Redes de Computadores

Camada Rede (Parte 1)

Algoritmos e protocolos de roteamento

Prof. Renê Pomilio de Oliveira

Slides baseados nas aulas da Profa. Dra. Kalinka Castelo Branco (ICMC/USP)
Prof. Dr. Anderson Chaves Carniel (UTFPR)

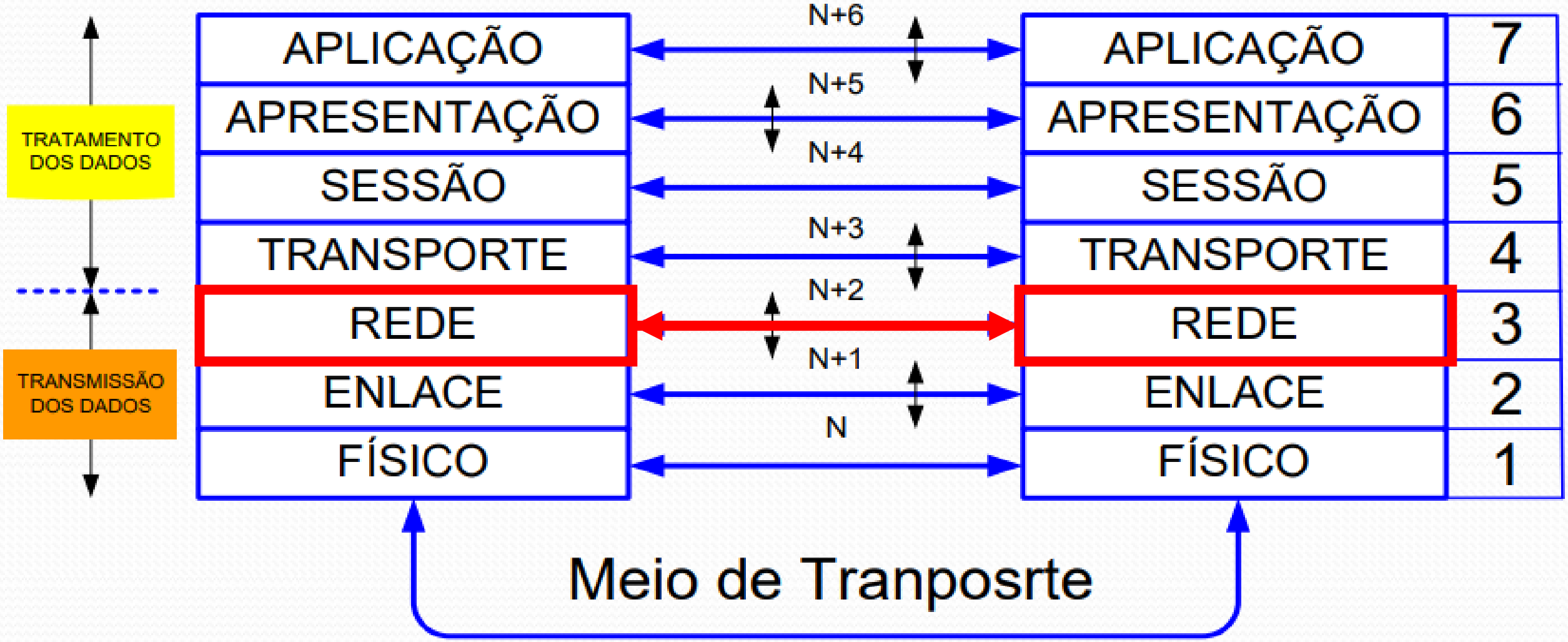
Câmpus Dois Vizinhos

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS DOIS VIZINHOS

MINISTÉRIO DA EDUCAÇÃO

BRASIL
GOVERNO FEDERAL

Entidades da Camada



Revisando...

- Camada Física:
  - Descreve os procedimentos e características mecânicas, elétricas e funcionais. **É responsável pela transmissão de bits de um ponto a outro**.
- Camada de Enlace:
  - Gerencia a transmissão, **detectando e corrigindo erros** na camada física, buscando obter um canal confiável. Separa as **mensagens em quadros**, inserindo aspectos como sincronização, controle de erro e controle de fluxo.

Camada de Rede - Funções

- Camada de Rede:
  - Estabelece, mantém e termina conexões lógicas, é responsável pela tradução de endereços lógicos ou nomes em endereços físicos(roteamento).
  - Provê os meios funcionais para a transmissão de dados orientada ou não-orientada à conexão

Camada de Rede – Funções (Resumo)

- multiplexação
- endereçamento
- mapeamento entre endereços de rede e endereços de enlace
- Roteamento
- estabelecimento e liberação de conexões de rede
- controle de congestionamento

Camada de Rede – Propriedades

- Antes de enviar os dados, é ajustado uma conexão entre as partes.
- Assim que a conexão é estabelecida, as partes negociam os parâmetros, a qualidade e o custo do serviço oferecido.
- A comunicação flui nas duas direções, e os pacotes são enviados em sequência.
- Controle de fluxo é fornecido automaticamente.

Camada de Rede – Funcionalidades

- Transportar pacotes entre os sistemas finais da rede
- A camada de rede deve ter uma entidade em cada sistema final ou roteador da rede
- 3 funções importantes:
  - ✓ Determinação de caminhos.
  - ✓ Comutação.
  - ✓ Estabelecimento de conexão.

Camada de Rede – Determinação de caminhos

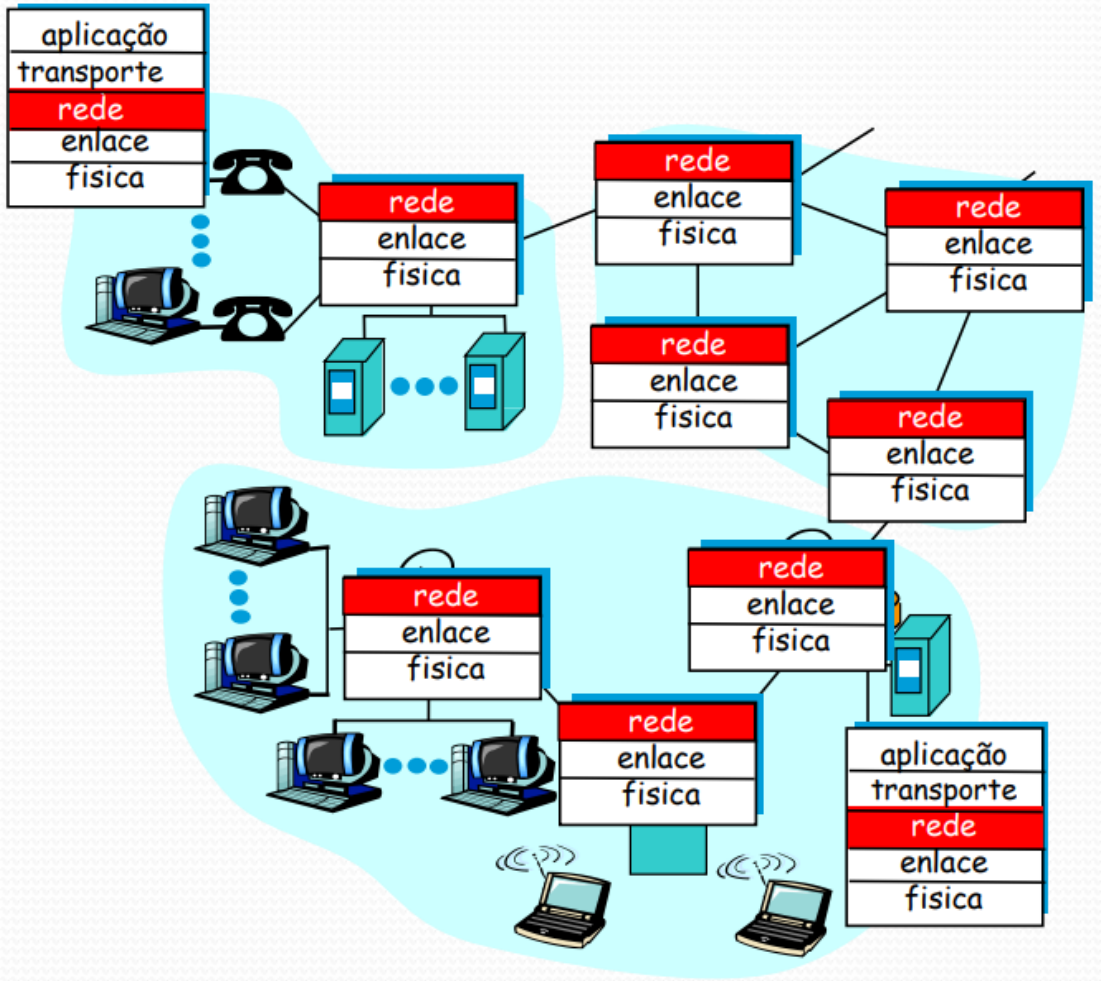
- As rotas escolhidas pelos pacotes entre a origem e o destino. Algoritmos de roteamento

Camada de Rede – Comutação

- Ato de mover pacotes entre as portas de entrada e de saída dos roteadores

Camada de Rede – Estabelecimento de conexão

- Algumas arquiteturas de rede exigem o estabelecimento de **circuitos virtuais** antes da transmissão de dados

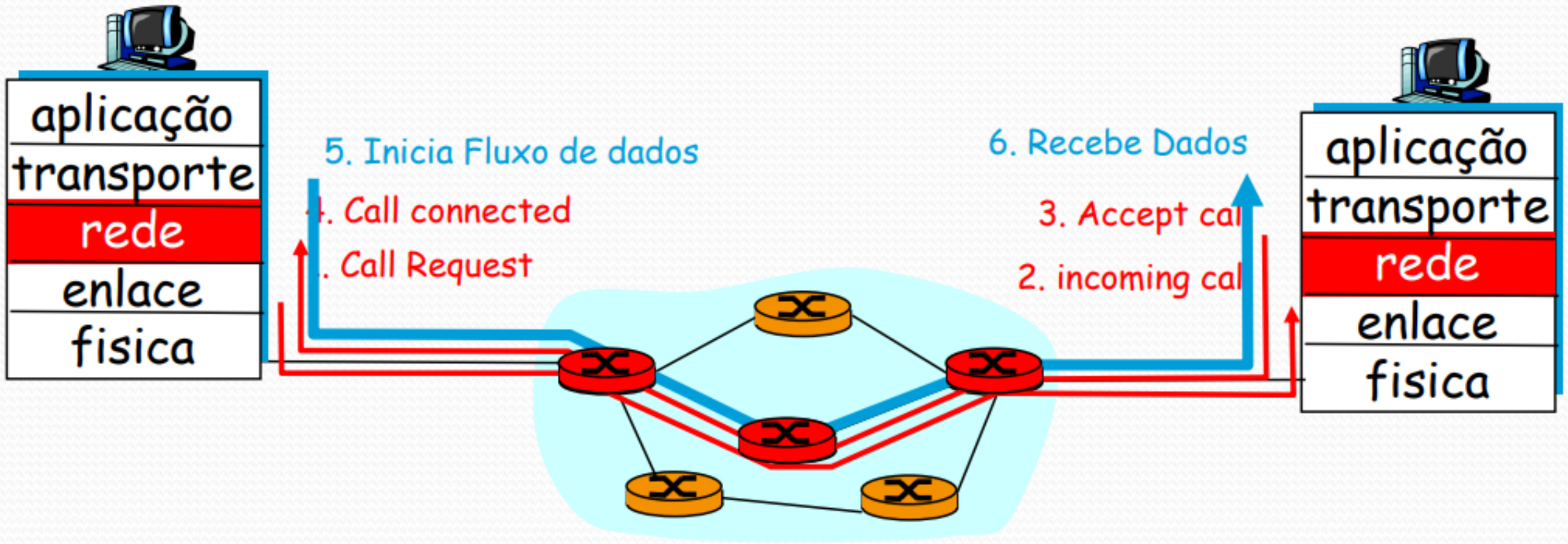


Camada de Rede – Circuitos Virtuais (VC)

- “A ligação entre a origem e o destino emula uma ligação telefônica”
  - A rede controla a conexão entre a origem e o destino
- Estabelecimento da conexão deve proceder o envio de dados. Liberação da conexão após os dados recebidos estejam completos.
- Cada roteador na rota mantém informação de estado para conexão que passa por ele.
- A banda e os recursos do roteador podem ser alocado por VC
  - Controle de Qualidade de Serviço por VC

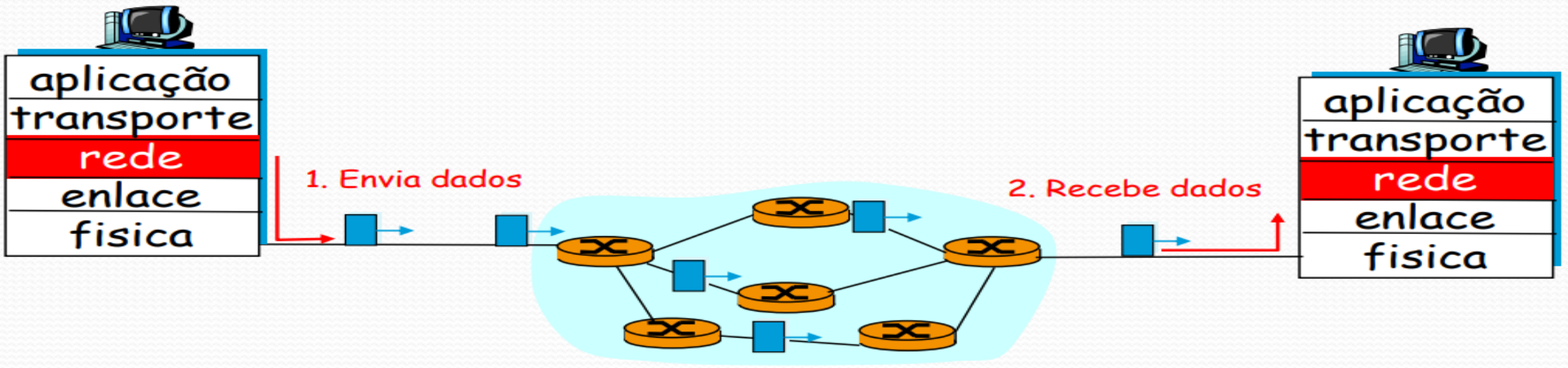
Camada de Rede – Circuitos Virtuais: Sinalização

- Usado para estabelecer, manter e encerrar Circuitos Virtuais



Camada de Rede – Datagrama (Modelo da internet)

- Não existem conexões na camada de transporte
- Não há informação de estado de conexão nos roteadores
  - Não existe conexão na camada de rede
- Pacotes tipicamente transportam o endereço de destino
  - Pacotes para o mesmo destino podem seguir diferentes rotas



Camada de Rede – Datagrama vs Circuito Virtual

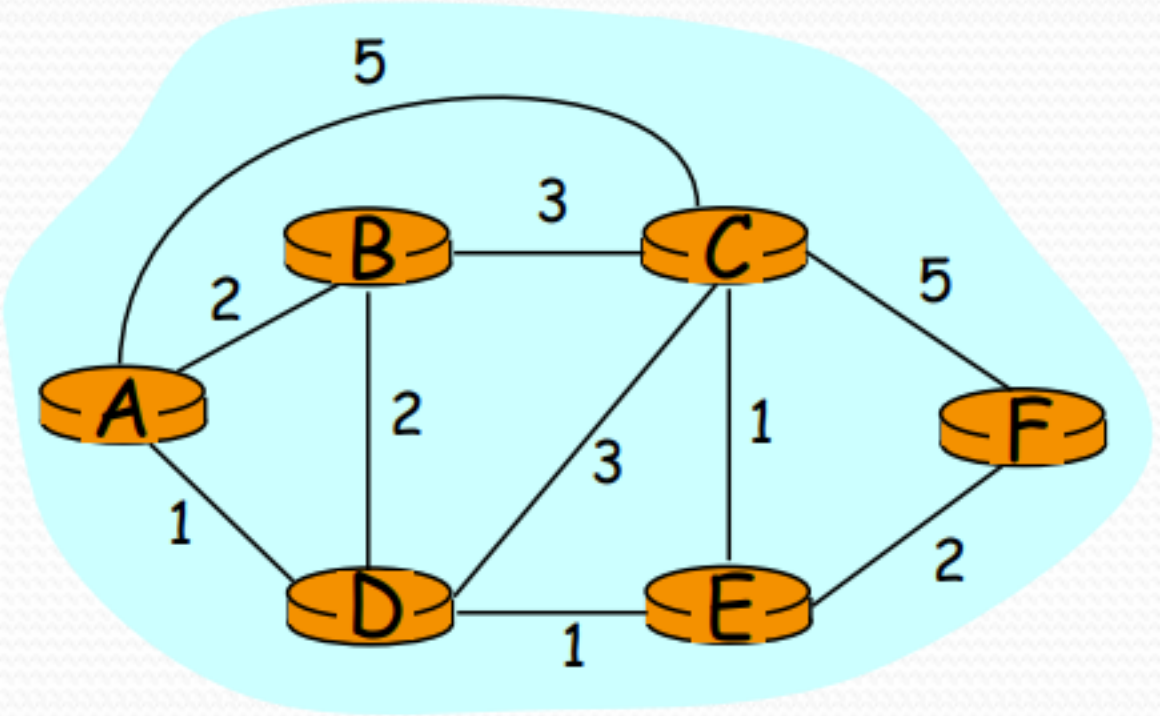
- Datagrama
  - Dados trocados entre computadores
  - Sistemas finais inteligentes: Podem adaptar-se, realizar controle e recuperação de erros
  - A rede é simples
  - Muitos tipos de enlaces (cabeada, Wi-fi..
- Circuito Virtual
  - Originário da telefonia
  - Conversação humana
  - Tempos estritos, exigências de confiabilidade
  - Sistemas finais “burros”: Telefones e Complexidade dentro da rede

Camada de Rede – Roteamento

- Objetivo principal:
  - ✓ **Determinar "bons" caminhos** (sequência de roteadores) através da rede de origem até o destino.

Camada de Rede – Roteamento

- Algoritmos de roteamento são descritos por grafos:
  - ➢ Nós do gráfico são roteadores
  - ➢ Arestas do gráfico são enlaces
    - ✓ Custo do enlace: atraso, preço ou nível de congestão
- <u>Bons caminhos:</u>
  - ✓ tipicamente corresponde aos caminhos de menor custo
  - ✓ caminhos redundantes



Classificação dos Algoritmos de Roteamento

- Informação global ou descentralizada
  - **Global**
    - ✓ Todos os roteadores tem informações completas da topologia e do custo dos enlaces
    - ✓ algoritmos **"Link state"**
  - **Descentralizada**
    - ✓ Roteadores só conhecem informações sobre seus vizinhos e os enlaces para eles
    - ✓ Processo de computação interativo, troca de informações com os vizinhos
    - ✓ algoritmos **"Distance vector"**

Classificação dos Algoritmos de Roteamento

- Estático ou Dinâmico?
  - **Estático**
    - ✓ As rotas mudam lentamente ao longo do tempo
  - **Dinâmico**
    - ✓ As rotas mudam mais rapidamente
      - Atualizações periódicas
      - Podem responder a mudanças no custo dos enlaces

Algoritmo Link-state

- **Algoritmo de Dijkstra's**
  - ✓ Topologia de rede e custo dos enlaces são conhecidos por todos os nós.
    - ➢ Implementado via "*link state broadcast*"
    - ➢ Todos os nós têm a mesma informação
  - ✓ Computa caminhos de menor custo de um nó (origem/fonte) para todos os outros nós
    - ➢ Fornece uma tabela de roteamento para aquele nó
  - ✓ Convergência: após **k** iterações, conhece o caminho de menor custo para **k** destinos.

Algoritmo Link-state - Dijkstra's cálculos de rota

- Notação:
  - $C(i,j)$: custo do enlace do nó i ao nó j. Custo é infinito se não houver ligação entre i e j.
  - $D(v)$: valor atual do custo do caminho da fonte ao destino v
  - $P(v)$: nó predecessor ao longo do caminho da fonte ao nó v, isto é, antes do v
  - N: conjunto de nós cujo caminho de menor custo é definitivamente conhecido

Algoritmo “Distance Vector”

- Iterativo:
  - Continua até que os nós não troquem mais informações.
- Assíncrono
  - ✓ Os nós não precisam trocar informações simultaneamente!
- Distribuído
  - ✓ Cada nó se comunica apenas com os seus vizinhos, diretamente conectados

Algoritmo “Distance Vector”

- Estrutura de Dados da Tabela de Distância
  - ✓ Cada nó tem sua própria tabela
  - ✓ Linha para cada possível destino
  - ✓ Coluna para cada roteador vizinho
  - ✓ Exemplo: no nó X, para destino Y via vizinho Z:

Algoritmo “Distance Vector” - Resumo

- Iterativo, assíncrono: cada iteração local é causada por:
  - Mudança de custo dos enlaces locais
  - Mensagem do vizinho: seu caminho de menor custo para o destino mudou
- Distribuído
  - Cada nó notifica seus vizinhos apenas quando seu menor custo para algum destino muda
    - Vizinhos notificam seus vizinhos e assim por diante