Trabalhos de Redes II

- Implementação de um **App de *Instant Messaging*** (IM)
  - ✓ *Instant Messaging* é a categoria de aplicativos de trocas de mensagens instantâneas, como WhatZap, Telegram, *chats* em geral, etc.
  - ✓ Utilização de GUI
    - GUI do cliente? GUI do servidor?
    - Por se tratar de *software* amplamente utilizado, a GUI deste tipo de aplicativo é praticamente um padrão
  - ✓ Utilização de endereços de rede (IP)
    - O trabalho final deverá executar em um **ambiente de rede real**, e o ambiente de testes será o laboratório de redes da uesb
    - Contudo, para efeito de desenvolvimento/teste lembre-se que você tem a disposição a classe de endereços especiais *localhost* (127.0.0.0)
    - Se não lembra, revisar Tanenbaum, 4ª Ed., Cap 5.6.2, Figura 5.56

Trabalhos de Redes II

➪ Implementação de um **App de *Instant Messaging*** (IM)

✓ Utilização de *sockets*

- A comunicação entre os *hosts* será feita através da utilização de *sockets* em Java, conforme visto em sala de aula
- Serão utilizados **sockets TCP (**mensagens do tipo JOIN e LEAVE) e **sockets UDP (**mensagens do tipo SEND)
- Veja APDUs abaixo

✓ Utilização de APDUs

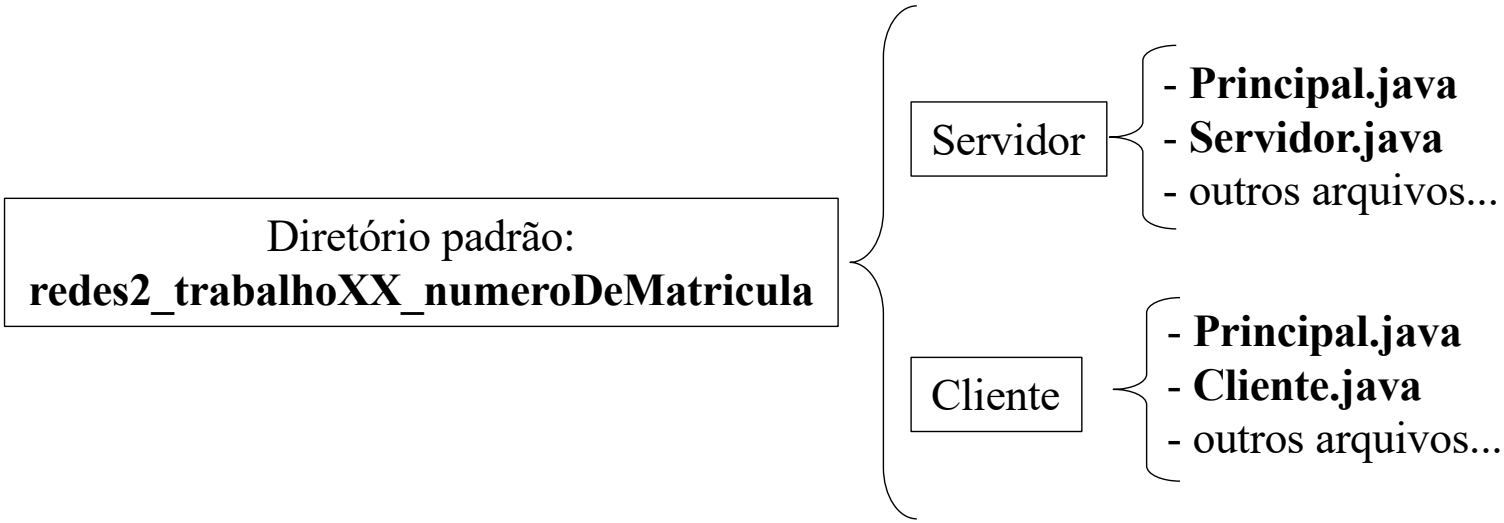
- Toda aplicação de rede tem o seu conjunto próprio de APDUs (unidade de dados do protocolo da aplicação)
- Isso é feito no momento da especificação do *software* e norteará o funcionamento lógico da aplicação
- Neste trabalho, esta tarefa **já foi feita pelo professor** e sua implementação deverá conter **3 APDUs: JOIN, LEAVE e SEND**
  - Funcionamento conforme vista em aula
  - Mais detalhes nos próximos *slides*

Trabalhos de Redes II

➪ Implementação de um **App de *Instant Messaging*** (IM)

✓ Utilização de **arquitetura Cliente/Servidor**

- Como **clientes e servidores possuem códigos distintos**, a estrutura de diretórios e sub-diretórios abaixo deverá ser adotada para entrega do trabalho
- OBS: estrutura semelhante já foi utilizada na disciplina de programação concorrente no trabalho de *fork/threads*

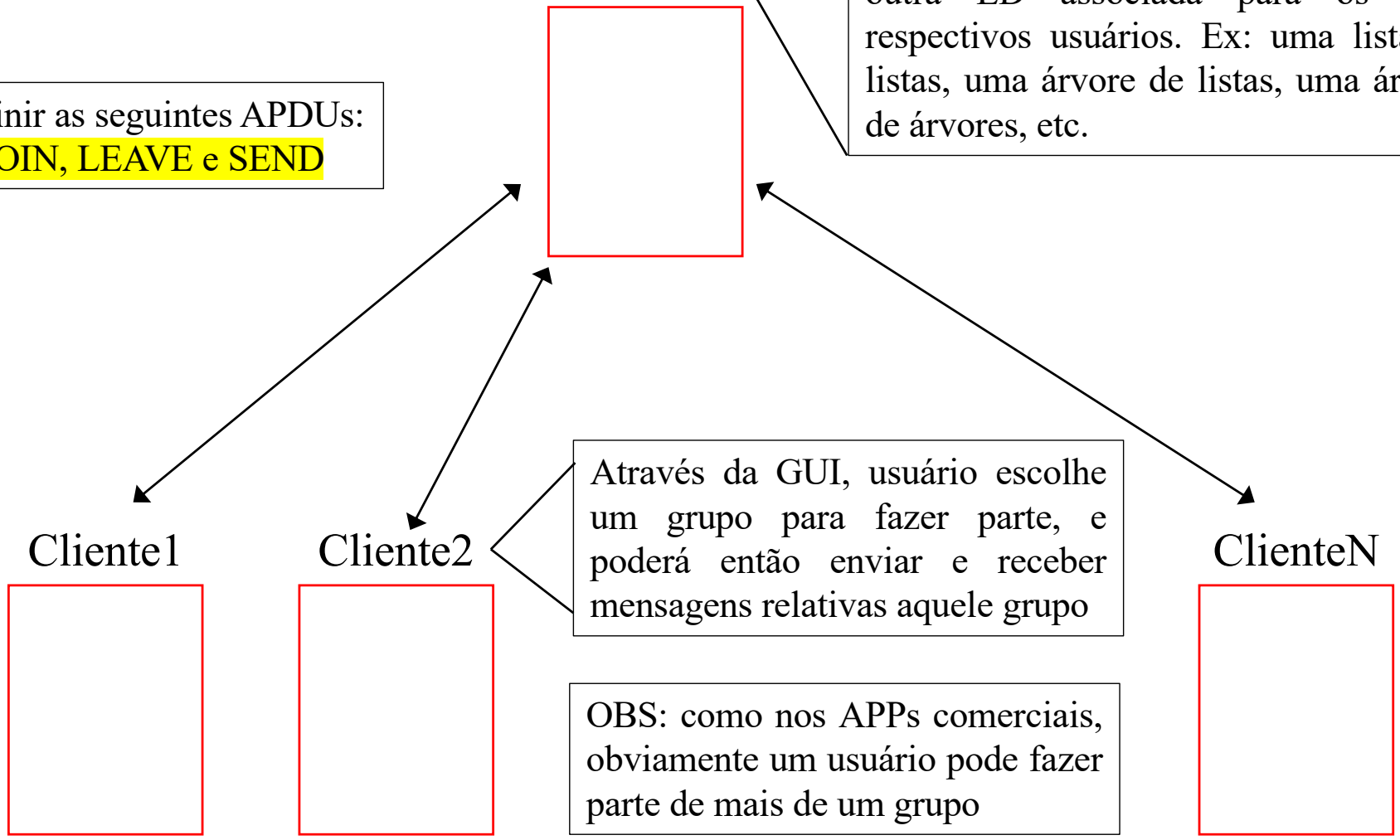


Trabalhos de Redes II

Definir as seguintes APDUs:
JOIN, LEAVE e SEND

Servidor

Gerencia os grupos e os usuários por grupo, ou seja, uma ED para os grupos e outra ED associada para os seus respectivos usuários. Ex: uma lista de listas, uma árvore de listas, uma árvore de árvores, etc.



Cliente1

Cliente2

ClienteN

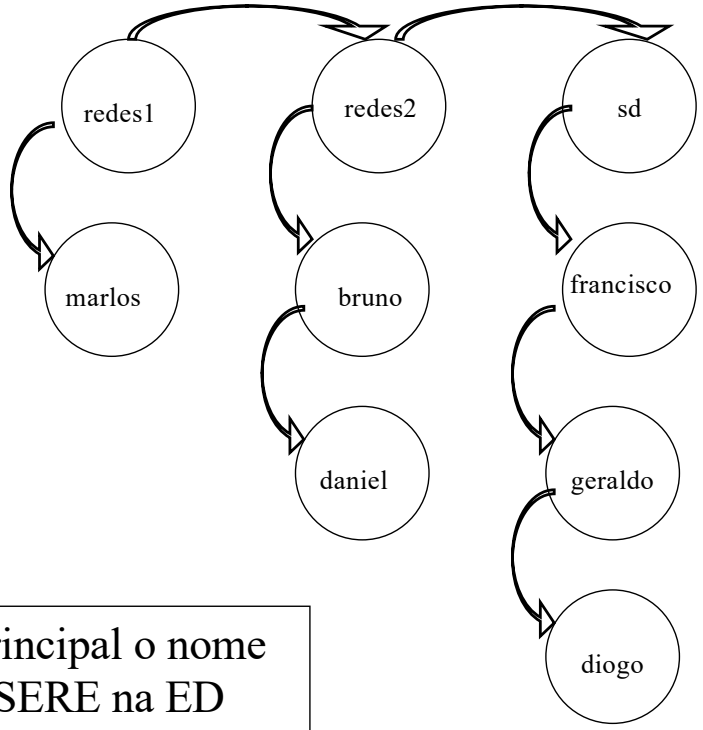
Através da GUI, usuário escolhe um grupo para fazer parte, e poderá então enviar e receber mensagens relativas aquele grupo

OBS: como nos APPs comerciais, obviamente um usuário pode fazer parte de mais de um grupo

Trabalhos de Redes II

APDU JOIN possui 2 parâmetros:
Qual é o **nome do usuário**[1] que deseja participar de qual **grupo**[2]

Servidor



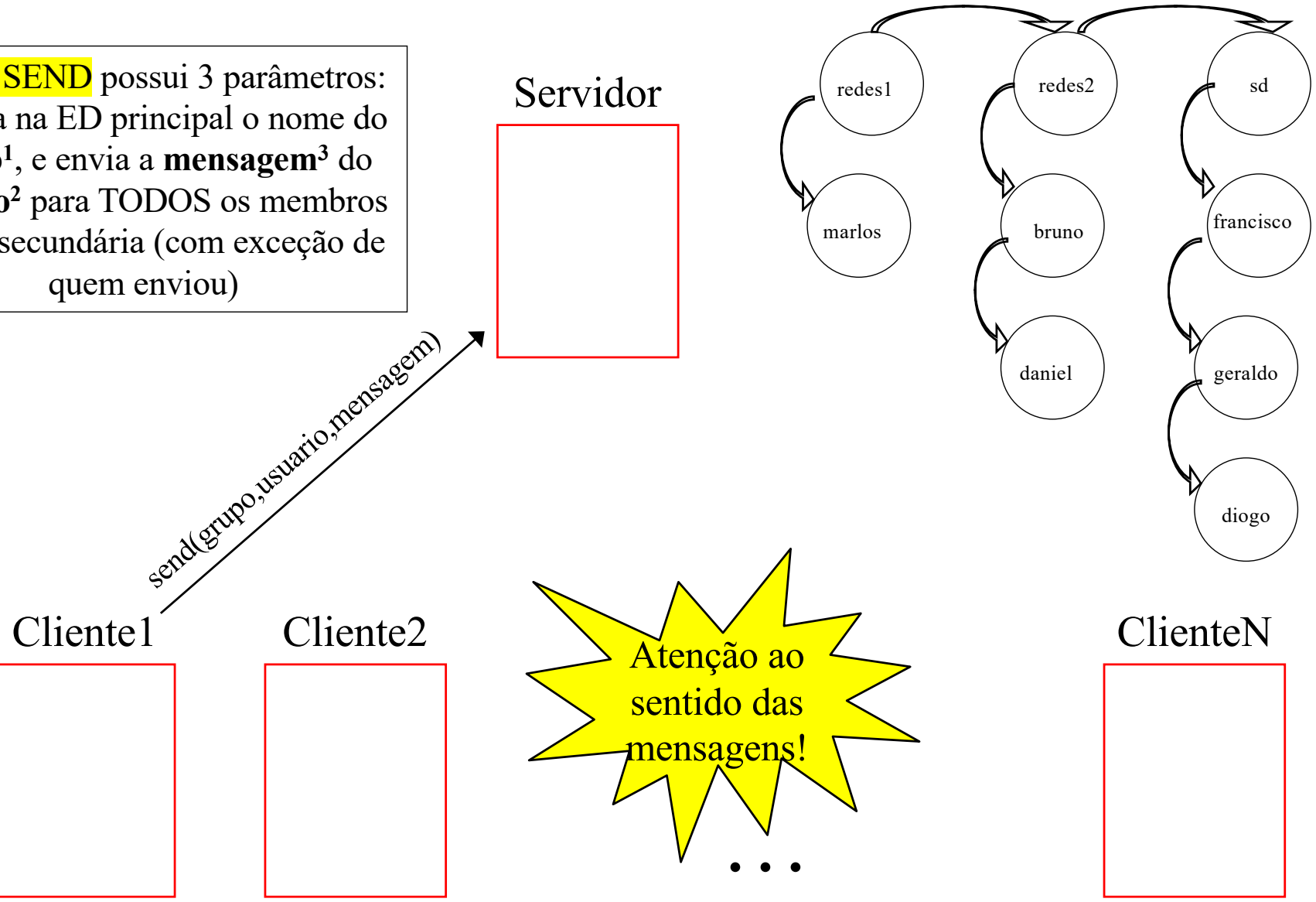
join(grupo,usuário)

Procura na ED principal o nome do grupo, e INSERE na ED secundária o nome do usuário

Cliente1 Cliente2 ... ClienteN

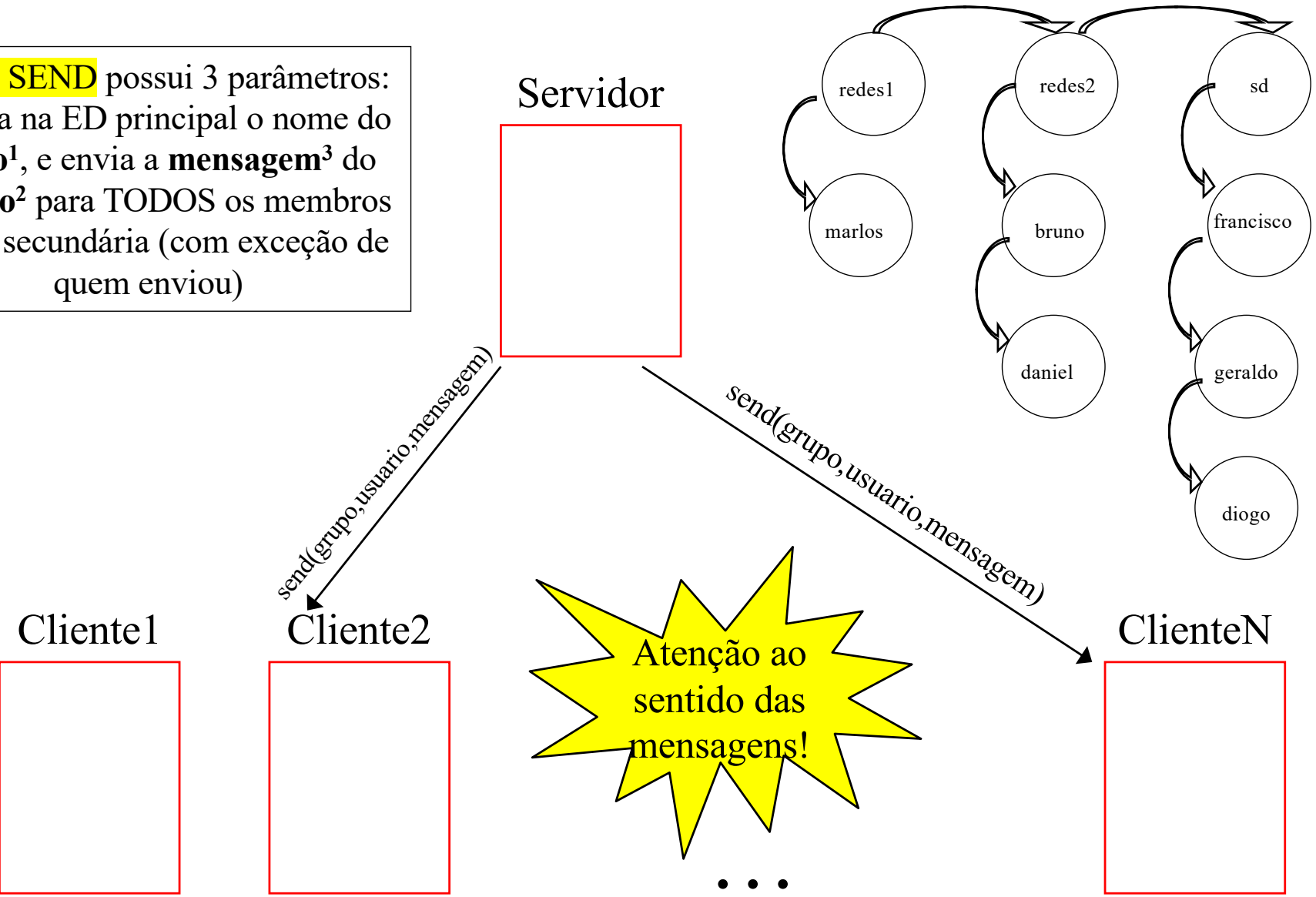
Trabalhos de Redes II

APDU SEND possui 3 parâmetros: Procura na ED principal o nome do **grupo**[1], e envia a **mensagem**[3] do **usuário**[2] para TODOS os membros na ED secundária (com exceção de quem enviou)



Trabalhos de Redes II

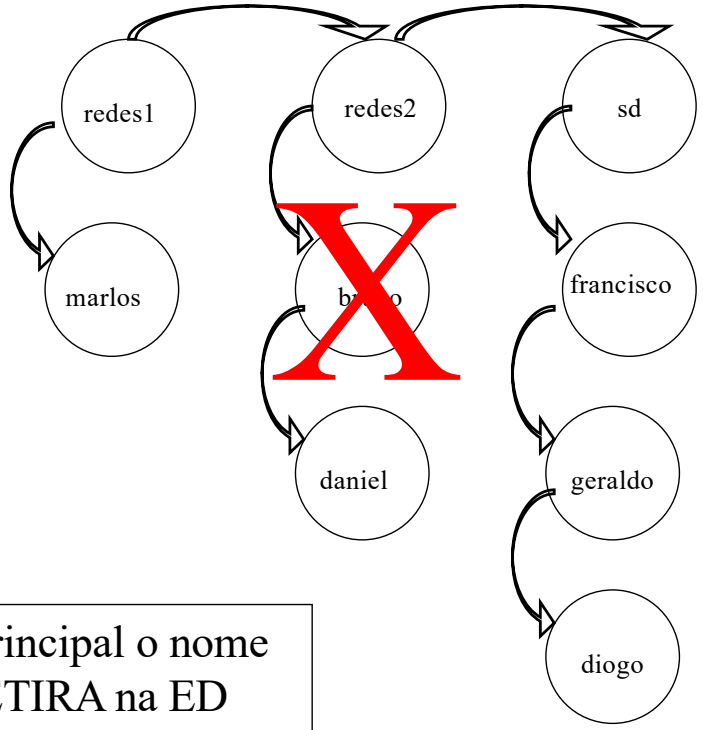
APDU SEND possui 3 parâmetros: Procura na ED principal o nome do **grupo**[1], e envia a **mensagem**[3] do **usuário**[2] para TODOS os membros na ED secundária (com exceção de quem enviou)



Trabalhos de Redes II

APDU LEAVE possui 2 parâmetros: Qual é o **nome do usuário**[1] que deseja SAIR de qual **grupo**[2]

Servidor



leave(grupo,usuário)

Procura na ED principal o nome do grupo, e RETIRA na ED secundária o nome do usuário

Cliente1 Cliente2 ... ClienteN