

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

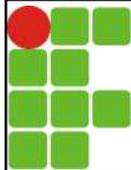


Introdução às Redes de Computadores

Turma : 20192.1.01405.1N

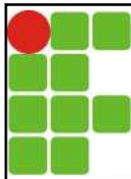
Camada de Enlace – Parte I

Prof. Thiago Dutra <thiago.dutra@ifrn.edu.br>



Agenda – Camada de Enlace

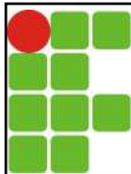
- Introdução
- Detecção e Correção de Erros
- Protocolos de Acesso Múltiplo
- Endereçamento
- Padrão Ethernet
- Padrão Wi-Fi



Agenda – Parte I

- Introdução
 - Visão Geral
 - Serviços
- Detecção e Correção de Erros
 - Paridade, Checksum e CRC
- Protocolos de Acesso Múltiplo
 - Particionamento de Canal
 - Acesso Aleatório
 - Revezamento

3



Visão Geral

■ Pilha TCP/IP

5. Aplicação
4. Transporte
3. Rede
2. Enlace
1. Física

4

Visão Geral

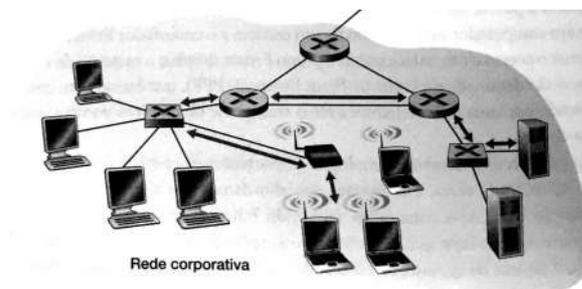
■ Terminologias

- **Nó** = qualquer dispositivo que implemente um protocolo de enlace (camada 2)
 - Ex.: Hosts, roteadores, switches, access points, ...
- **Enlace** = canais de comunicação que conectam nós adjacentes sobre o caminho de comunicação
 - Enlaces podem ser com ou sem fios
- Pacote = **quadro** ou **frame**
 - Encapsula um datagrama da camada 3 (rede)

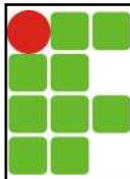
5

Visão Geral

- A camada de enlace tem a **responsabilidade de transferir um datagrama de um nó para o nó adjacente através de um enlace individual**
 - No caminho fim-a-fim, o datagrama passa por enlaces diferentes, que podem executar protocolos diferentes, cada qual com características distintas



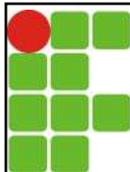
6



Visão Geral

- Analogia com transporte de pessoas
 - Turista viajando de Ceará-Mirim até Fortaleza guiado por um agente de viagens
 - Ceará-Mirim -> Natal = Trem
 - Natal -> Aeroporto = Carro
 - Aeroporto -> Fortaleza = Avião
- Entidades
 - Turista = datagrama
 - Segmento de transporte = enlace
 - Meio de transporte = protocolo de enlace
 - Agente de viagens = protocolo de roteamento

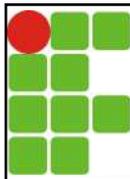
7



Serviços

- Embora o serviço básico de mover os datagramas entre nós adjacentes por um único enlace seja realizado por todos os protocolos de enlace, os detalhes do serviço podem variar de um protocolo para outro
- Principais serviços que podem ser ofertados pelos protocolos de enlace:
 - Enquadramento dos dados
 - Acesso ao enlace
 - Entrega confiável
 - Controle de fluxo
 - Detecção e Correção de erros
 - Modo de transmissão (Half-duplex ou Full-duplex)

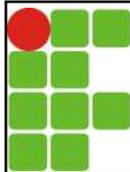
8



Serviços

- **Enquadramento dos dados:** encapsular os datagramas em quadros acrescentando cabeçalhos (no início do pacote) e trailers (no final do pacote). O formato dos quadros é específico de cada protocolo;
- **Acesso ao enlace:** controlados pelos protocolos MAC (Medium Access Control). Define as regras para escrita e leitura de dados no meio físico. Pode ser muito simples (ou até inexistente) nos meios com apenas um transmissor e receptor e bem complexo quando temos acesso múltiplo e compartilhado ao meio (ex.: Ethernet com único enlace de broadcast onde apenas um nó pode transmitir por vez);

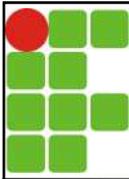
9



Serviços

- **Entrega confiável:** garantir a entrega de quadros entre nós adjacentes. Exige a implementação de confirmações de recebimento, retransmissões e numeração dos quadros (detectar quadros fora de sequência), gerando sobrecarga de processamento. Em geral utilizado por meios com altas taxas de erro (ex.: wireless) para evitar o aumento das retransmissões fim-a-fim.
- **Controle de fluxo:** busca garantir que um nó transmissor não "entupa" um nó receptor, evitando as perdas de pacote. O mecanismo mais comum é limitar o fluxo na origem, por exemplo, fazendo com que o transmissor pergunte ao receptor quanto pacotes pode enviar.

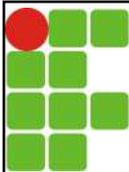
10



Serviços

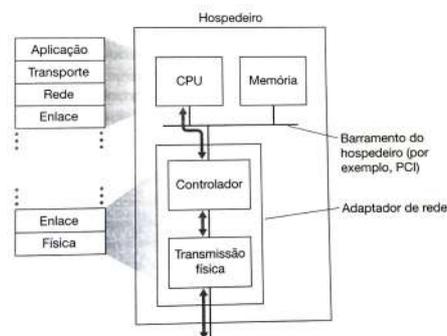
- **Detecção de erros:** busca erros na transmissão dos bits que são introduzidos por atenuação e ruídos. Campos são adicionados para que o receptor verifique a existência de erros através de algum algoritmo específico (ex.: paridade, CRC, ...). Importante saber que **não existem algoritmos 100% eficazes**. Como não é necessário repassar o pacote, em geral quando um erro é detectado o receptor solicita ao transmissor que reenvie o quadro.
- **Correção de erros:** semelhante a detecção de erros com o acréscimo da possibilidade de corrigir o erro encontrado, evitando assim a necessidade de retransmissão. Em geral tanto a detecção e correção ocorrem em hardware.

11

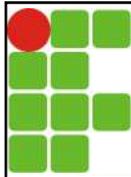


Serviços

- Onde esses serviços de enlace são executados nos nós?
 - Maior **parte em hardware**, nas placas de redes (NIC)
 - Mas também **parte em software**, CPU do hospedeiro
 - **Emissor:** ativa o controlador, encapsula o datagrama, inclui bits de verificação de erros, controle de fluxo, ...
 - **Receptor:** extrai o datagrama e repassa para a camada superior, procura erros, controle de fluxo, ...



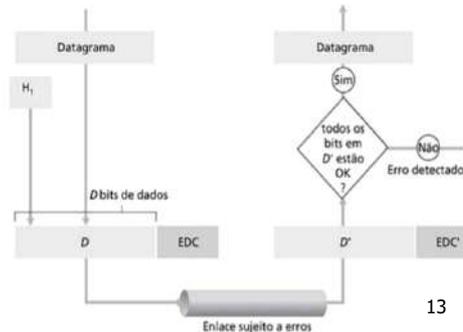
12



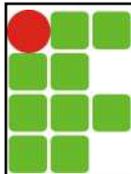
Detecção e Correção de Erros

■ Cenário para detecção e correção de erros

- EDC (Error Detection-and-Correction)
 - Bits acrescentados para possibilitar a detecção e correção de erros
 - Em geral, um maior campo EDC gera melhor resultado na detecção e correção
- D (Data)
 - Dados protegidos pela verificação de erros
 - Pode incluir campos de cabeçalho (endereçamento, sequência, ...)
- A detecção e correção de erros não é 100% confiável
 - Protocolos podem deixar passar alguns erros, mas é raro



13

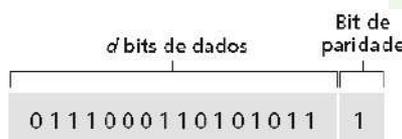


Verificação de Paridade

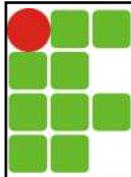
■ Paridade com bit único

- Detecta erro de um único bit
- Verifica a quantidade de bits "1" em D + EDC

Paridade	D (dados)	EDC	Quantidade de 1s
Par	101	0	2
Par	101001	1	4
Ímpar	101	1	3
Ímpar	101001	0	3



14



Verificação de Paridade

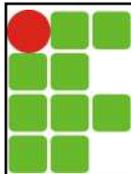
Exemplos de detecção com da paridade de bit único

Paridade	D+EDC	D'+EDC'	Deteção	Quantidade de 1s
Par	1010	1110	Sim	3
Par	101001	1010010	Sim	3
Ímpar	1011	0011	Sim	2
Ímpar	101001	0110010	Não	3

Problemas com a detecção de bit único

- Além de não detectar erros em mais de um bit, a probabilidade de ocorrência de erro em bits independentes é baixa
- Frequentemente, os erros acontecem em aglomerações de bits (rajadas)
- Probabilidade de não detecção de erro nesse algoritmo -> 50%

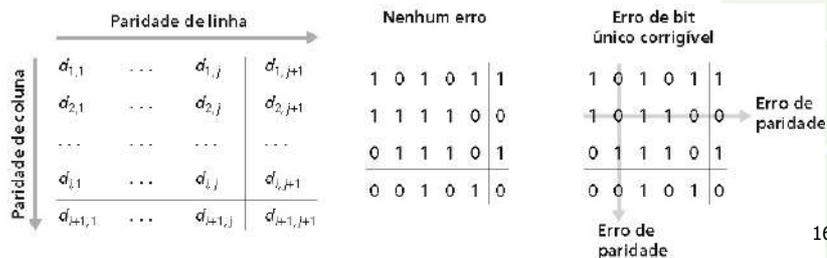
15



Verificação de Paridade

Paridade bidimensional

- Detecta e corrige erro de um único bit
 - Também detecta (mas não corrige) combinações de dois erros
- Os bits de D são divididos em i filas e j colunas e um bit de paridade é calculado para cada fila e coluna
 - EDC = $i + j + 1$ bits de paridade



16

Checksum

- **Checksum = soma de verificação**
 - Detectar "erros" (ex.: inversão de bits) no pacote transmitido (nota: utilizado somente na camada de transporte)
- **Emissor**
 - Trata o conteúdo do segmento como uma sequência de inteiros de 16 bits
 - **Checksum = adição (soma em complemento de 1) do conteúdo do segmento**
 - **Transmissor coloca o campo checksum no cabeçalho de transporte**
- **Receptor**
 - **Calcula o checksum do segmento recebido**
 - Verifica se o valor do checksum calculado é igual ao presente no cabeçalho
 - NÃO -> Erro detectado
 - SIM -> Nenhum erro detectado. **Mas pode haver erros?**

17

Checksum

(1) 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0
 (2) 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 (3) 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0
 (S1) = Soma parcial 1
 (S) = Soma
 (T) = Transbordo
 (Ch) = Checksum

(1) + (2) = (S1)

0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0 (1)
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 (2)
 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 (S1)

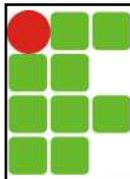
(S1) + (3) = (S)

1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 (S1)
1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 (3)
 ① 0 1 0 0 1 0 1 0 1 0 1 1 0 0 0 0 0 0 (T)
 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 (S)

Complemento de 1 (S) -> (Ch)

0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 (S)
1 0 1 1 0 1 0 1 0 0 1 1 1 1 1 0 (Ch)
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 (SF)

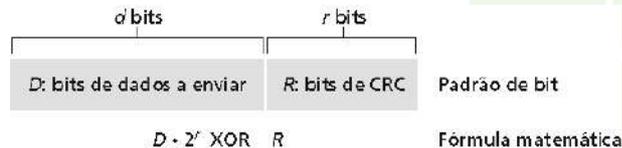
18



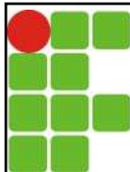
CRC

■ CRC (Cyclic Redundance Check)

- Verificação de redundância cíclica
- Encara os bits de dados, D , como um número binário
 - Códigos CRC também são conhecidos como **códigos polinomiais**, pois é possível considerar a cadeia de bits como um polinômio sendo os 0s e 1s os coeficientes do polinômio
- Escolhe um padrão gerador de $r + 1$ bit, G
 - Existem padrões internacionais de G com 8, 12, 16 e 32 bits



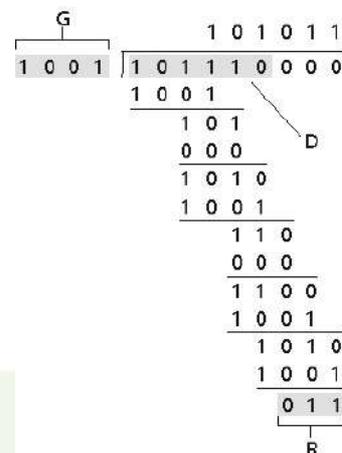
19

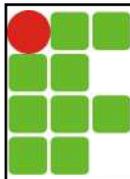


CRC

■ Objetivo: escolhe r bits de CRC, R , tal que

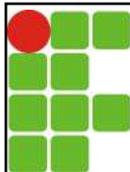
- $\langle D, R \rangle$ é divisível de forma exata por G (módulo 2)
- Receptor conhece G , divide $\langle D, R \rangle$ por G . **Se o resto é diferente de zero, erro detectado!**
- Pode detectar todos os erros em sequência (burst errors) com comprimento menor que $r + 1$ bit
- **Largamente usado na prática (Ethernet, 802.11 Wifi, ATM)**





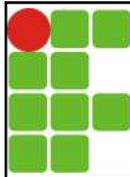
Protocolos de Acesso Múltiplo

- Basicamente existem dois tipos de "enlaces":
 - **Ponto-a-ponto**: meio dedicado entre dois nós
 - PPP para acesso discado
 - Enlace ponto-a-ponto entre switch e nó
 - **Broadcast**: meio compartilhado por vários nós
 - Ex.: Ethernet tradicional, HFC, Wifi 802.11 e Satélite comunitário



Protocolos de Acesso Múltiplo

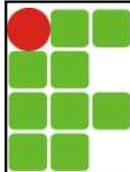
- A dificuldade existe nos enlaces broadcast
 - Duas ou mais transmissões simultâneas pelos nós -> interferência
 - **Colisão** = um nó recebendo dois ou mais sinais ao mesmo tempo
- Necessidade de controlar o acesso ao meio
 - **Protocolo de acesso múltiplo** = algoritmo distribuído que determina como as estações compartilham o canal, isto é, determinam quando cada estação pode transmitir
 - **Comunicação sobre o compartilhamento do canal deve usar o próprio canal**
 - Nenhum canal fora-de-banda para coordenação



Protocolos de Acesso Múltiplo

- Protocolo de Acesso Múltiplo Ideal para um Canal de broadcast de taxa R bps
 - 1. Quando um nó transmite sozinho, ele pode enviar a uma taxa R bps
 - 2. Quando M nós querem transmitir, cada um envia a uma taxa média R/M
 - 3. Totalmente descentralizada:
 - Nenhum nó especial para coordenar transmissões
 - Nenhuma sincronização de relógios e compartimentos
 - 4. Simples para proporcionar uma implementação barata

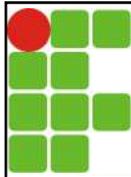
23



Protocolos de Acesso Múltiplo

- Existem três classes gerais
 - **Particionamento de canal**
 - Divide o canal em pedaços menores (compartimentos de tempo, frequência)
 - Aloca um pedaço para uso exclusivo de cada nó
 - **Acesso aleatório**
 - Canal não dividido, permite colisões
 - "Recuperação" das colisões
 - **Revezamento**
 - Nós transmitem nos seus turnos, porém nós com mais volume para enviar podem usar turnos mais longos

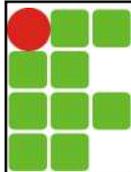
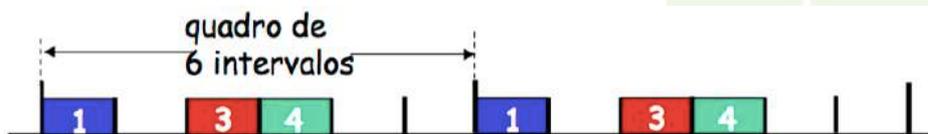
24



Particionamento de Canal

■ TDMA (Time Division Multiple Access)

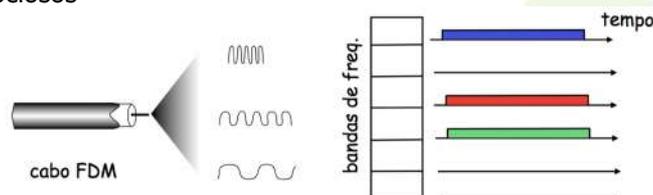
- Acesso ao canal em rodadas
- Cada estação controla um compartimento ("slot") de tamanho fixo a cada rodada
 - Tamanho = tempo para transmitir os pacotes
- Intervalos não usados ficam ociosos
 - Ex.: LAN com 6 estações -> 1,3,4 têm pacotes, 2,5 e 6 ficam ociosos

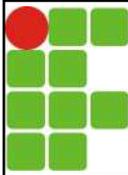


Particionamento de Canal

■ FDMA (Frequency Division Multiple Access)

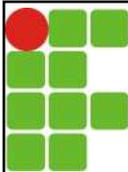
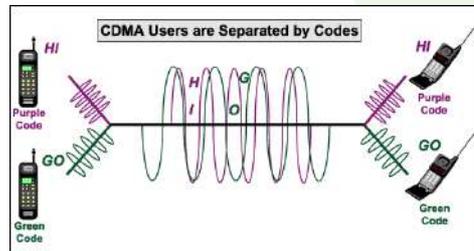
- Espectro do canal dividido em bandas de frequência
- Cada estação recebe uma banda de frequência fixa
- Tempo de transmissão não usado nas bandas de frequência fica ocioso
 - Ex.: LAN com 6 estações -> 1,3,4 têm pacotes, 2,5 e 6 ficam ociosos





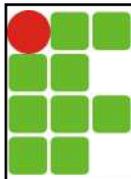
Particionamento de Canal

- **CDMA (Code Division Multiple Access)**
 - Atribuição de um código específico para cada nó
 - Cada nó utiliza seu código para codificar os bits que envia
 - Se os nós forem bem escolhidos, todos os nós podem transmitir simultaneamente
 - Analogia com o coquetel onde pares de pessoas conversam ao mesmo tempo, mas cada par falando em um idioma diferente



Acesso Aleatório

- Quando o nó tem um pacote a enviar:
 - Transmite com toda a taxa do canal R
 - Não há uma regra de coordenação a priori entre os nós
- Dois ou mais nós transmitindo -> "colisão"
- Um protocolo MAC de acesso aleatório especifica:
 - Como detectar colisões
 - Como as estações se recuperam das colisões (ex., via retransmissões atrasadas)
- Exemplos de protocolos MAC de acesso aleatório:
 - Slotted ALOHA, ALOHA, CSMA e CSMA/CD



Slotted ALOHA

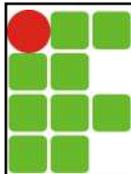
■ Suposições

- Todos os quadros são do mesmo tamanho
- Tempo dividido em slots de mesmo tamanho (suficiente para transmitir 1 quadro)
- Nós começam a transmitir quadros apenas no início dos slots
- Os nós são sincronizados
- Se 2 ou mais nós transmitem no slot, todos os nós detectam a colisão

■ Operação

- Quando um nó obtém um novo quadro, ele transmite no próximo slot
- Sem colisão, o nó pode enviar o novo quadro no próximo slot
- Se há colisão, o nó retransmite o quadro em cada slot subsequente com probabilidade p até o sucesso

29



Slotted ALOHA

■ Prós

- Um único nó ativo pode transmitir continuamente com a taxa completa
- Altamente descentralizada: somente slots em nós precisam ser sincronizados
- Simples

■ Contras

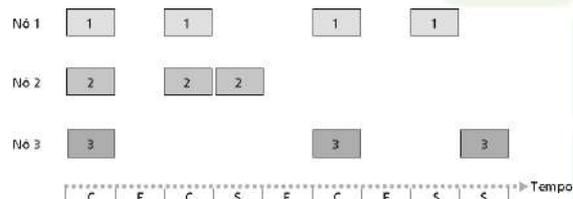
- Colisões, desperdício de slots, Slots ociosos
- Nós podem detectar colisão em menos tempo do que o tempo para transmitir o pacote
- Sincronização de clock
- **Baixa eficiência = 37%**

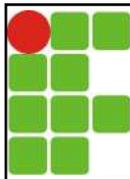
Legenda

C = Intervalo de colisão

E = Intervalo vazio

S = Intervalo bem-sucedido

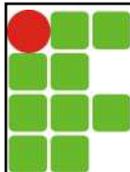
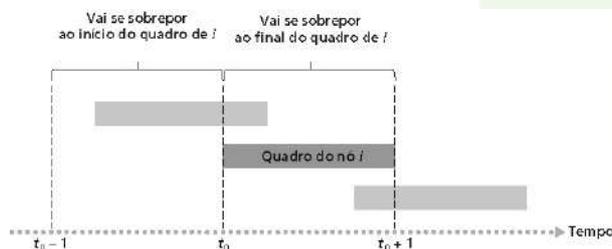




ALOHA

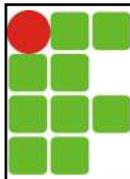
■ Unslotted ALOHA

- **Operação mais simples, não há sincronização**
- Assim que o primeiro quadro chega
 - Transmite imediatamente -> Aumento da probabilidade de colisão
 - Quando colide, retransmite imediatamente com probabilidade p
- **Baixa eficiência**
 - 18%
 - Pior que slotted



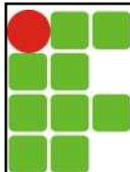
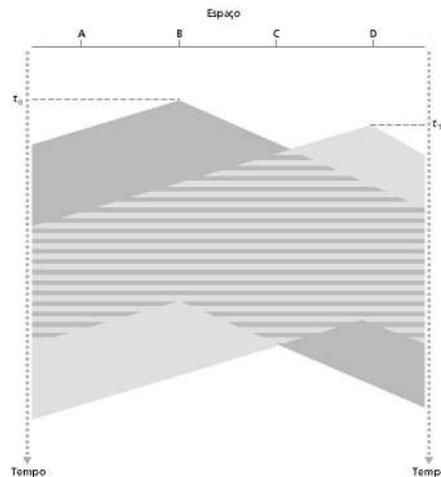
CSMA

- **Carrier Sense Multiple Access**
 - Acesso múltiplo com detecção de portadora
- **O nó que deseja transmitir escuta o meio antes de iniciar a transmissão**
 - Canal vazio -> transmite o pacote
 - Canal ocupado -> transmissão adiada
- **Analogia humana**
 - Ouça antes de falar -> detecção de portadora



CSMA

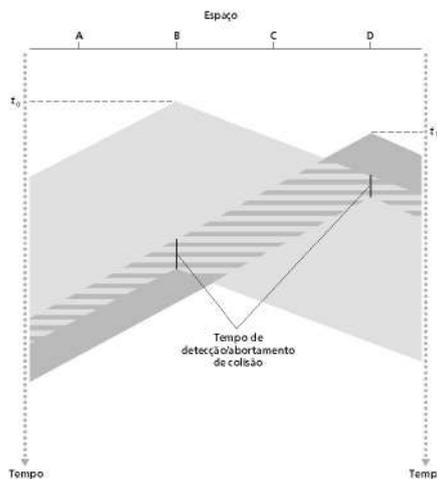
- Mesmo escutando o canal antes de transmitir, colisões podem ocorrer
 - Atraso de propagação = dois nós podem não ouvir a transmissão um do outro
 - Colisão -> todo o tempo de transmissão do pacote é desperdiçado
 - Importante perceber o papel da distância e do atraso de propagação na determinação da probabilidade de colisão



CSMA/CD

- O CSMA/CD incorpora um melhor mecanismo de detecção de colisões (collision detected) ao CSMA
 - O nó transmissor continua escutando o canal durante a transmissão
 - Colisões são detectadas dentro de pouco tempo
 - Colisão detectada -> transmissão abortada
 - Redução de desperdício do canal
- Detecção de colisão
 - Fácil em LANs cabeadas: medição da intensidade do sinal, comparação dos sinais transmitidos e recebidos
 - Difícil em LANs sem fio: receptor desligado enquanto transmite, intensidade do sinal recebido abafada pelo sinal da transmissão local
 - Analogia humana: se alguém começar a falar ao mesmo tempo que você, pare de falar

CSMA/CD



35

Revezamento

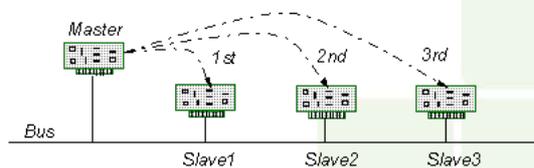
- Protocolos MAC com particionamento de canais:
 - Compartilham o canal eficientemente quando a carga é alta e bem distribuída
 - Ineficiente nas cargas baixas: atraso no acesso ao canal.
 - A estação consegue uma banda de $1/N$ da capacidade do canal, mesmo que haja apenas 1 nó ativo!
- Protocolos MAC de acesso aleatório
 - Eficiente nas cargas baixas: um único nó pode usar todo o canal
 - Cargas altas: excesso de colisões
- Revezamento
 - Buscam unir o melhor dos dois mundos!

36

Revezamento

■ Polling (seleção)

- Nó mestre convida os "escravos" a transmitirem um de cada vez
- Normalmente utilizado com nos escravos "burros"
- Problemas
 - Sobrecarga da seleção (overhead polling)
 - SPOF (Único ponto de falha) no mestre
 - Latência

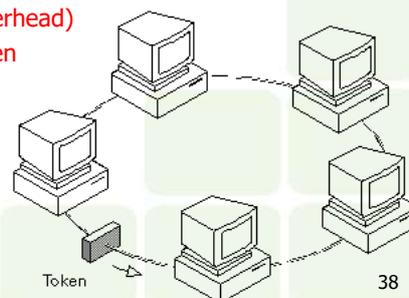


37

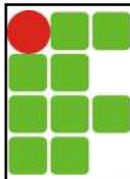
Revezamento

■ Passagem de Permissão

- Permissão de controle (token) passada de um nó para o próximo sequencialmente
- Um nó só "segura" a permissão se tiver dados para transmitir
- Problemas
 - Sobrecarga da permissão (token overhead)
 - SPOF (Único ponto de falha) no token
 - Latência

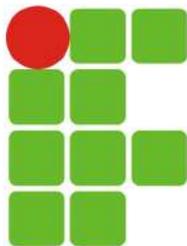


38



Referências

- KUROSE, J. F. e ROSS, K. - **Redes de Computadores e a Internet** – 6a Ed., Pearson, 2013.
- KUROSE, J. F. e ROSS, K. - **Redes de Computadores e a Internet** – 5a Ed., Pearson, 2010.



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE



Introdução às Redes de Computadores

Turma : 20192.1.01405.1N

Camada de Enlace – Parte I

Prof. Thiago Dutra <thiago.dutra@ifrn.edu.br>