

Sistemas Computacionais Distribuídos

**Prof. Marcos José Santana
SSC-ICMC-USP**

São Carlos, 2008

Grupo de Sistemas Distribuídos e Programação Concorrente

**Departamento de Sistemas
de Computação - SSC**

Sistemas Computacionais Distribuídos

5a. Aula

RPC – Remote Procedure Call

Conteúdo

- ◆ **Conceito**

RPC – Conceitos Básicos

- **Mecanismo: simples, rápido, versátil e largamente utilizado;**
- **Baseado na estrutura:**
requisição/resposta/bloqueio;
- **Copia modelo de chamadas locais (LPC):**
 - Chamada procedimento
 - Bloqueio do programa principal
 - Retorno do resultado

RPC – Conceitos Básicos

- **Mecanismo com:**
 - **Verificação de tipos de dados;**
 - **Chamadas a procedimentos remotos;**
 - **Roteamento automático da chamada;**
 - **Transparência na linguagem de programação;**
 - **Programação manual ou automática;**
 - **Serviços RPC embutidos na linguagem;**
 - **Biblioteca de apoio para aplicações RPC**

RPC – Conceitos Básicos

➤ IPC – Inter-Process Communication

- **Cliente:** envia mensagem;
- **Servidor:** recebe mensagem;
- **Sem sincronismo.**

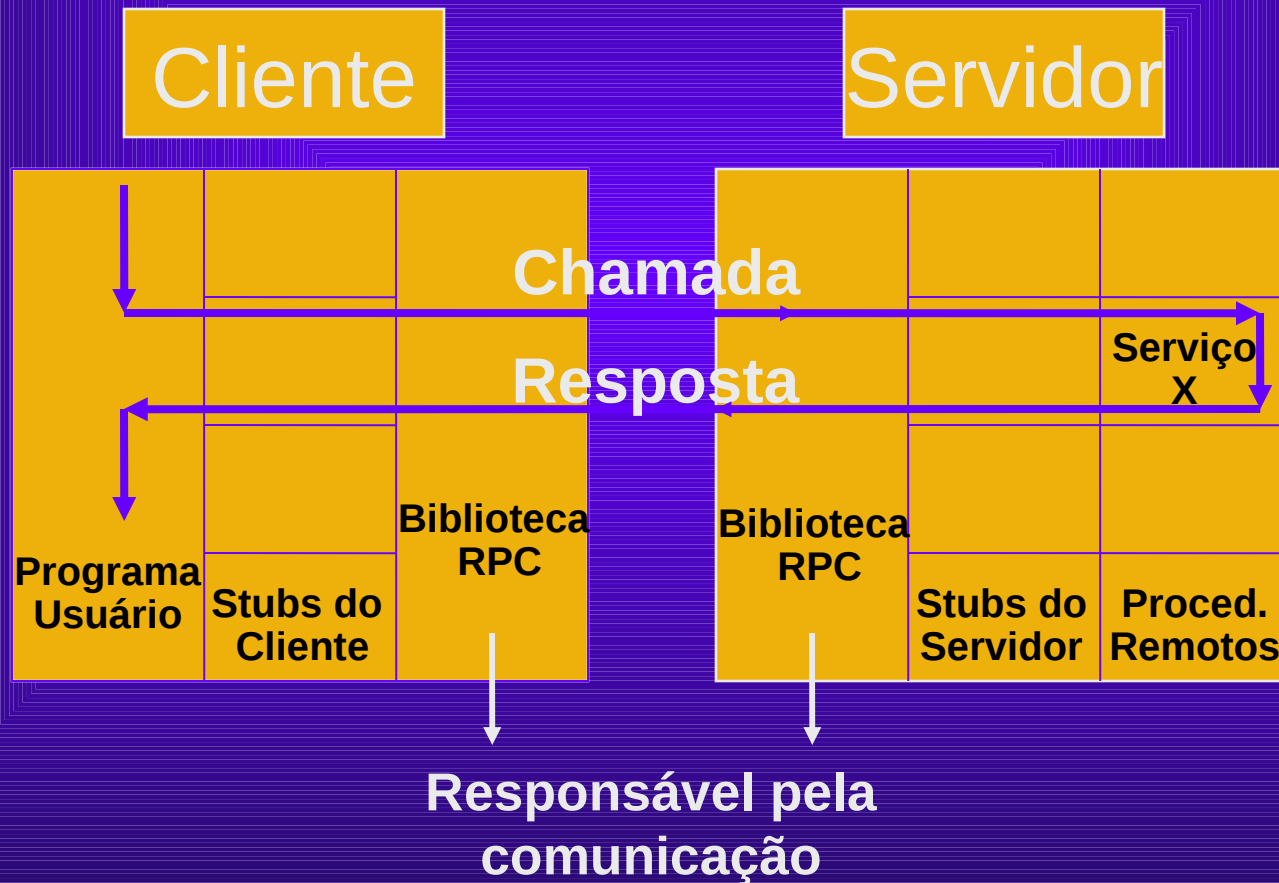
➤ RPC – caso especial de IPC

- **Exige sincronismo (forma “pura”);**
- **Passa parâmetros (clientes ⇒ servidor);**
- **Bloqueia processo origem até resposta;**
- **Libera processamento cliente.**

RPC – Conceitos Básicos

- **RPC – 2 mensagens são usadas:**
 - **Requisição** (cliente \Rightarrow servidor);
 - **Resposta** (servidor \Rightarrow cliente);
 - Sincronismo durante chamada.
- **RPC – copia conceitos gerais de LPC:**
 - Passagem de parâmetros;
 - Pode ser integrado à linguagem;
 - Pode ter a mesma sintaxe;
 - Pode ter a mesma semântica;
 - Integração com o ambiente de programação.

RPC – Mecanismo de Operação



RPC – Mecanismo de Operação

- **STUBs: Manipuladores de dados**
- **STUB cliente (nome procedimento objetivo):**
 - Intercepta chamada remota ;
 - Prepara mensagem para transmissão.
- **STUB servidor:**
 - recebe mensagem;
 - identifica processo;
 - repassa a chamada:
 - Prepara mensagem com resposta.

RPC – Mecanismo de Operação

- **Geração dos STUBs:**
 - **Manual – responsabilidade do programador**
 - **Automática – uso de bibliotecas e geradores**
 - Linguagem de descrição de parâmetros;
 - define interface entre cliente e servidor;
 - Geração dos stubs.

RPC – Mecanismo de Operação

➤ Representação de dados:

- **Ambientes homogêneos:**
 - **Simplificação – nada é necessário.**
- **Ambientes heterogêneos**
 - **Formato intermediário (ex. XDR)**
 - **2 conversões por mensagem**
- **Negociação:**
 - **Se homogêneo – sem conversão**
 - **Se heterogêneo – uso de padrão intermediário**

RPC – Semântica das Chamadas

- **Sistemas Distribuídos:**
 - **Potencial para paralelismo;**
 - **Podem ocorrer falhas:**
 - **Cliente;**
 - **Servidor;**
 - **Meio de comunicação.**
 - **Mecanismos para superar as falhas;**
 - **Manutenção da consistência.**

RPC – Semântica das Chamadas

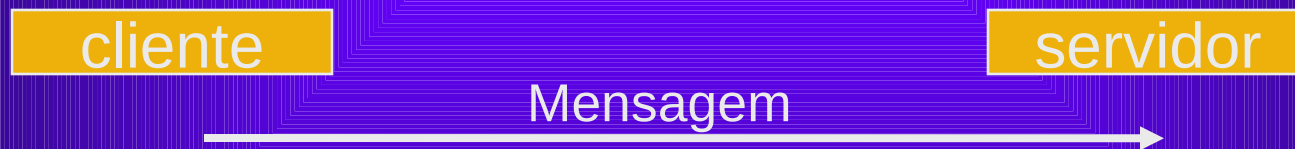
- **Semântica das chamadas:**
 - Oque ocorre quando há falhas?
- **4 semânticas gerais são possíveis (3 são usadas para RPCs):**
 - Possivelmente recebida (maybe);
 - Pelo-menos-uma-vez (at-least-once);
 - No-máximo-uma-vez (at-most-once);
 - Exatamente-uma-vez (exactly-once)

RPC – Semântica das Chamadas

- Semântica das chamadas:
 - Oque ocorre quando há falhas?
- 4 semânticas gerais são possíveis (3 são usadas para RPCs):
 - Possivelmente recebida;
 - **Pelo-menos-uma-vez (at-least-once);**
 - **No-máximo-uma-vez (at-most-once);**
 - **Exatamente-uma-vez (exactly-once)**

RPC – Semântica das Chamadas

- **Possivelmente recebida:**
 - Não utilizada em RPC!
 - Pode funcionar com o uso de LANs (alta confiabilidade).



Não há espera:

operação assíncrona

RPC – Semântica das Chamadas

- **Pelo-menos-uma-vez (at-least-once):**
 - **Cliente envia requisição;**
 - **Espera resposta durante intervalo pré-definido;**
 - **Se não receber re-transmite a requisição;**
 - **Servidor pode executar mais de uma vez**
 - **Operações devem ser “idempotentes”**

RPC – Semântica das Chamadas

- **Pelo-menos-uma-vez (at-least-once):**
 - Cliente envia requisição;
 - Espera resposta durante intervalo pré-definido;
 - Se não receber re-transmite a requisição;
 - Servidor pode executar mais de uma vez
 - Operações devem ser “idempotentes”;
 - Idempotente \Rightarrow podem ser repetidas.
 - Mecanismo **COMPLICADO!**

RPC – Semântica das Chamadas

- **No-máximo-uma-vez (at-most-once):**
 - **Cliente envia requisição;**
 - **Espera resposta durante intervalo pré-definido;**
 - **Se não receber re-transmite a requisição;**
 - **Servidor (se não falhar) executará apenas uma vez**
 - **Requisições duplicadas são detectadas;**
 - **Protocolo mais complexo.**

RPC – Semântica das Chamadas

- **No-máximo-uma-vez (at-most-once):**
 - Cliente envia requisição;
 - Espera resposta durante intervalo pré-definido;
 - Se não receber re-transmite a requisição;
 - Servidor (se não falhar) executará apenas uma vez
 - Requisições duplicadas são detectadas;
 - Protocolo mais complexo.
 - Mecanismo **ACEITÁVEL!**

RPC – Semântica das Chamadas

- **Exatamente-uma-vez (exactly-once):**
 - **Cliente envia requisição;**
 - **Espera resposta durante intervalo pré-definido;**
 - **Se não receber re-transmite a requisição;**
 - **Servidor executará exatamente uma única vez**
 - **Ações atômicas devem ser implementadas;**
 - **Servidor baseado em transações atômicas;**
 - **Procedimento complexo.**

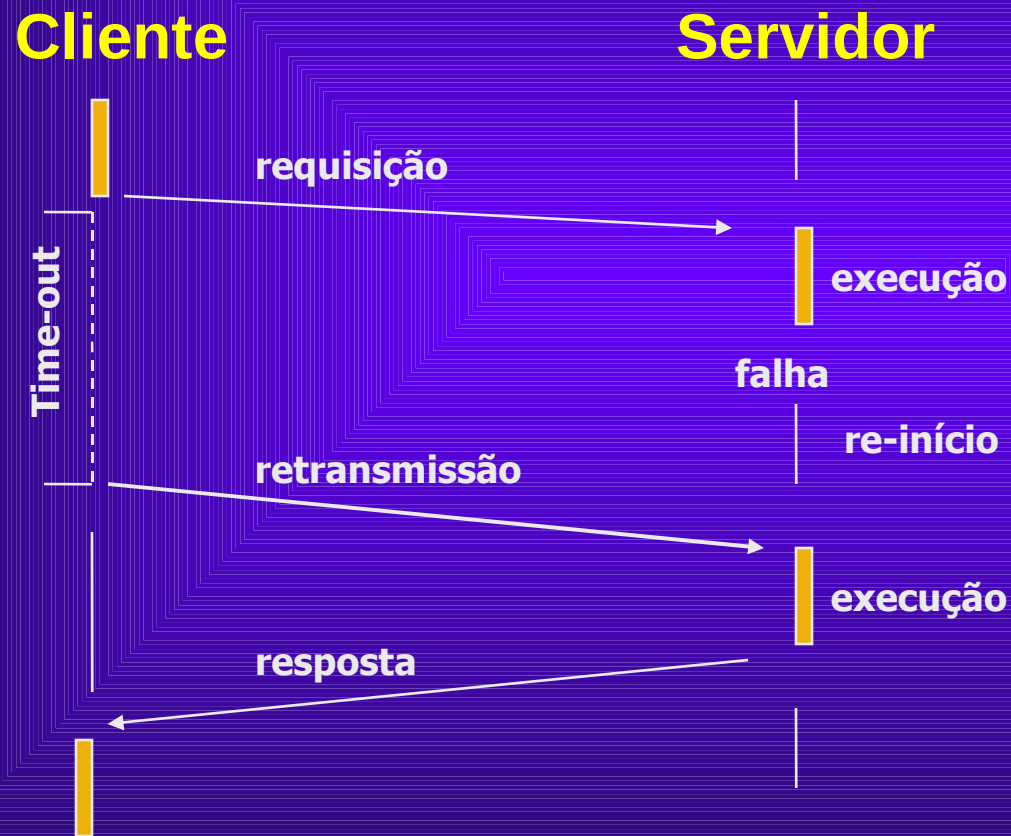
RPC – Semântica das Chamadas

- **Exatamente-uma-vez (exactly-once):**
 - Cliente envia requisição;
 - Espera resposta durante intervalo pré-definido;
 - Se não receber re-transmite a requisição;
 - Servidor executará exatamente uma única vez
 - Ações atômicas devem ser implementadas;
 - Servidor baseado em transações atômicas;
 - Procedimento complexo.
 - Mecanismo **MAIS FORTE! (e altamente complexo)**

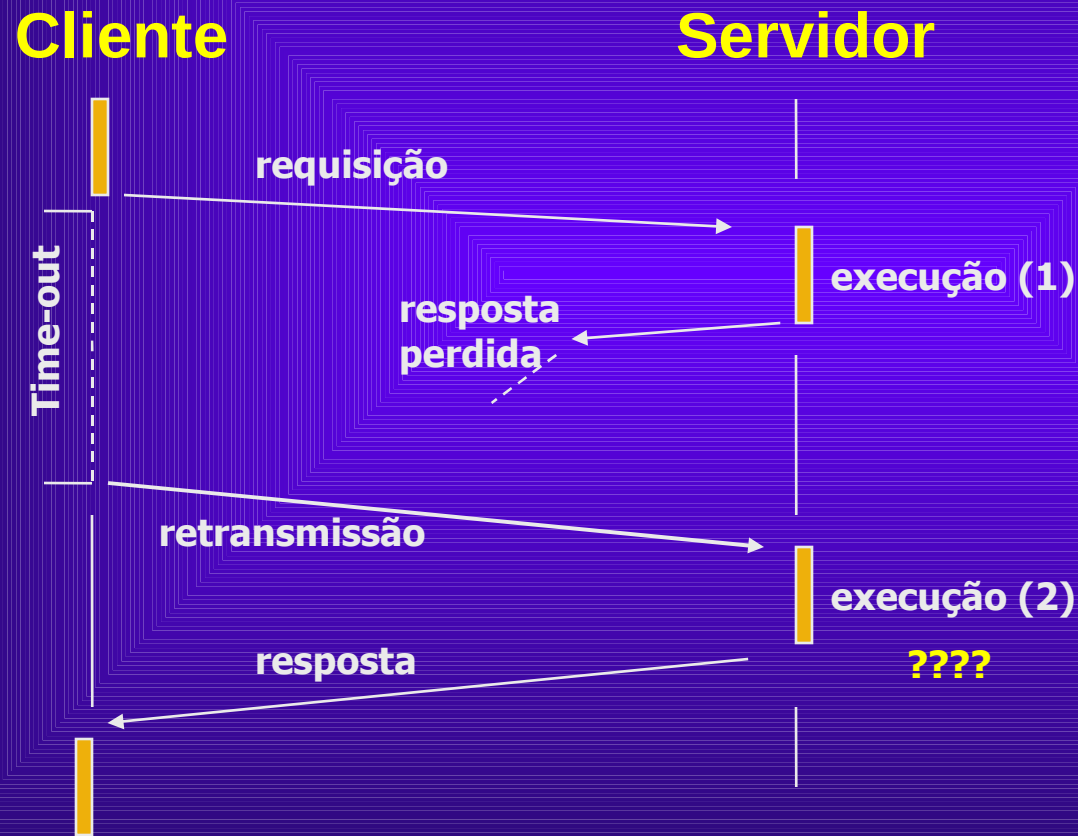
RPC – Modelos de falhas (a)



RPC – Modelos de falhas (b)



RPC – Modelos de falhas (c)



RPC – Modelos de falhas (d)



RPC – Binding

➤ Preparação da RPC:

- **Endereço do servidor:**
 - fixo no programa;
 - fixo em um arquivo;
 - fornecido manualmente;
 - “binding service” ⇒ “serviço de preparação”
 - endereço fixo (?)
 - servidor de nomes (?)

RPC – Binding

➤ **Preparação da RPC:**

- **Binding Service:**
 - servidores registram serviços no BS (exportação);
 - clientes requisitam endereço (importação);
 - servidores avisam quando vão parar (remoção);
 - Em caso de falhas clientes detectam ausência.

RPC – Binding

➤ Gerenciamento de Servidores

- Diversas instâncias em uma ou mais máquinas;
- Qual utilizar:
 - Escolha arbitrária (Binder)
 - Gerente de servidores:
 - endereço do gerente fornecido pelo Binder;
 - Cliente requisita processo servidor \Rightarrow nova instância;
 - acesso exclusivo.
 - Banco de processos servidores:
 - endereço do gerente fornecido pelo Binder;
 - Cliente requisita processo servidor;
 - Gerente escolhe um que esteja livre
 - Acesso exclusivo ao cliente.
 - Cada chamada ao servidor utiliza uma nova instância.

RPC – Protocolo de Suporte

- **Protocolo de suporte: camada de transporte;**
- **RPC implementado acima da camada de transporte;**
 - Normalmente **NO-MÁXIMO-UMA-VEZ;**
 - Mecanismos de gerenciamento devem ser construídos;
- **Exemplos de protocolos:**
 - UDP (não confiável)
 - TCP (confiável)

Exemplos de RPC

- **RPC – Sistema TRICE**
 - Universidade de Southampton – U.K.
- **Dois mecanismos implementados:**
 - SSP – semântica “PELO-MENOS-UMA-VEZ”
 - MSP – semântica “NO-MÁXIMO-UMA-VEZ”
 - Ambos utilizam o P-Service;
 - P-Service → datagrama (equivalente ao UDP)

SSP - Single Shot Protocol

- **SSP – Single Shot Protocol**
 - **Binder (preparador) → Servidor de NOMES**
tabelas estáticas;
 - **Protocolo de Suporte → P-Service;**
 - **P-Service → datagrama (equivalente ao UDP).**

SSP - Single Shot Protocol

- Mensagens SSP

cliente

servidor

Requisição (parâmetros)

Resposta (resultados)

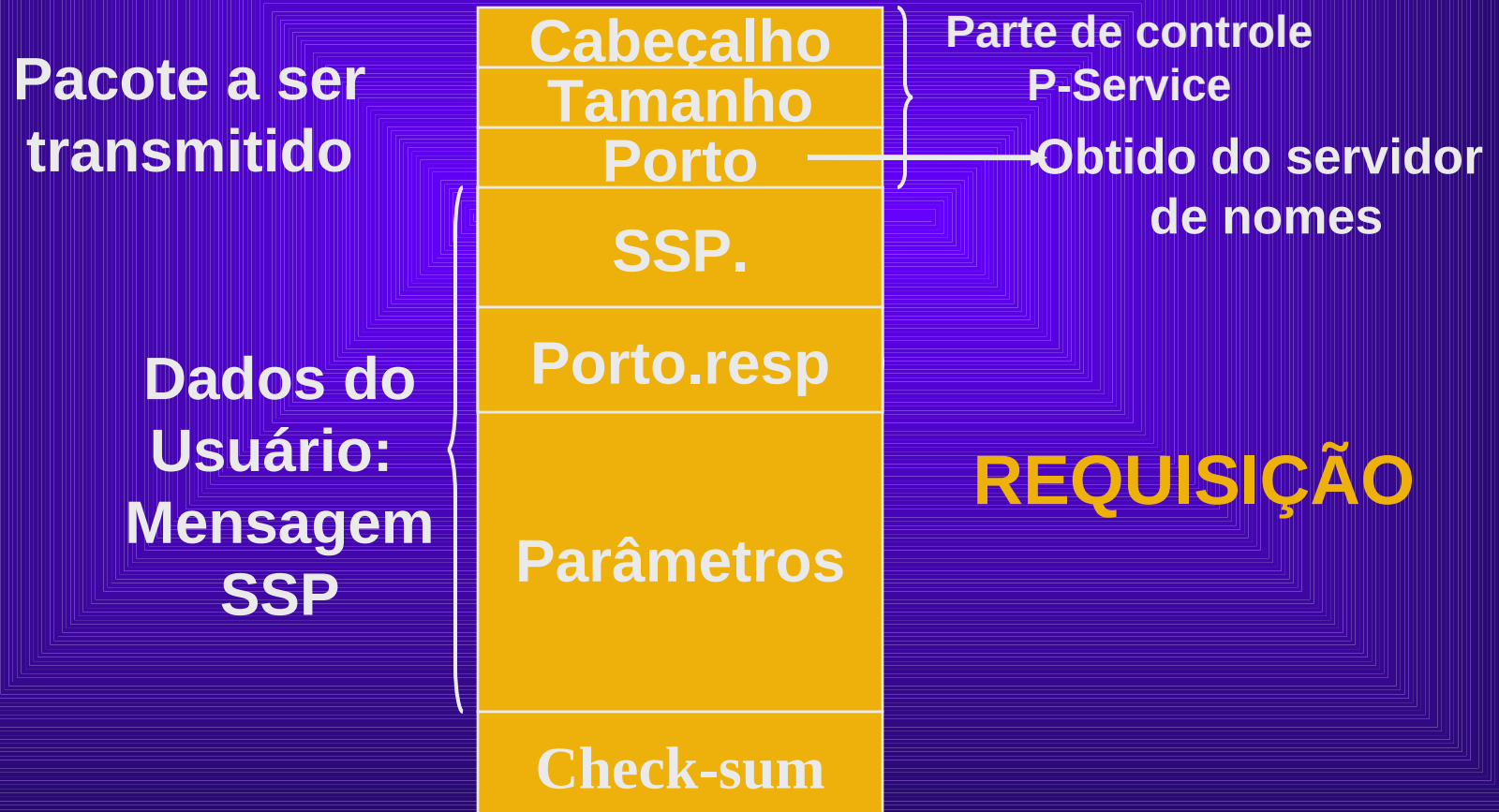
SSP - Single Shot Protocol

➤ Mensagens básicas SSP



SSP - Single Shot Protocol

➤ Mensagens básicas SSP



SSP - Single Shot Protocol

➤ Mensagens básicas SSP



SSP - Single Shot Protocol

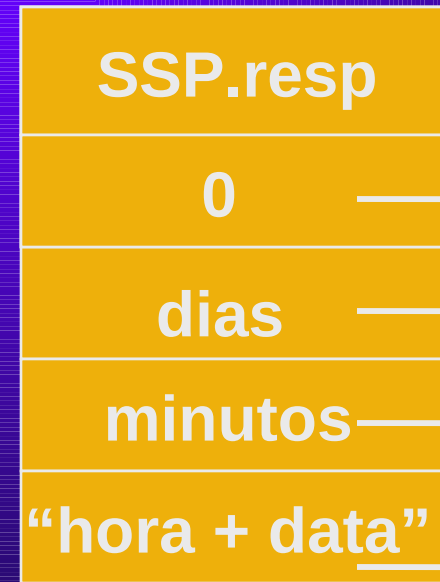
- Mensagens para obter hora no sistema TRICE

**CLIENTE
(E.T.)**



↓
Um único
parâmetro

**SERVIDOR
Serv. Tempo**



→ OK!

→ No. dias...

→ Desde...

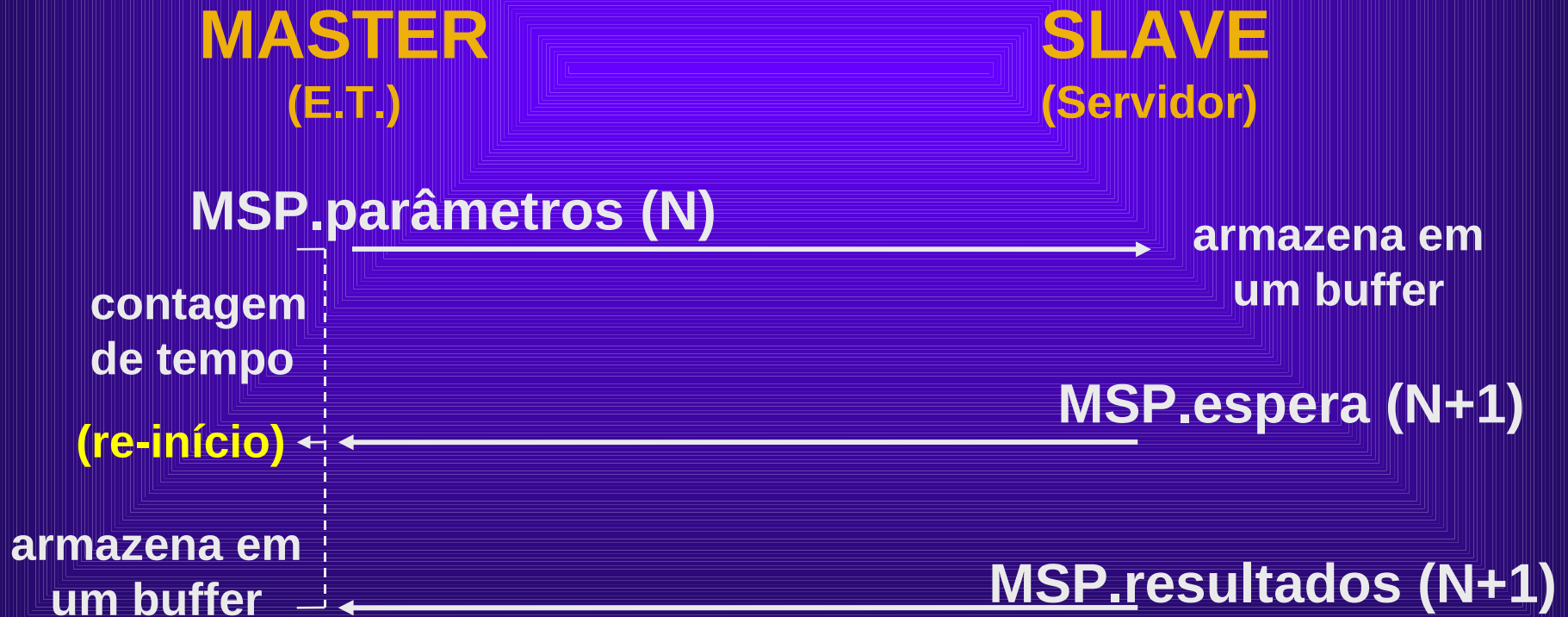
→ String...

MSP – Master-Slave Protocol

- Definido na mesma camada que SSP;
- Utiliza o P-Service como suporte;
- Cria um circuito virtual (confiável);
- Implementa semântica NO-MÁXIMO-UMA-VEZ.

MSP – Master-Slave Protocol

- Mensagens MSP



MSP – Master-Slave Protocol

- **Garantia da semântica NO-MÁXIMO-UMA-VEZ:**
 - **(1) E.T. não recebe a resposta:**
 - Time-out;
 - Requisição é repetida.
 - **(2) Mensagens fora de ordem:**
 - Numeração de mensagens;
 - Requisição e resposta aceitas somente em ordem.
 - **(3) Mensagens em duplicata:**
 - Detectadas pelo controle de numeração;
 - Armazenamento de estado;
 - Mensagem anterior é resgatada no servidor.

Protocolos no Sistema TRICE

- Para entender onde se encaixam o SSP e o MSP





FIM!