

Sistemas Distribuídos

Arquiteturas, Protocolos e Comunicação

Prof. Rodrigo Campiolo

08/08/20

Tópicos

- Arquiteturas de sistema
 - Exemplos de arquiteturas
 - Ferramentas
 - Atividade de projeto de arquitetura
- Especificação de protocolos
 - Protocolos textuais (*plain text*)
 - Protocolos binários
- Comunicação por fluxo
 - Sockets e portas
 - Primitivas de comunicação
 - Exemplo

Tópicos

- Comunicação por datagrama
 - Socket e pacotes de datagrama
 - Primitivas de comunicação
 - Exemplo
- Comunicação multicast IP
 - Socket multicast
 - Primitivas de comunicação
 - Exemplo

Arquitectura de Sistemas

- Ejemplos de arquitectura
 - Moodle → Ejemplos → Ejemplos de Arquitectura

Arquitetura de Sistemas

- Ferramentas
 - Drawio: <https://app.diagrams.net/>
 - Lucid Chart: <https://www.lucidchart.com>
 - yEd: <https://www.yworks.com/products/yed>
 - Microsoft Visio

Arquitetura de Sistemas

- Atividades

- Projetar a arquitetura de um sistema de monitoramento de serviços.

Moodle → Materiais → Atividade - Projeto de arquitetura de sistema.

Especificação de Protocolos

- Protocolos textuais
 - formato textual (*plain text*)
 - entendível por humanos
 - mais simples para depurar
 - codificação dos caracteres (ASCII, ...)
 - tamanhos de campos e parâmetros não textuais
 - exemplos: http 1.1, ftp.

No.	Time	Source	Destination	Protocol	Length	Info
→ 3...	18.359...	192.168.1....	200.19.73....	HTTP 5...		GET / HTTP/1.1
← 3...	18.392...	200.19.73....	192.168.1....	HTTP 5...		HTTP/1.1 200 OK (text/html)
▶ Internet Protocol Version 4, Src: 192.168.1.134, Dst: 200.19.73.48 ▶ Transmission Control Protocol, Src Port: 37236, Dst Port: 80, Seq: 1, Ack: 1, Len: 476 ▶ Hypertext Transfer Protocol						
▶ GET / HTTP/1.1\r\n Host: www.utfpr.edu.br\r\n Connection: keep-alive\r\n Upgrade-Insecure-Requests: 1\r\n User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ch Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,* Accept-Encoding: gzip, deflate\r\n Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7\r\n ▶ Cookie: _ga=GA1.3.1853717048.1582064067\r\n \r\n						
0040 42 7e 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 B~GET / HTTP/1.1 0050 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 75 74 66 70 ··Host: www.utfp 0060 72 2e 65 64 75 2e 62 72 0d 0a 43 6f 6e 6e 65 63 r.edu.br ··Connec 0070 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 tion: ke ep-alive 0080 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 65 63 75 ··Upgrad e-Insecu 0090 72 65 2d 52 65 71 75 65 73 74 73 3a 20 31 0d 0a re-Reque sts: 1· 00a0 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 User-Age nt: Mozi 00b0 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 4c 69 lla/5.0 (X11; Li 00c0 6e 75 78 20 78 38 36 5f 36 34 29 20 41 70 70 6c nux x86_ 64) Appl 00d0 65 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 eWebKit/ 537.36 (00e0 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 6b KHTML, l ike Geck						

Figura: Protocolo textual - HTTP GET para o sítio da UTFPR.

Especificação de Protocolos

- Protocolos binários
 - formato binário
 - cabeçalhos bem definidos (bits ou bytes)
 - tamanho reduzido dos pacotes
 - equivalência de tipos e campos (ex: int)
 - exemplos: dns, rip, udp, tcp

No.	Time	Source	Destination	Protocol	Length	Info
→ 3...	18.311...	192.168.1....	8.8.8.8	DNS	76	Standard query 0xbeb6 A www.utfpr.
← 3...	18.341...	8.8.8.8	192.168.1....	DNS	1...	Standard query response 0xbeb6 A w
User Datagram Protocol, Src Port: 56131, Dst Port: 53						
Domain Name System (query)						
Transaction ID: 0xbeb6						
Flags: 0x0100 Standard query						
0... .. = Response: Message is a query						
.000 0... .. = Opcode: Standard query (0)						
.... ..0. = Truncated: Message is not truncated						
.... ...1 = Recursion desired: Do query recursively						
....0.. = Z: reserved (0)						
....0 = Non-authenticated data: Unacceptable						
Questions: 1						
Answer RRs: 0						
Authority RRs: 0						
Additional RRs: 0						
Queries						
[Response In: 364]						
0000	00 40 77 bb 55 10 64 5a	04 64 60 b4 08 00 45 00	.@w·U·dZ ·d`...E·			
0010	00 3e 13 3c 40 00 40 11	55 35 c0 a8 01 86 08 08	·>·<@·@· U5·.....			
0020	08 08 db 43 00 35 00 2a	50 b0 be b6 01 00 00 01	...C·5·* P·.....			
0030	00 00 00 00 00 00 03 77	77 77 05 75 74 66 70 72w ww·utfpr			
0040	03 65 64 75 02 62 72 00	00 01 00 01	·edu·br·			

Figura: Protocolo binário – Consulta DNS para o sítio da UTFPR.

Especificação de Protocolos

- Questões importantes
 - especificação do cabeçalho
 - especificação das mensagens
 - documentação
 - desempenho x simplicidade
 - verificação formal

Comunicação por Fluxo

- Sockets e Portas

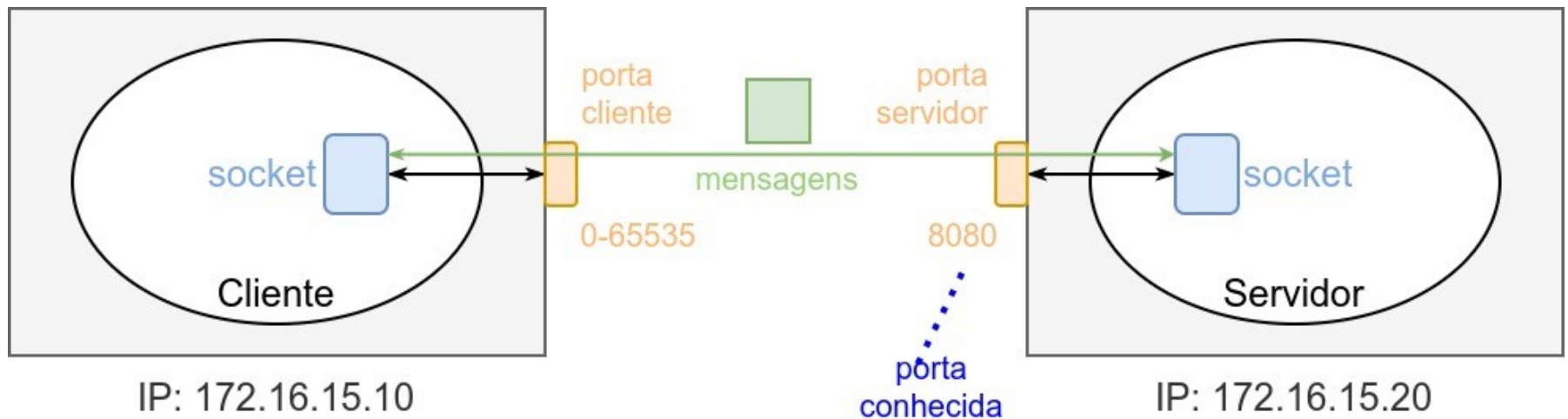
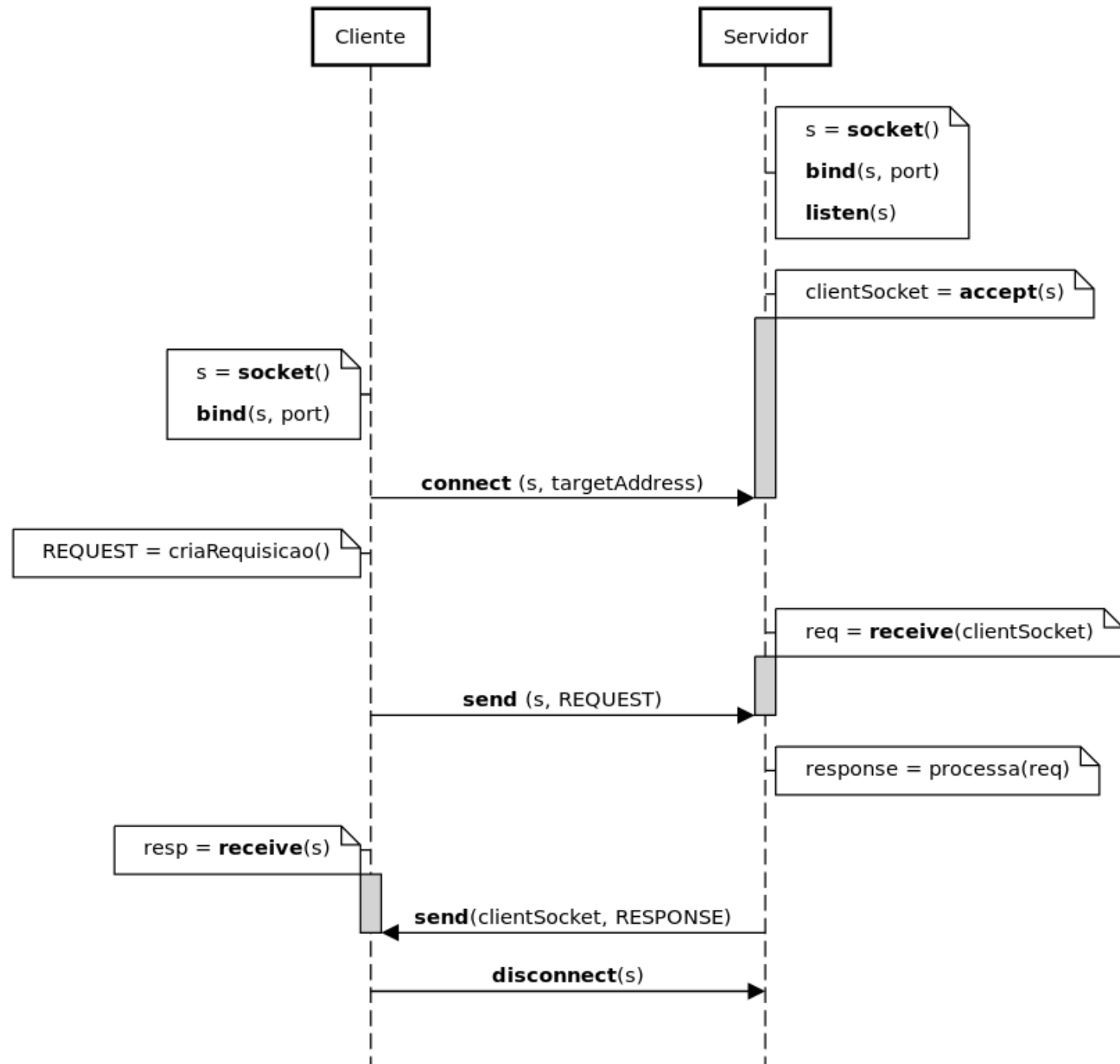


Figura: Comunicação por sockets (Fonte: Coulouris).

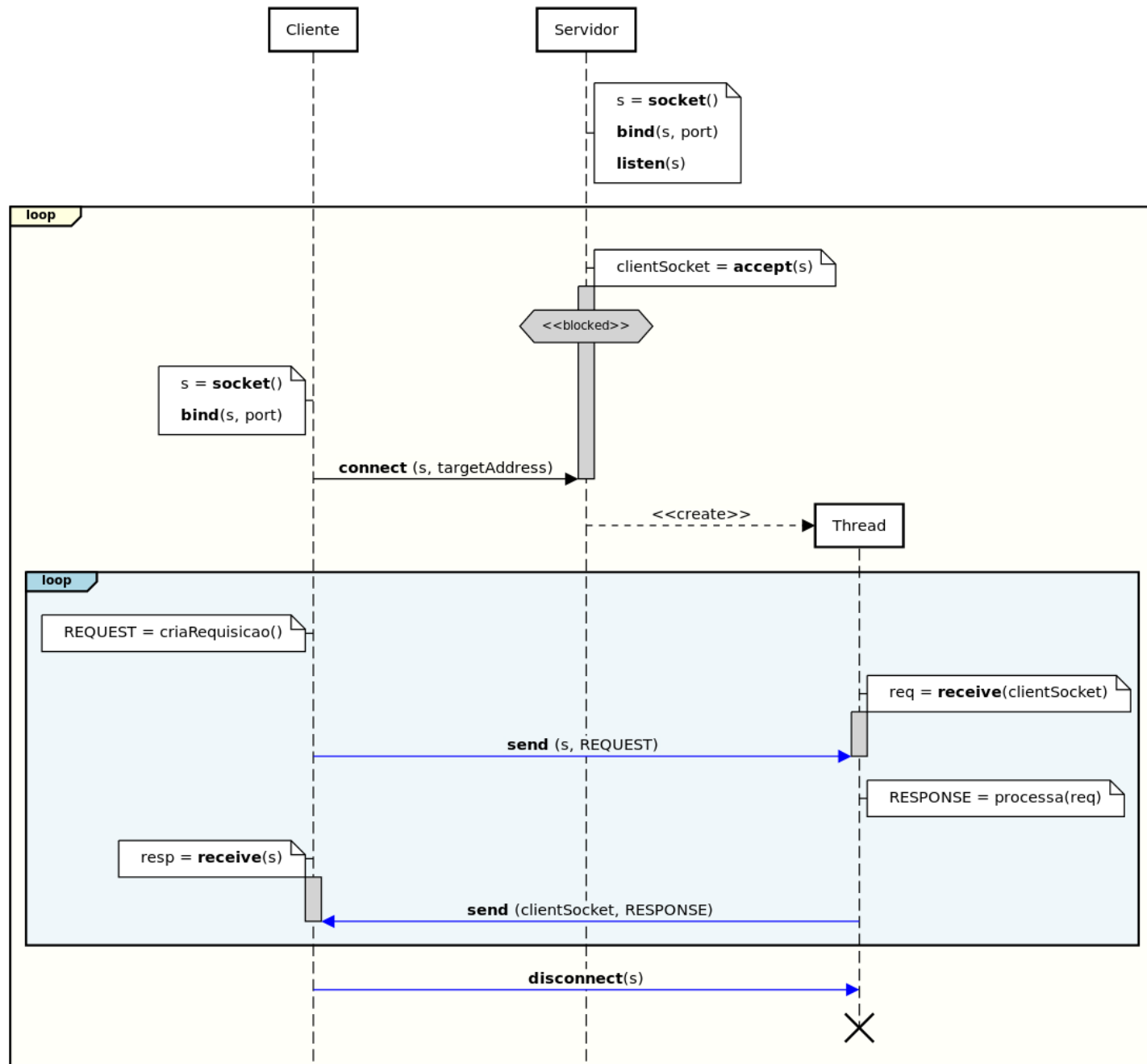
Comunicação por Fluxo

- Primitivas de comunicação por fluxo
 - socket (stream)
 - bind
 - listen
 - accept
 - connect
 - send
 - receive
 - disconnect

Comunicação por fluxo (stream)



Comunicação por fluxo (stream)



Comunicação por Fluxo

- Exemplo de código
 - Moodle → Exemplos → Aula TCP

Comunicação por Fluxo

- Atividades
 - Modifique o código do exemplo para que o cliente e o servidor possam enviar mensagens assincronamente entre si.

Comunicação por Datagrama

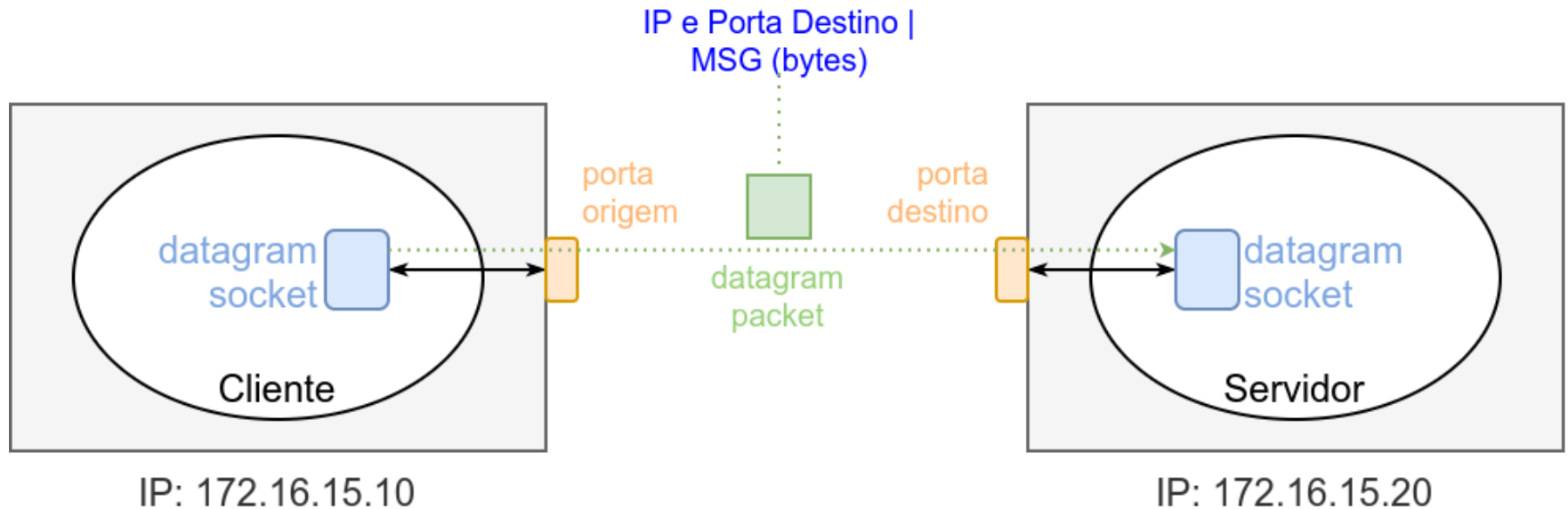
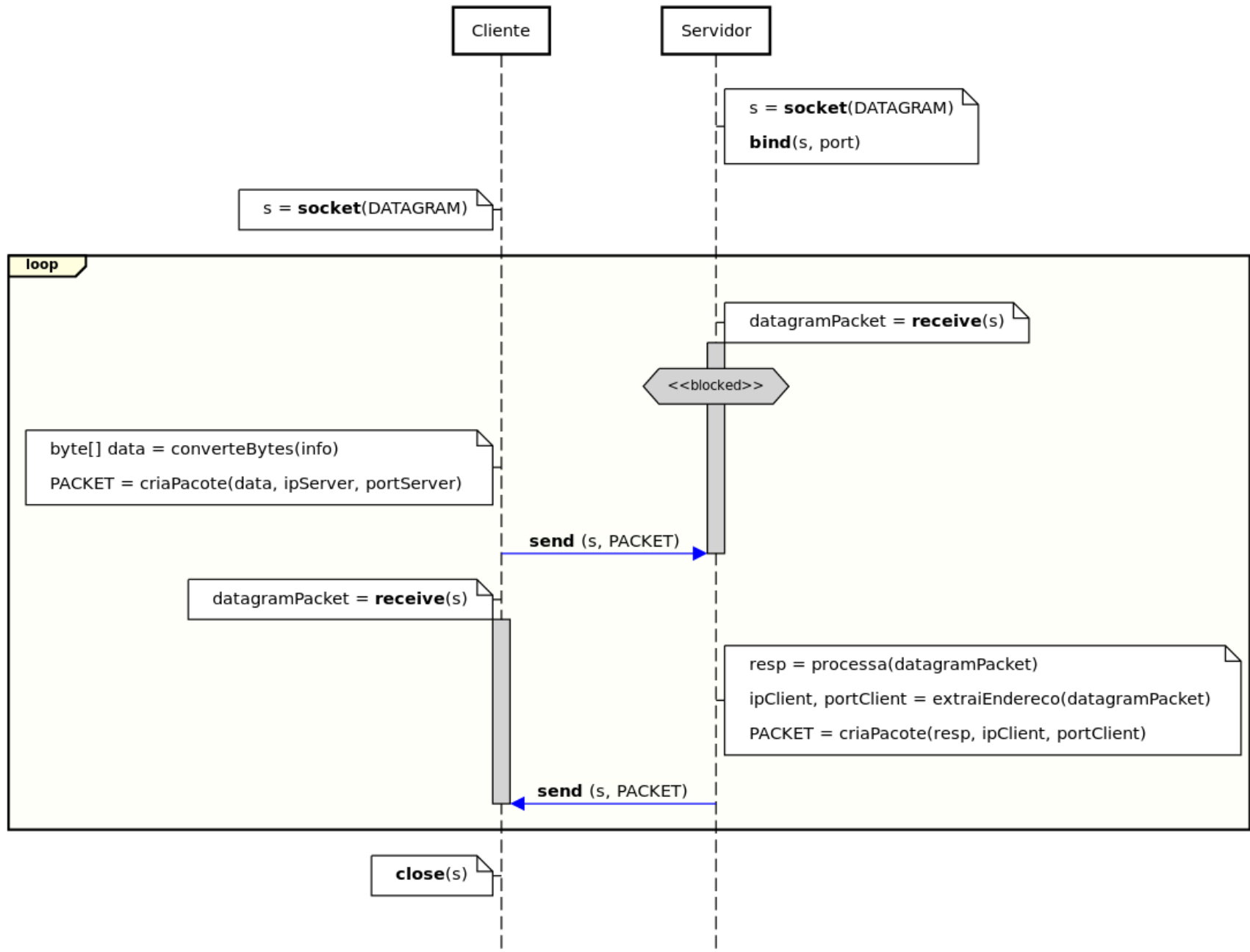


Figura: Comunicação por datagrama.

Comunicação por Datagrama

- Primitivas de comunicação por datagrama
 - socket (datagram)
 - bind
 - send
 - receive
 - close
 - timeout

Comunicação por datagrama



Comunicação por Datagrama

- Exemplo de código
 - Moodle → Exemplos → Aula UDP

Comunicação por Datagrama

- Atividades
 - Modifique o código do exemplo para que o cliente e o servidor possam enviar datagramas assincronamente entre si.

Comunicação Multicast IP

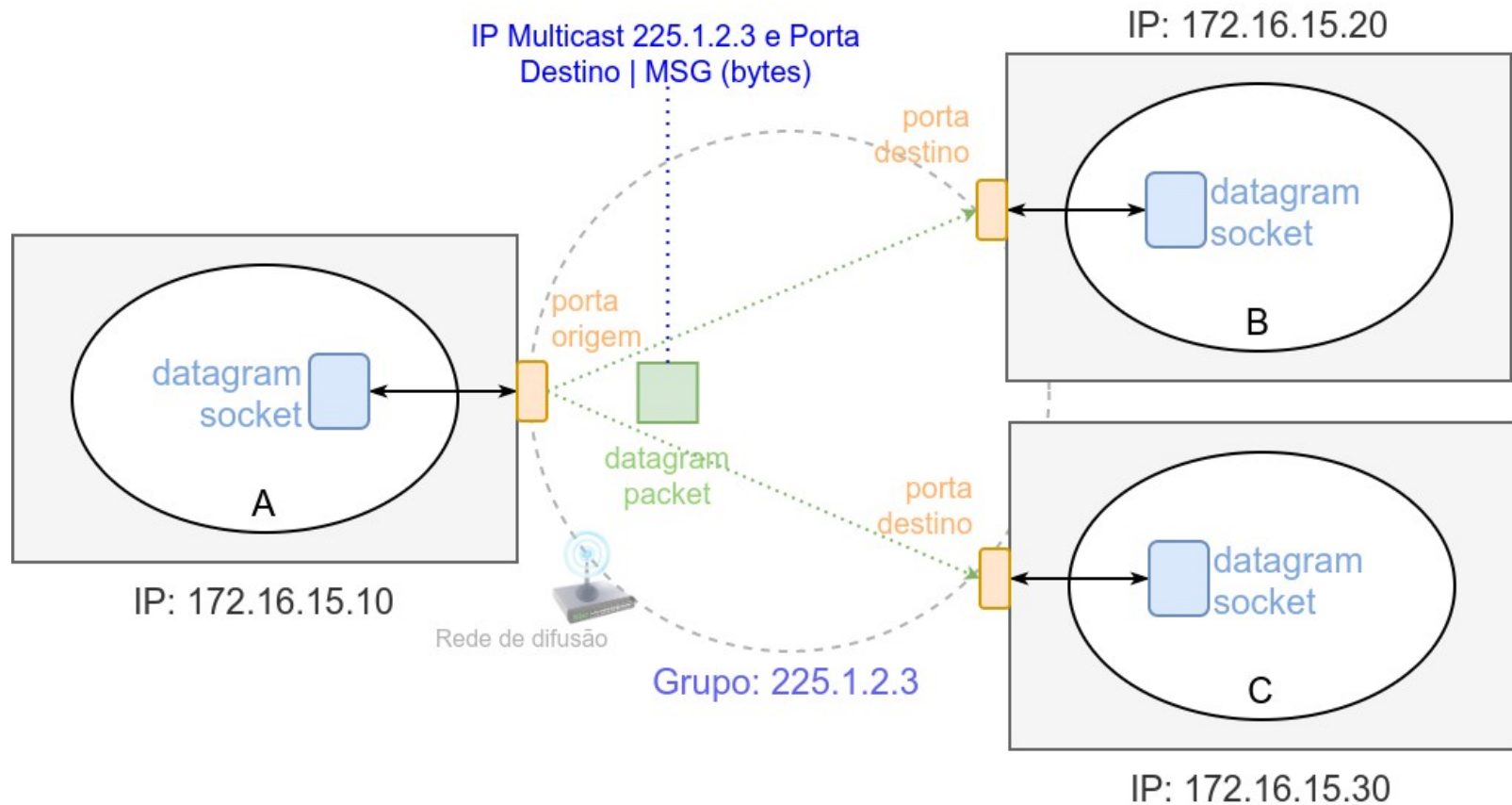
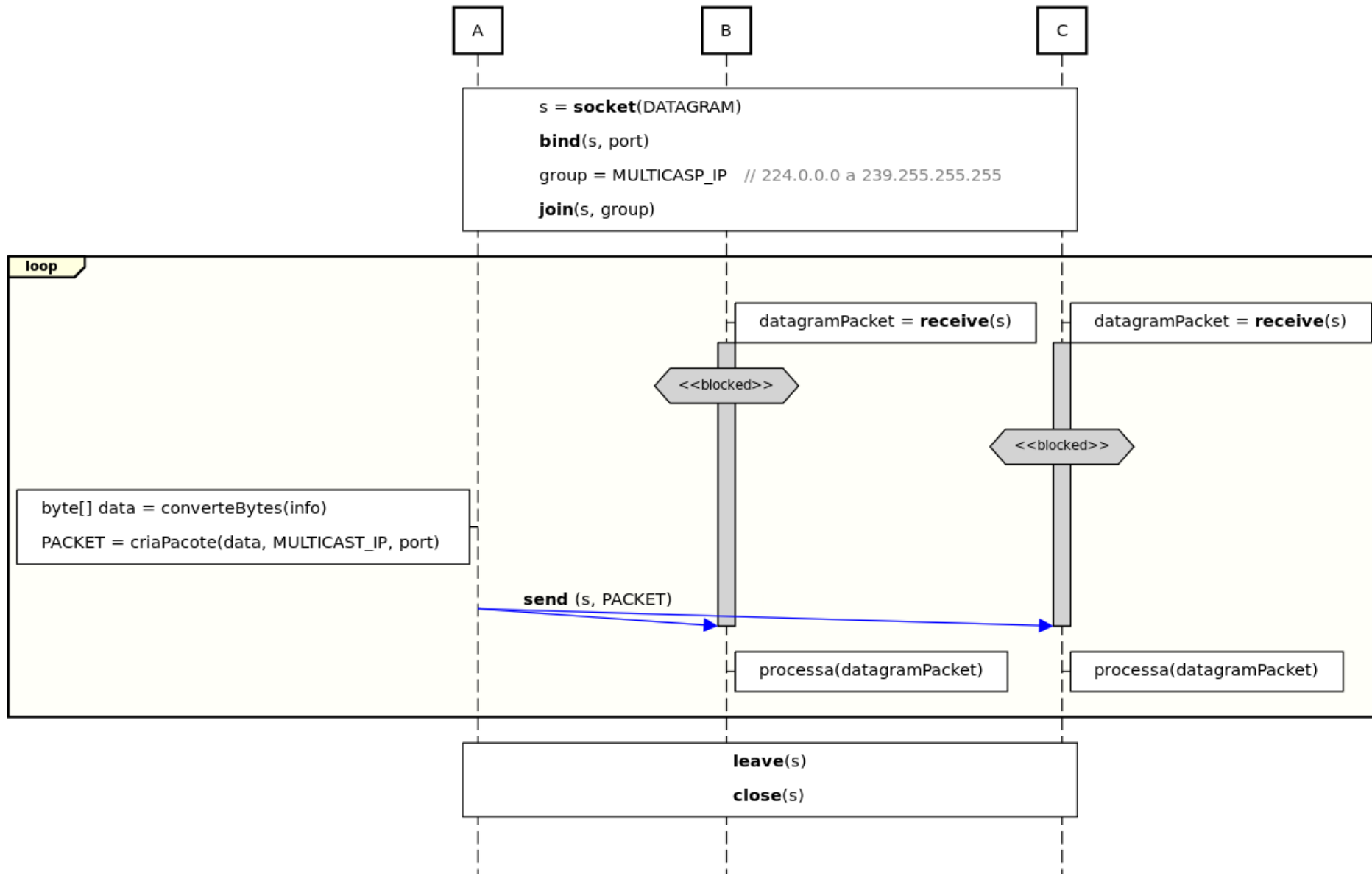


Figura: Comunicação multicast IP.

Comunicação Multicast IP

- Primitivas de comunicação por multicast IP
 - socket (datagram)
 - bind
 - **join**
 - **leave**
 - send
 - receive
 - close

Comunicação Multicast IP



Comunicação Multicast IP

- Exemplo de código
 - Moodle → Exemplos → Aula Multicast IP

Comunicação Multicast IP

- Atividades
 - Modifique o código do exemplo para enviar mensagens com o seguinte formato:

`OPERACAO : OPERACAO_ESPECIFICO`

- `OPERACAO = 1`
 - `OPERACAO_ESPECIFICO = APELIDO:MSG`
- `OPERACAO = 2`
 - `OPERACAO_ESPECIFICO = APELIDO:CODIGO_EMOJI`
 - `CODIGO_EMOJI`
 - `1 = :-)`
 - `2 = :-(`
 - `3 = :-P`

* `OPERACAO` e `CODIGO_EMOJI` devem ocupar 1 byte

Referências

COULOURIS, George F; DOLLIMORE, Jean; KINDBERG, Tim; BLAIR, Gordon. **Sistemas distribuídos: conceitos e projeto**. 5. ed. Porto Alegre: Bookman, 2013.

Apêndice

- Comandos
 - **ss**: ferramenta para visualizar e analisar sockets.
ex: mostrar os sockets tcp (t) para conexão (l) em formato numérico (n) e o processo associado (p).
`# ss -ltnp`
 - **netcat**: lê e escreve dados em conexões tcp e udp.
ex1: criar um servidor tcp na porta 10000
`# netcat -l -p 10000`
ex2: acessar um servidor tcp na porta 10000
`# netcat localhost 10000`
 - **tcpdump**: captura de pacotes.
ex: capturar e mostrar os pacotes tcp enviados na interface loopback na porta 10000.
`# tcpdump -i lo tcp port 10000 -X`

Apêndice

- Comandos (cont.)
 - protocol**: especificação de campos de protocolos.
`# protocol 'message type':16,args:16,'data (0 a 255 bytes)':64 -n`

[illegible]