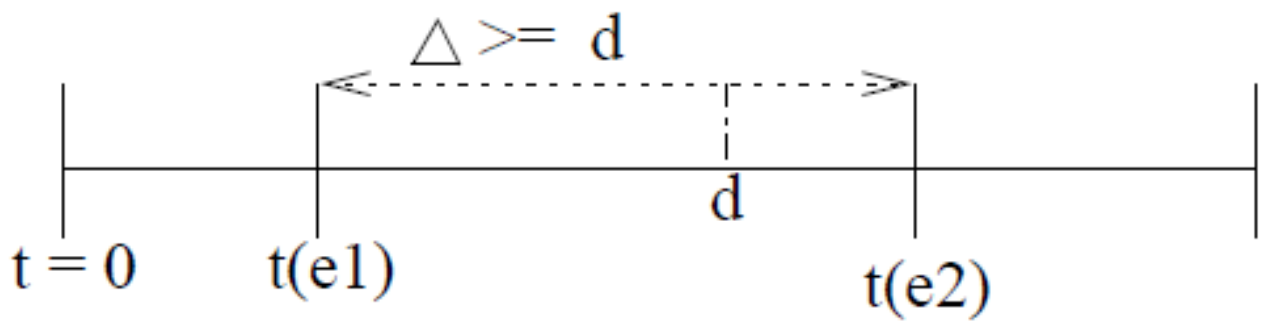
- Classificação
  - Performance
    - Imposta a uma resposta dada pelo sistema
    - Garante que o sistema computacional está funcionando perfeitamente
  - Comportamental
    - Imposta a um estímulo gerado pelo ambiente
    - Garante que o ambiente se comporta bem

Restrições de Tempo

- Classificação (comportamental ou *performance*)
  - *Delay*
  - *Deadline*
  - *Duration*

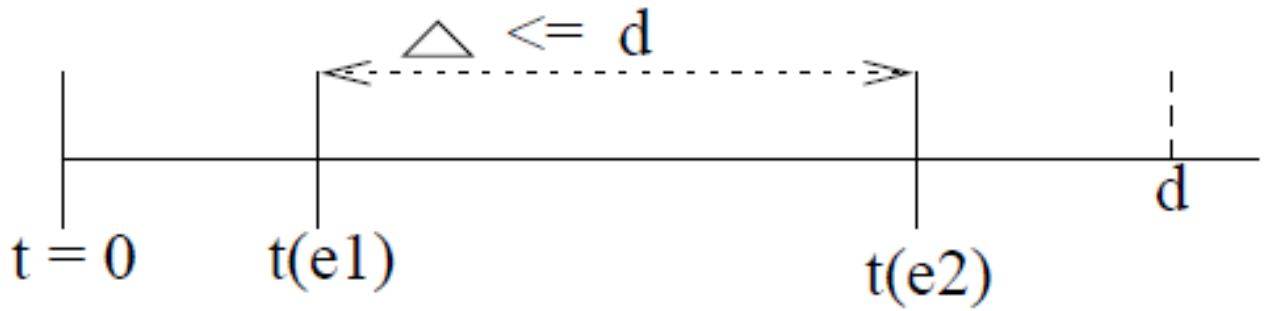
Restrições de Tempo

- *Delay*
  - Tempo mínimo entre a ocorrência de dois eventos arbitrários



Restrições de Tempo

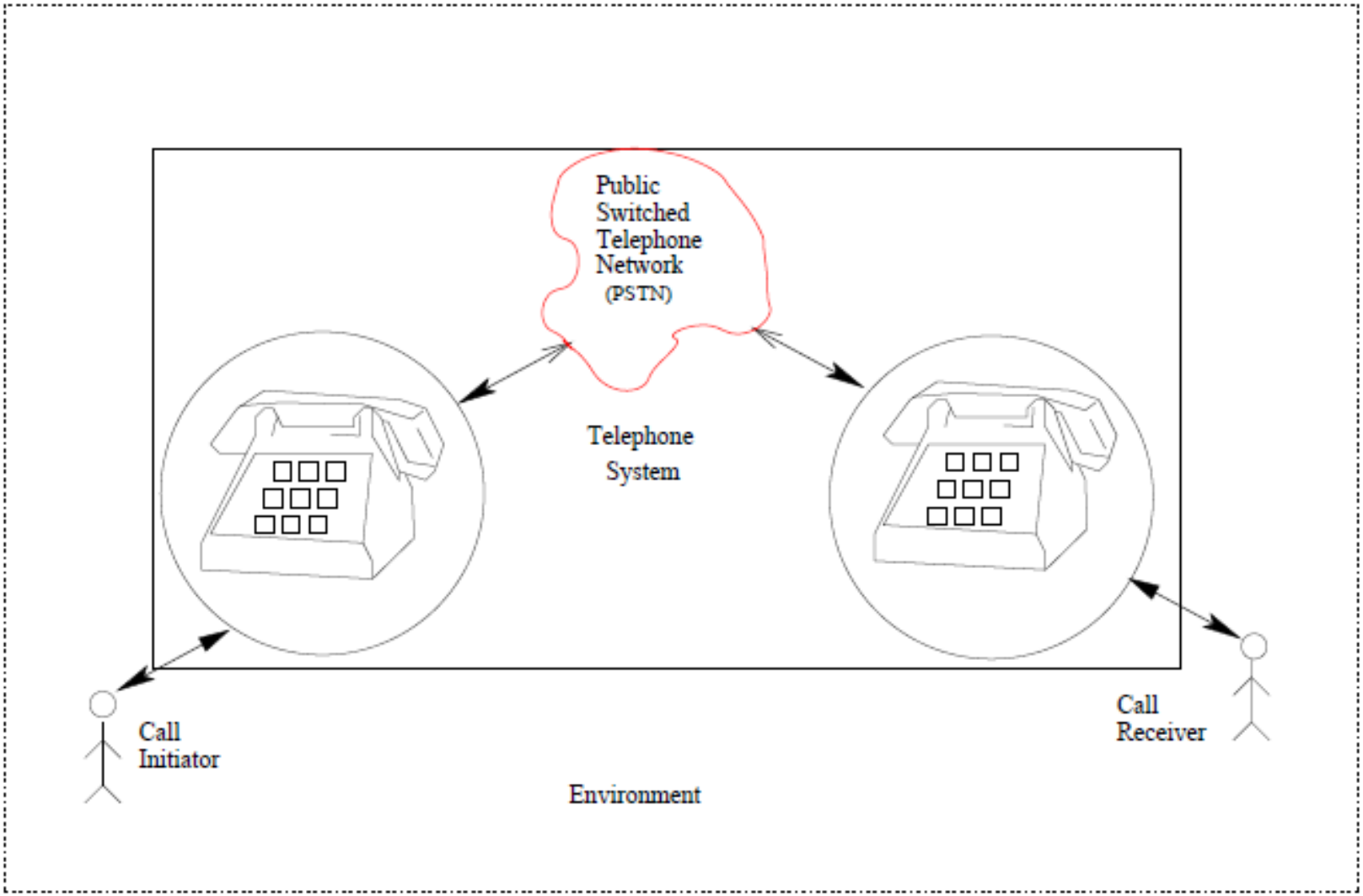
- *Deadline*
  - Tempo máximo da ocorrência de dois eventos



Restrições de Tempo

- *Duration*
  - Tempo específico de atuação de um evento

Restrições de Tempo



Restrições de Tempo

- Exemplos
  - *Deadline*
    - *Stimulus Stimulus* (SS) – Comportamental
      - Uma vez completado o primeiro dígito o usuário deve digita o próximo em até 5 segundos. Caso contrário o tom de ocupado soará.
    - *Stimulus Response* (SR) – Performance
      - Um vez levantado o fone do gancho o sistema deve produzir um tom de discagem. Caso contrário o som de beeping é produzido até que o fone seja colocado no gancho

Restrições de Tempo

- Exemplos
  - *Deadline*
    - *Response Stimulus* (RS) – Comportamental
      - Um vez soado o tom de discagem, o primeiro digito deve ser discado em até 30 segundos. Caso contrário o sistema entra em espera e um tom de ocupado é gerado.
    - *Response Response* (RR) – Performance
      - Uma vez que o tom de chamando do destinatário soar um tom de chamando equivalente também deve soar. Caso contrário a chamada deve ser terminada.

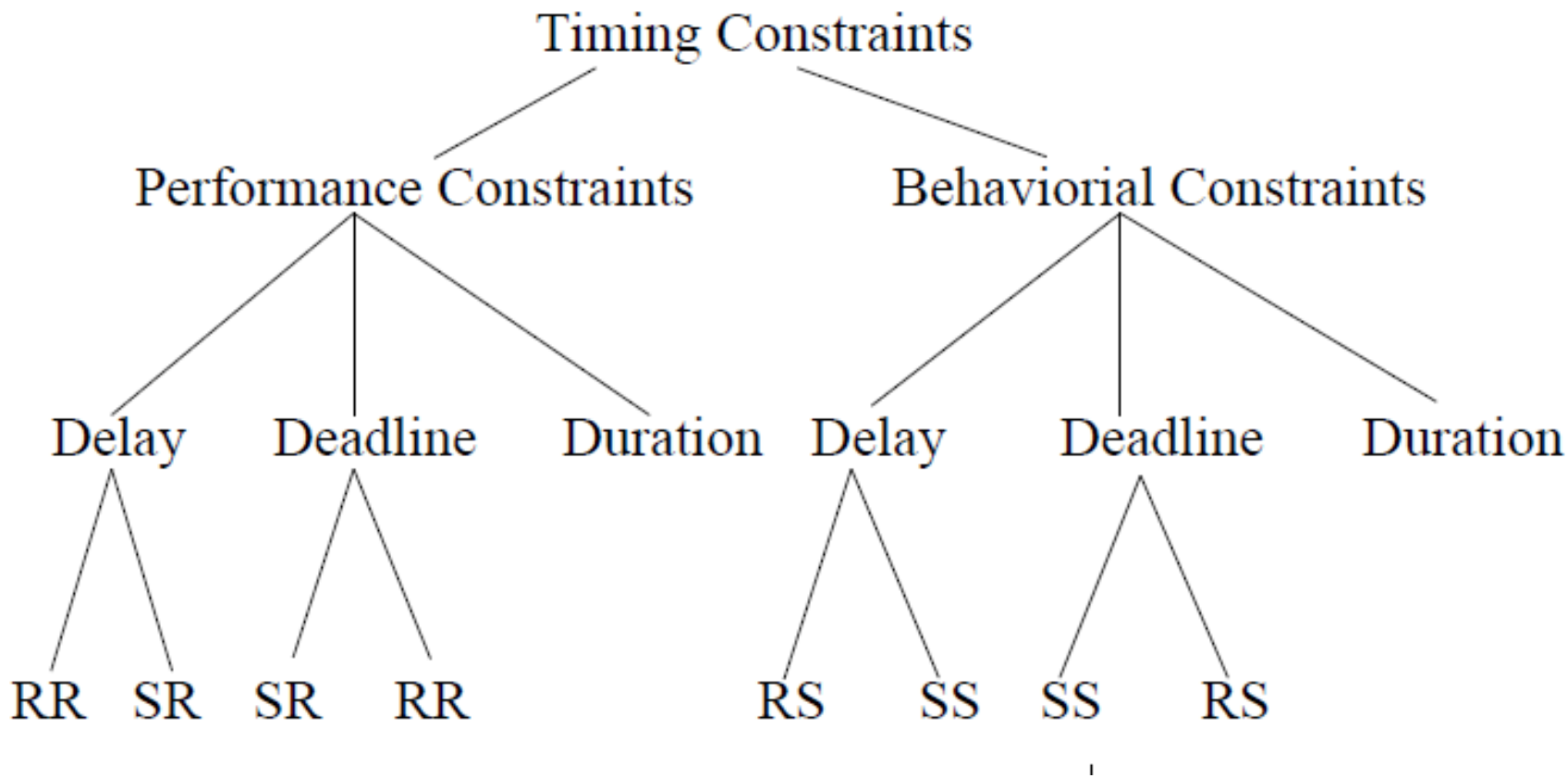
Restrições de Tempo

- Exemplos
  - *Delay*
    - *Stimulus Stimulus* (SS) – Comportamental
      - Uma vez discado um digito, o próximo digito só pode ser discado após 1 segundo. Caso contrário um som de beep é soado enquanto o fone não é posto no gancho.

Restrições de Tempo

- Exemplos
  - *Duration*
    - *Se você pressionar o botão por menos de 15 segundos, o sistema se conecta com a operadora local*
    - *Se você pressionar por um intervalo de 15 a 30 segundos, o sistema se conecta a operadora internacional*
    - *Se pressionar por um tempo maior que 30 segundos, produz um tom de discagem quando liberado*

Restrições de Tempo



Referência

- http://nptel.iitm.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Real%20time%20system/New_index1.html