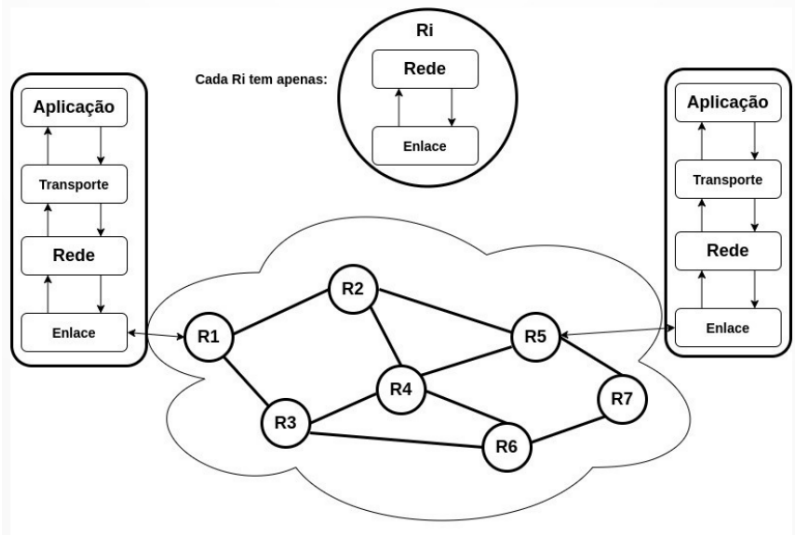
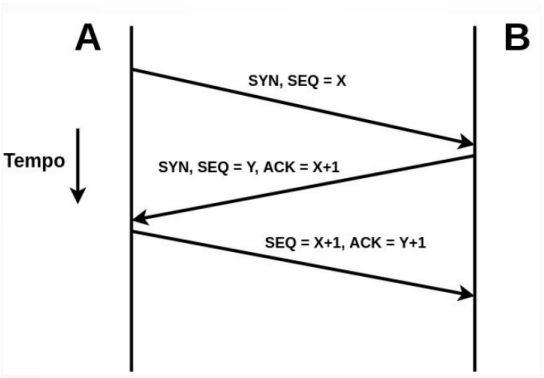
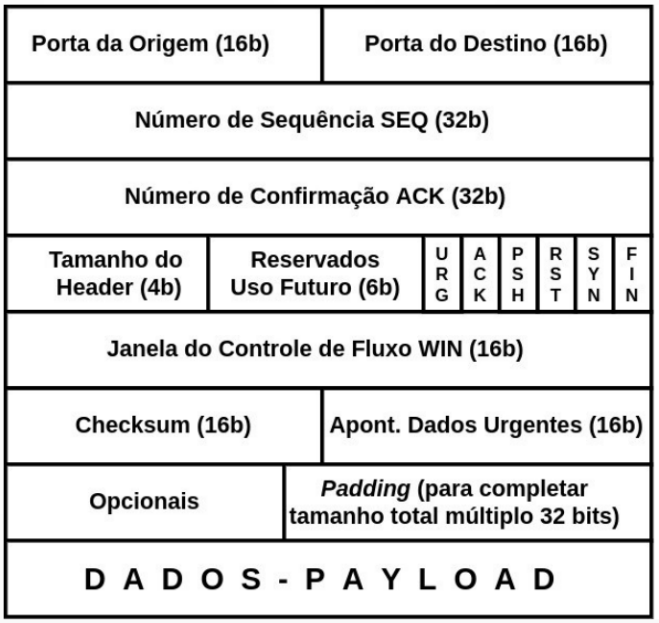
Redes 2

O Protocolo TCP

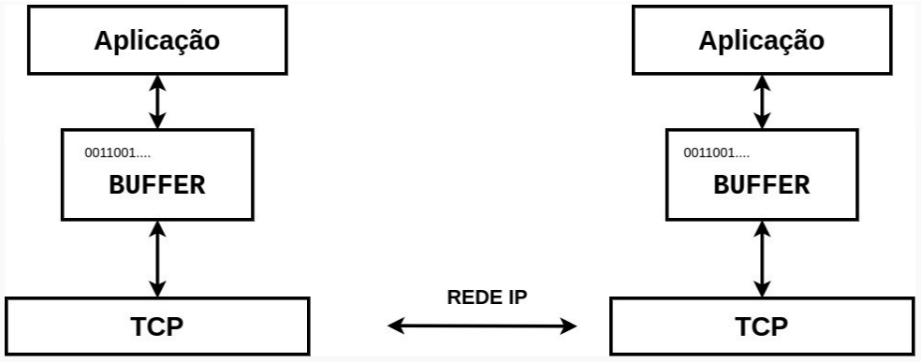
- Comunicação de processos
  - 

  - Para transporte na Internet, podem ser usados dois protocolos
    - UDP: não confiável e não orientado à conexão (sobra trabalho para o programador de aplicação)
    - TCP: confiável e orientado à conexão (resolve os problemas que o IP “deixa passar”)
- TCP
  - Antes de comunicar precisa estabelecer conexão entre dois processos
  - A abertura de conexão do TCP se chama “Three-Way Handshake” (aperto de mão em 3 vias)
  - Three-Way Handshake
    - SYN: um flag do TCP usado para abertura da conexão
    - SEQ: o número de sequência do primeiro byte do segmento (o primeiro byte da conexão tem número de sequência aleatório, o que evita que duas conexões contíguas iniciando em 1, 2, etc., o que faria pacotes perdidos na rede se confundirem)
    - ACK: confirmação de recebimento do TCP
      - Para dizer que recebeu até o byte com SEQ = X, responde ACK = X + 1
      - Confirmações são sempre contínuas, a não ser que opcionais sejam usados
    - 

    - TCP SYN Flooding
        - Um dos ataques mais famosos da história da INternet
        - Baseado na abertura de conexão TCP
        - Faz um número massivo de solicitações, nunca envia o passo 3 para nenhuma
        - Se a gerência de memória é pobre: overflow
- Header do Segmento TCP



    - Porta da Origem e Porta do Destino: usadas na identificação dos processos que se comunicam
    - Número de Sequência(SEQ): número de ordem do primeiro byte de dados do payload, identificando os dados sendo transmitidos (Obs: SEQ do primeiro byte do primeiro segmento da conexão é aleatório)
    - Número de Confirmação(ACK): confirmação de recebimento
        - No TCP, usasse uma técnica chamada piggybacking, onde o mesmo pacote que leva dados de A para B confirma o recebimento de dados de B por A
        - Ou seja, ACK do TCP indica o SEQ do próximo byte esperado, que é o SEQ do último byte recebido + 1
    - Tamanho do header: possui tamanho variável, devido aos opcionais.
        - Servidores comerciais ou de terceiros não vão aceitar seus opcionais
        - Portanto, quase sempre são 20 bytes, em palavras de 4 bytes (32 bits)
    - Reservados para Uso Futuro: dá margem à evolução do protocolo, guardando 6 bits
    - URG: flag que indica que há dados urgentes no segmento
        - O TCP não define o que é urgente, e sim as aplicações
        - Indica onde no campo payload estão os dados urgentes
    - ACK: flag que indica que o segmento leva uma confirmação de recebimento
        - Leva o número de sequência do próximo byte esperado
    - PSH: flag indicando que os bytes deste segmento devem ser entregues imediatamente para a aplicação de destino
        - Uso: mouse remoto

- RST: flag usada pelo TCP em resposta a segmentos “malucos”
  - Exemplo: chega um pacote de uma conexão inexistente
  - Significado: “resete sua conexão, pois me mandou um segmento que não faz sentido”
- SYN: flag ligada no estabelecimento da conexão
- FIN: flag ligada no encerramento da conexão
- Checksum do TCP: usa o mesmo algoritmo do IP, ICMP, UDP
  - Soma grupos de 16 bits em complemento de 1, tira o complemento do resultado.
  - Destinatário inclui o checksum na soma, se der zero está certo
  - Inclui alguns campos do header IP no cálculo
- Controle de fluxo do TCP
  - Conceito básico: a origem não pode mandar mais dados do que o destino consegue receber
  - Cada byte transmitido em cada direção da conexão TCP tem um número de sequência
  - No caso do TCP, o destinatário tem um buffer de recepção de dados
  - 

- Janela do Controle de Fluxo TCP
  - Campo WIN do header do segmento TCP
  - O TCP informa sempre ao outro processo quantos bytes livres têm na sua janela
  - Ou seja, a janela de controle de fluxo do TCP é o número de bytes livres do buffer do receptor
  - Exemplo
    - Considere que o buffer do destino tem tamanho total de 80 bytes, SEQ do primeiro byte = 1
    - Recebe 30 bytes: manda ACK = 31 e WIN = 50
    - Recebe 20 bytes: manda ACK = 51, WIN = 30
    - Recebe 30 bytes: manda ACK = 81, WIN = 0
    - O emissor tem que esperar WIN>0 para continuar enviando
    - Após processar os dados, envia ACK = 81 e WIN = 80
  - Se a rede IP perde um pacote com esse segmento, acontece um deadlock (emissor aguarda segmento com WIN>0 e receptor já mandou um WIN>0)
  - Solução: timers (Temporizadores)
  - O Persistence Timer é usado nesse caso, se não chegar mais dados até um tempo limite, é feito novamente a comunicação com o emissor
  - Outro timer importante é o Keep Alive Timer, usado quando a conexão fica em silêncio por muito tempo

- Controle de congestionamento
    - O controle de fluxo permite à origem estimar a capacidade do destino
    - Entre a origem e o destino, está a rede
    - O controle de congestionamento permite à origem estimar a capacidade da rede
    - Uma janela de congestionamento é constantemente atualizada para refletir esta estimativa (janela = número de bytes)
    - Janela do controle de congestionamento = JCONG
    - Antes de toda transmissão as duas janelas são comparadas: Janela de Controle de Congestionamento e Janela do Controle de Fluxo
    - Só pode transmitir se o menor valor para o número de bytes
    - Não é definido no padrão original, e cada implementação do TCP calcula localmente