

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

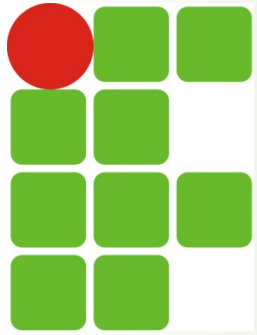


REDE FEDERAL
DE EDUCAÇÃO
PROFISSIONAL
E TECNOLÓGICA
1909-2009

Redes de Computadores

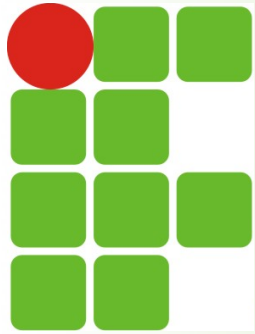
Camada de Rede





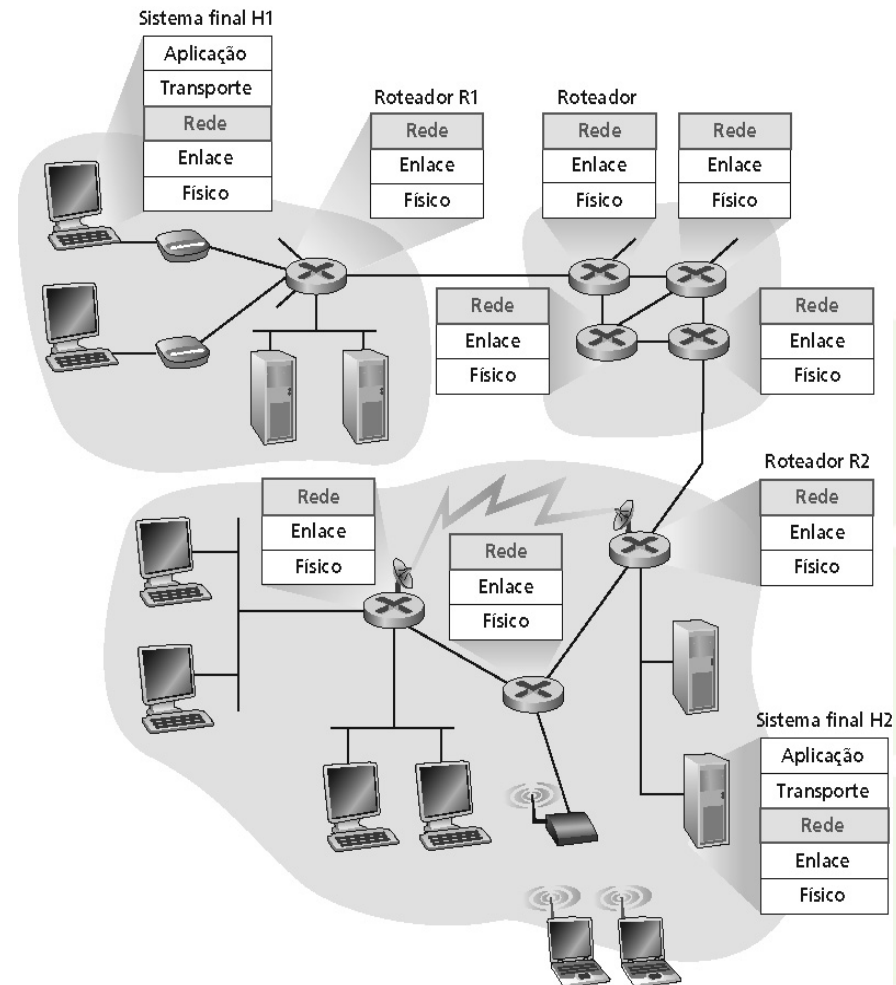
Objetivo

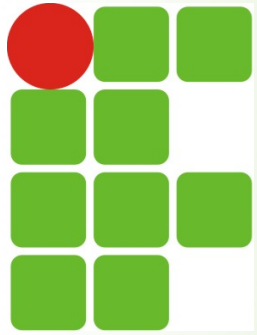
- Conhecer as características, funcionalidades e protocolos da camada de rede, especialmente os protocolos IP e ICMP
- Entender as principais características e princípios operacionais do serviço de entrega de datagramas não confiável e da função de roteamento
- Conhecer o formato e os campos de datagramas IP
- Entender as funcionalidades do protocolo ICMP e suas principais mensagens



Camada de Rede

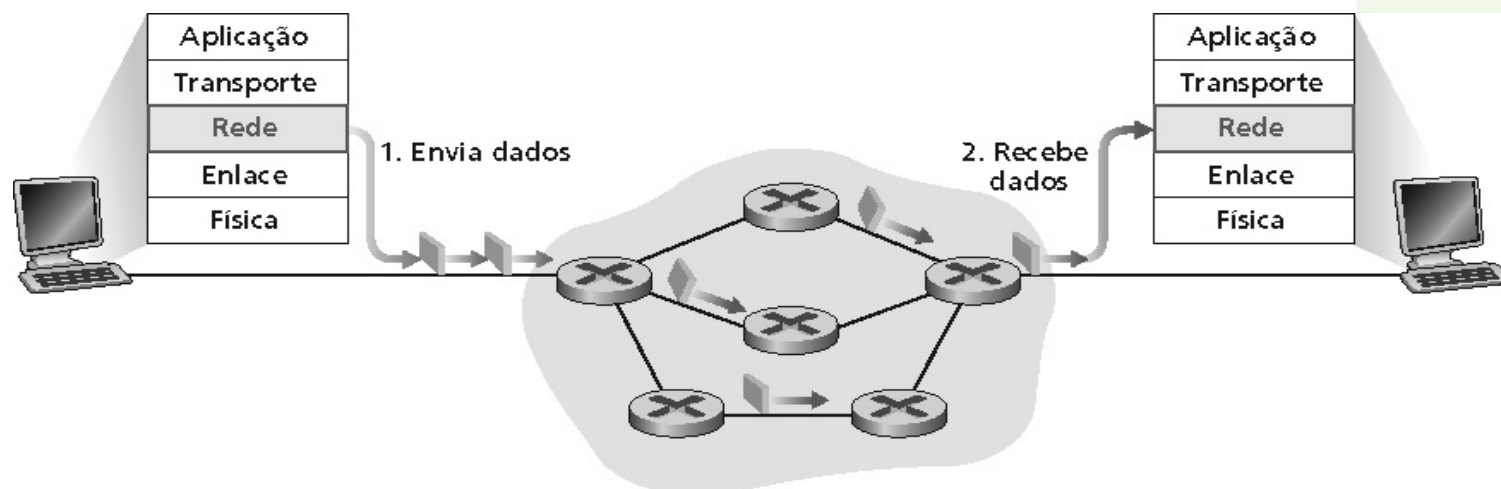
- Transporta segmentos do hospedeiro transmissor para o receptor;
- No lado transmissor, encapsula os segmentos em datagramas;
- No lado receptor, entrega os segmentos à camada de transporte;
- Protocolos da camada de rede em cada hospedeiro, roteador;
- Roteador examina campos de cabeçalho em todos os datagramas IP que passam por ele.

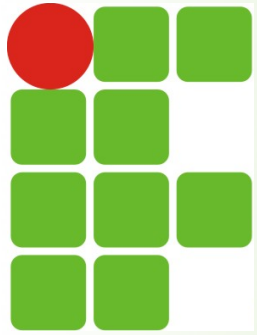




Redes de datagrama

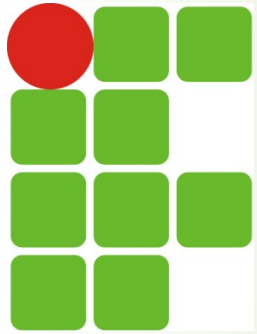
- Não existe estabelecimento de conexão na camada de rede
- Roteadores: não existe estado sobre conexões fim-a-fim
- O conceito "conexão" não existe na camada de rede
- Pacotes são encaminhados pelo endereço do hospedeiro de destino
- Pacotes para o mesmo destino podem seguir diferentes rotas





Funcionalidades da camada de rede

- Serviço de entrega
 - Aceita e encaminha os datagramas até o destino final, possivelmente por meio de roteadores intermediários
- Roteamento
 - Determina o caminho ou rota que cada datagrama deve seguir para alcançar a rede destino



Serviço de entrega

■ Característica

■ Serviço não confiável

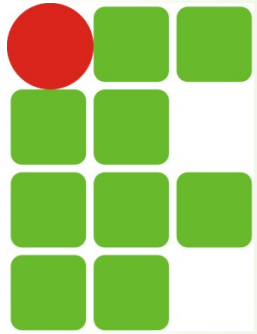
- Não garante a entrega dos datagramas
- Pode perder, retardar e duplicar datagrama
- Não garante a integridade dos dados

■ Serviço sem conexão

- Datagramas são individuais e independentes
- Seqüência dos datagramas não é assegurada

■ Paradigma de melhor esforço

- Descarta datagramas apenas em condições de falta de recursos ou erros de transmissão



Roteamento

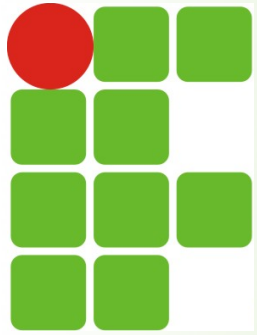
■ Característica

■ Modelo passo-a-passo (hop-by-hop)

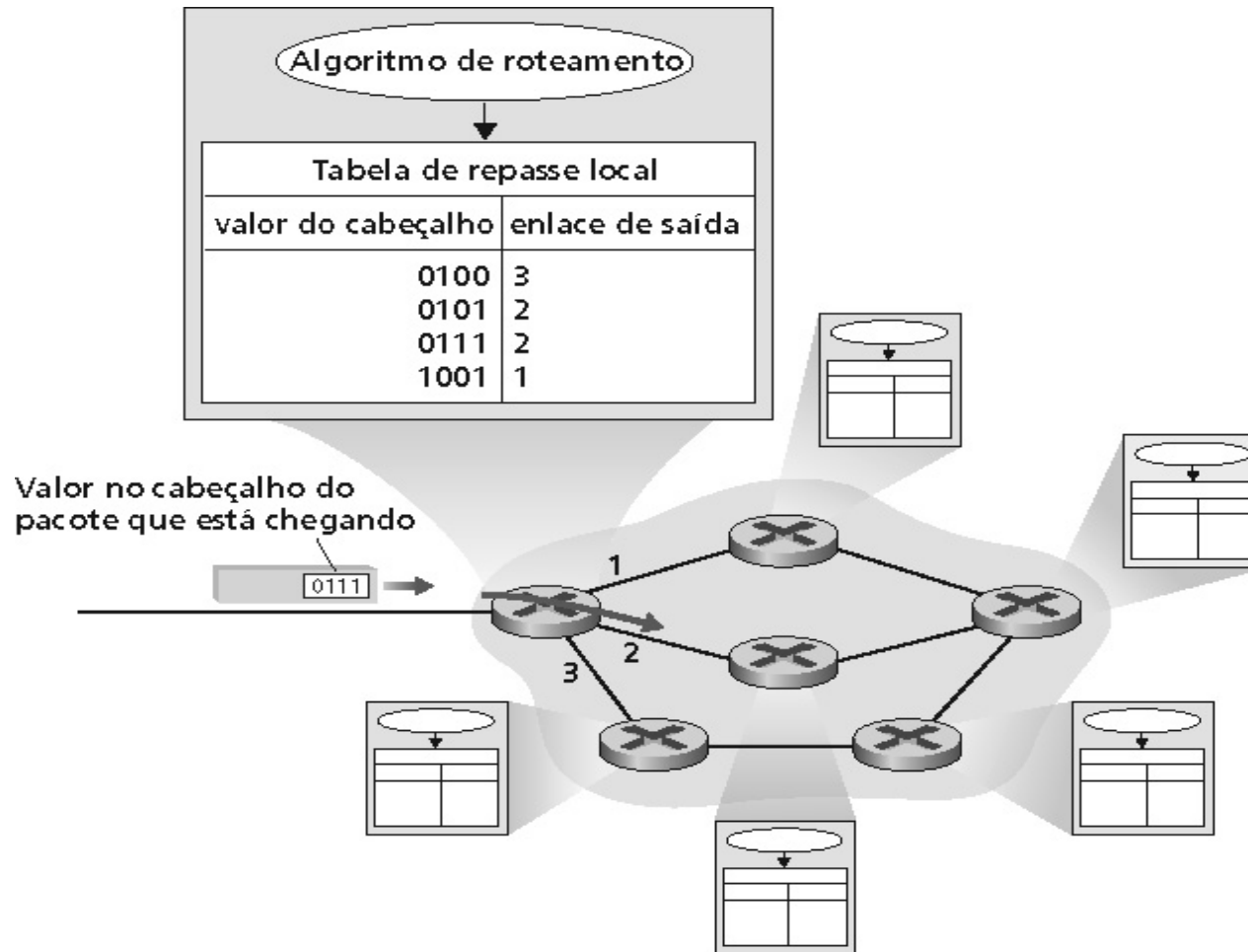
- Estações de uma mesma rede podem enviar datagramas diretamente entre si
- Estações em redes distintas devem enviar ao próximo roteador do caminho (next hop)

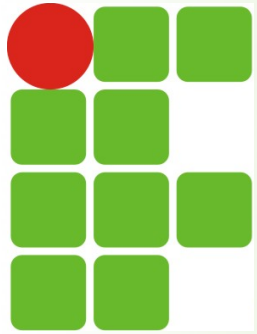
■ Tabela de roteamento

- Matem rotas para as diversas redes ou estações
- Rotas indicam apenas o próximo roteador do caminho



Interação entre roteamento e repasse





Composição

■ Protocolos

■ IP (*Internet Protocol*)

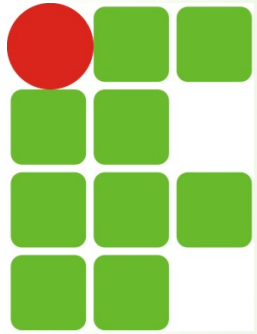
- Provê o serviço de entrega de datagramas
- Transporta informações dos protocolos ICMP, IGMP, TCP e UDP

■ ICMP (*Internet Control Message Protocol*)

- Permite a troca de mensagem de erros e controle entre as entidades da camada de rede

■ IGMP (*Internet Group Message Protocol*)

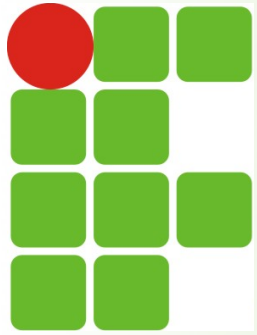
- Provê serviço de entrega Multicast



Protocolo IP

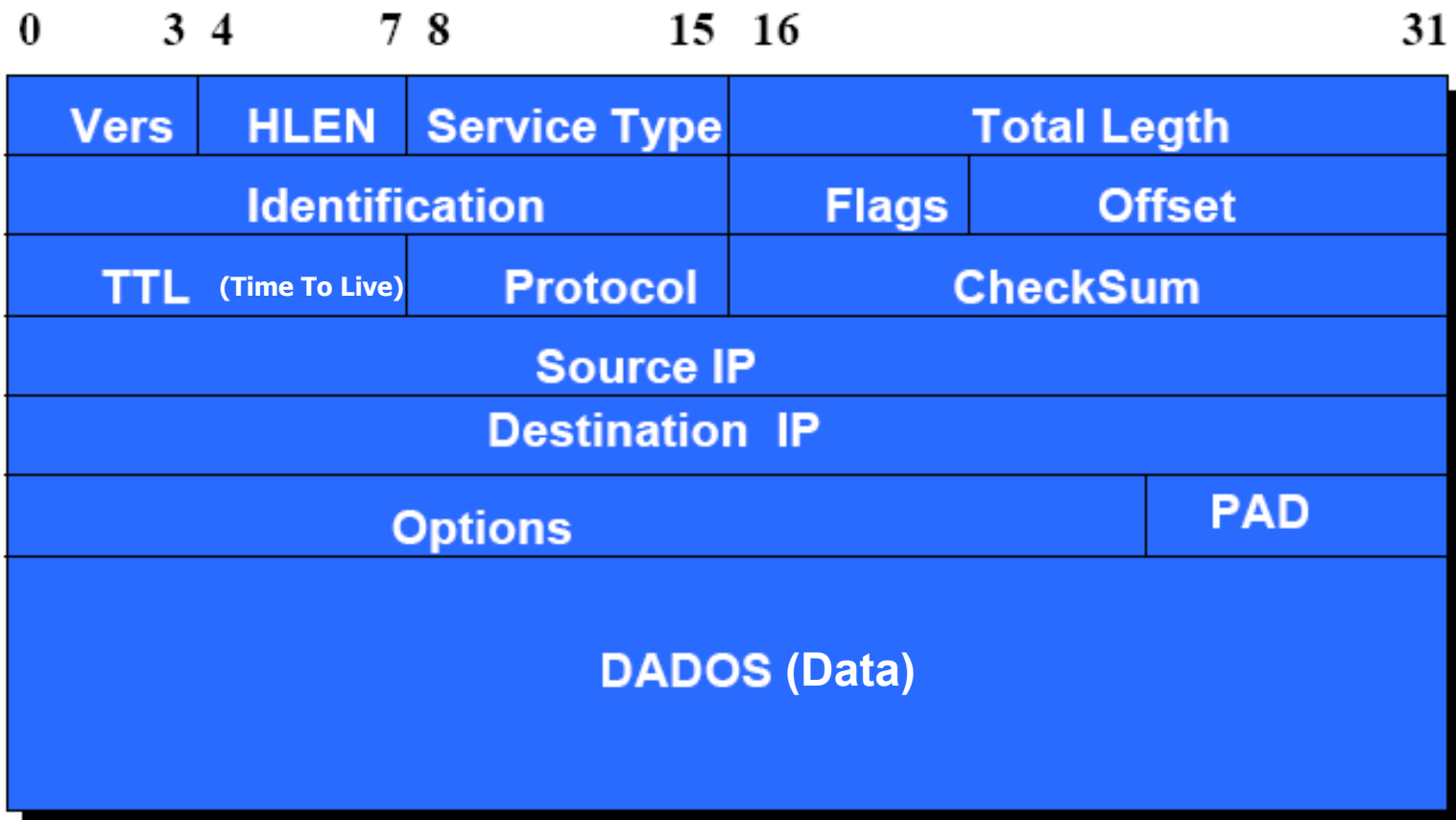
■ Fundamentos

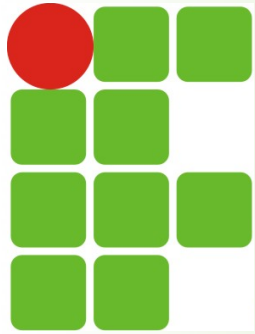
- Define a unidade básica (PDU) de transferência de dados, denominada **Datagrama IP**
 - Especifica o formato e a função dos campos
- Executa a função de roteamento
 - Escolhe a rota por onde os datagramas IP são enviados da origem até o destino
- Define as regras do serviço de entrega
 - Como os datagramas são processados
 - Como e quando erros são reportados
 - Quando datagramas são descartados



Protocolo IP

■ Formato do datagrama





Protocolo IP

■ Campos do datagrama

■ Vers

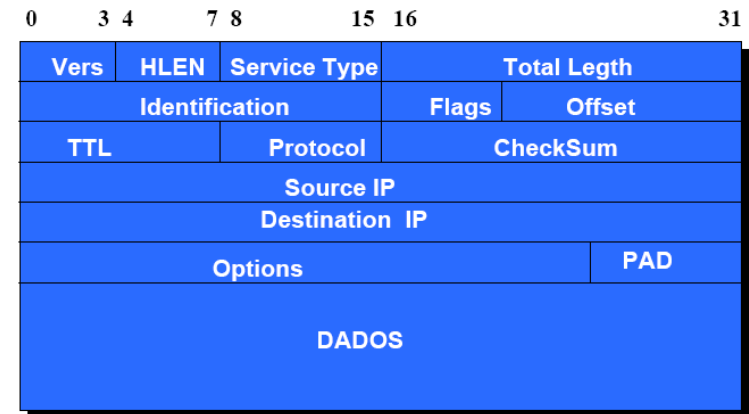
- Versão do IP utilizada. Versão desse formato é a versão 4 (IPv4)

■ Hlen

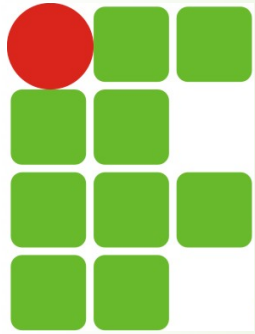
- Tamanho do cabeçalho do datagrama (palavras de 32 bits)
- O valor mínimo = 5

■ Total Len

- Tamanho total do datagrama em octetos (Bytes)
- Como esse campo é de 16 bits, então o maior datagrama possível é 65.535 (2^{16})



Data)



Protocolo IP

■ Campos do datagrama

■ Service Type

- Contém parâmetros referentes à qualidade de serviço desejada. Possui 8 bits que **indicavam** os seguintes requisitos

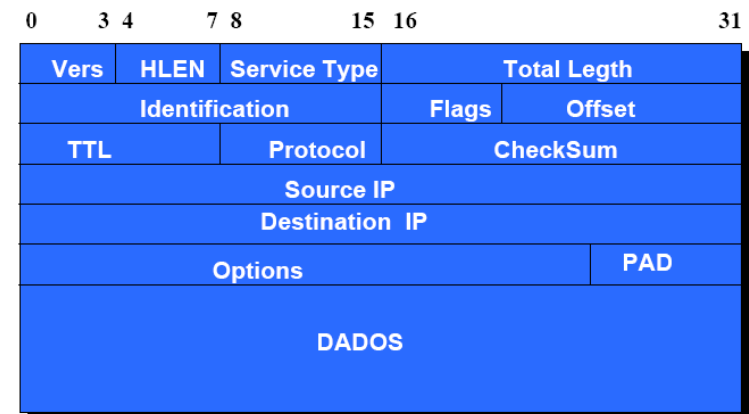
0	1	2	3	4	5	6	7
Precedência	D	T	R	0	0		

Bits 0 a 2: Precedência (3 bits)

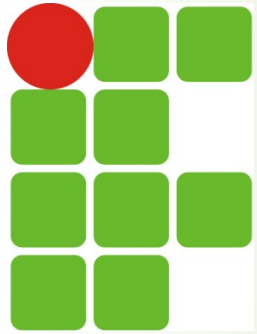
Bit 3 (D) : 0 = atraso normal 1 = baixo atraso

Bit 4 (T) : 0 = vazão normal 1 = vazão alta

Bit 5 (R) : 0 = confiabilidade normal 1 = alta conf.



Arquitetura DiffServ: RFC 2474 redefina o campo ToS(Tipo de Serviço) como campo de Serviços Diferenciados



Protocolo IP

■ Campos do datagrama

■ TTL (Time To Live)

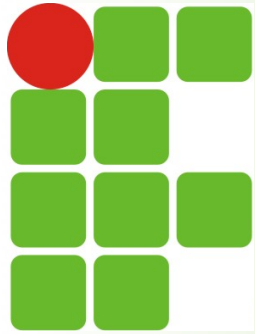
- Número máximo de roteadores (gateways) que um datagrama pode passar
- Cada gateway ao repassar um datagrama decrementa de um este valor, caso resulte em zero, o datagrama é descartado e o roteador gera uma mensagem de erro

■ Protocol

- Tipo do protocolo encapsulado no datagrama(TCP, UDP, ICMP, IGMP, IP, ...)

■ CheckSum

- Garante a integridade do cabeçalho
- Trata como seqüência de inteiros de 16 bits
- Soma, usando aritmética de complemento a 1, e calcula o complemento a 1 do resultado



Protocolo IP

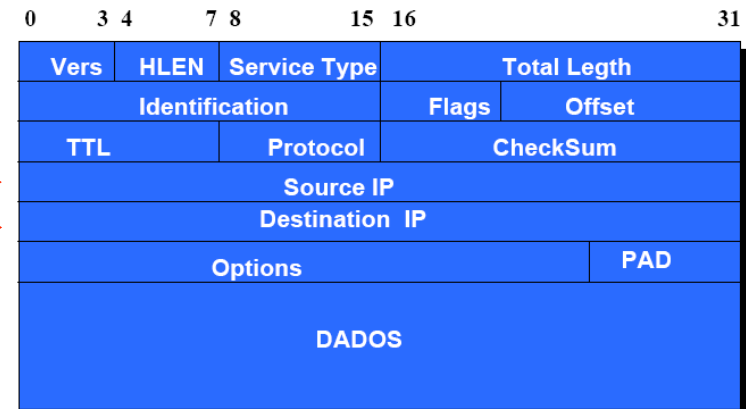
■ Campos do datagrama

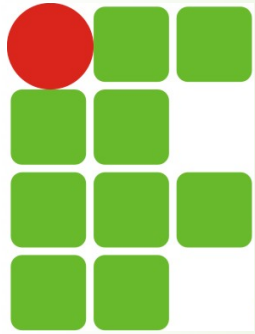
■ Source IP

- Endereço de rede da estação emissora do datagrama

■ Destination IP

- Endereço de rede da estação destino do datagrama

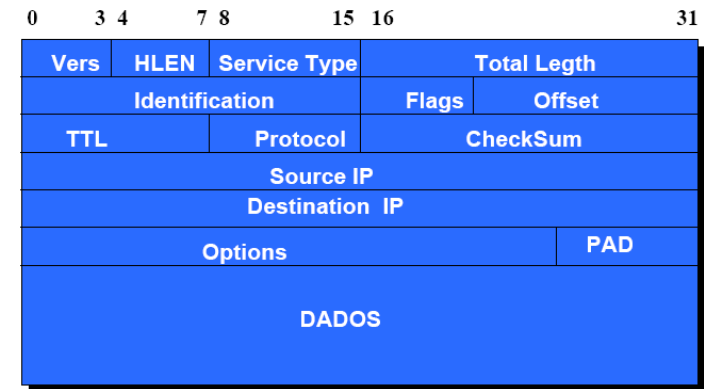


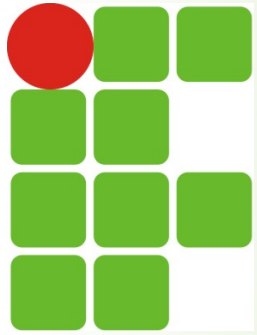


Protocolo IP

■ Campos do datagrama

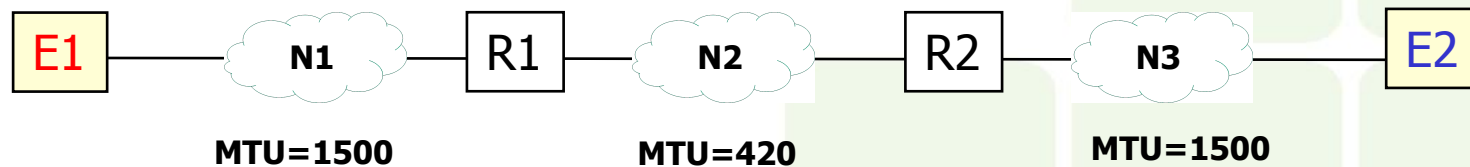
- Options (tamanho variável)
 - Registro de rota
 - Especificação de rota
 - Marca de Tempo (Timestamp) de cada gateway
 - Opção COPY / Fragmentação
- PAD
 - preenchimento com bits 0 (completar um tamanho múltiplo de 32)
- Dados:
 - Unidade de dados do protocolo superior (TCP, UDP, ICMP...)

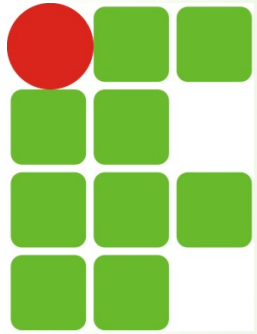




Protocolo IP

- MTU (*Maximum Transfer Unit*)
 - Cada tecnologia de rede física limita o tamanho máximo do quadro
 - O tamanho máximo do datagrama depende da tecnologia de rede física utilizada
 - O tamanho máximo do campo de dados de uma rede física é denominado MTU
 - MTU define o tamanho máximo do datagrama IP suportado na rede física.

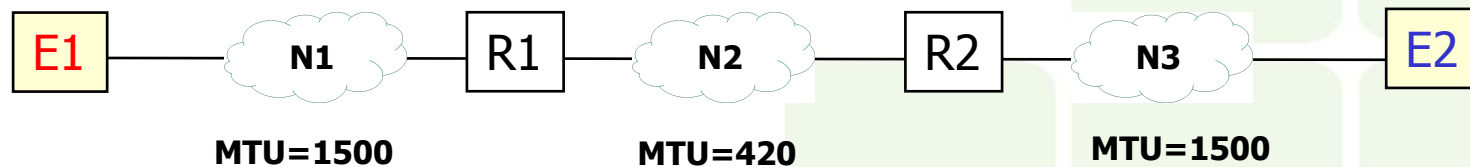


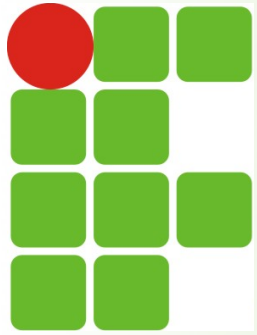


Protocolo IP

■ Fragmentação

- Processo de divisão de datagramas em pequenas partes, chamadas fragmentos
- Fragmentos e datagramas possuem formatos idênticos
- Datagrama original e seus respectivos fragmentos possuem cabeçalhos similares
- Fragmentos podem ser re-fragmentados.

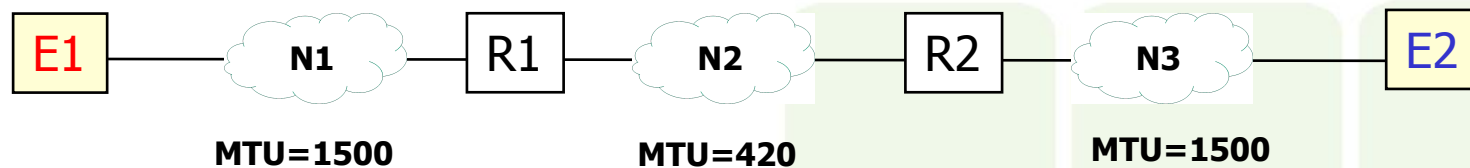


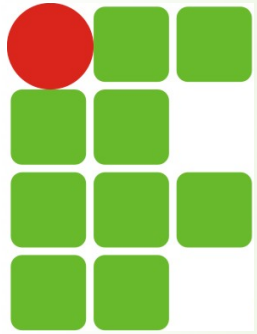


Protocolo IP

■ Remontagem

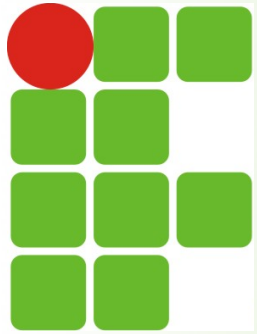
- Processo de recuperação do datagrama original a partir dos seus fragmentos
- É realizada apenas no destino final dos fragmentos
- O datagrama original não pode ser remontado quando algum fragmento é perdido ou atrasa
 - O destino final ativa um temporizador após a chegada de um fragmento
 - Fragmentos recebidos após a expiração do temporizador são descartados





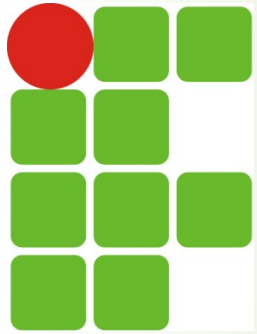
Protocolo IP

- Controle de fragmentação
 - Campo *Identification* (16 bits)
 - Número inteiro que identifica o datagrama original
 - Copiado para cada fragmento
 - Campo *Fragment OffSet* (13 bits)
 - Deslocamento dos dados do fragmento em relação ao datagrama original
 - Identificado em unidades de 8 bytes
 - Campo *Flags* (3 bits)
 - Serve ao controle de fragmentação
 - Indica se pode ou não fragmentar um datagrama
 - bit *Don't fragment* indica se pode ou não fragmentar
 - bit *More fragments* indica se houve fragmentação



Protocolo IP

- Algoritmo da fragmentação
 - O cálculo do tamanho do fragmento é função do MTU
 - Passos
 - Passo 1: Definir a quantidade de fragmentos
 - Passo 2: Definir o tamanho máximo do fragmento (em octetos)
 - Passo 3: Definir o tamanho da carga útil
 - Passo 4: Montar os fragmentos com os dados dos passos 1, 2 e 3 e configurar os Flags *Don't fragment* e *More fragments*



Protocolo IP

■ Algoritmo da fragmentação

■ Passo 1:

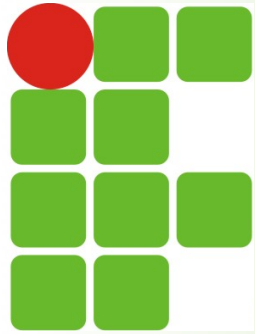
- $QTDF = DTGO / MTU$; Divisão inteira do datagrama original pela MTU do novo enlace
 - Se o resto da divisão for maior que zero, soma-se 1 a quantidade QTDF

■ Passo 2:

- $TMF = (MTU - HLen * 4) / 8$ (Divisão inteira, maior fragmento possível em octetos)

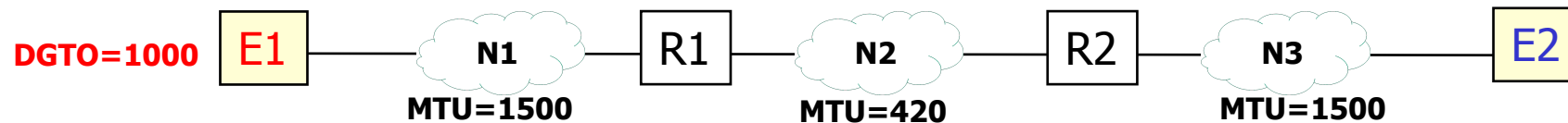
■ Passo 3:

- $TAD = TMF * 8$ (Tamanho da área de dados)
- Variação do OffSet = TMF
- Passo 4: Montar os fragmentos com os dados dos passos 1, 2 e 3



Protocolo IP

■ Exemplo de fragmentação



Identification	More Fragments	Fragment OffSet	HLen (Bytes)	Total Len	Data
555	0	0	5	1000	980

Datagrama original (DTGO)

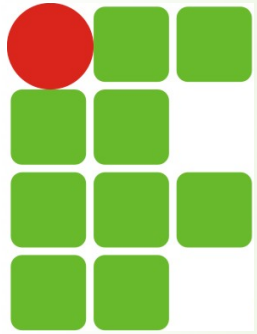
$QTDF = 980 / 400 \rightarrow 2,45$, logo $QTDF = 3$
 $TMF = (MTU - HLen * 4) / 8 \rightarrow (420 - 5 * 4) / 8 = 50$
 $TAD = TMF * 8 = 50 * 8 = 400$
 Variação do OffSet = $TMF = 50$

Identification	More Fragments	Fragment OffSet	HLen (Bytes)	Total Len	Data
555	1	0	20	420	400
555	1	50	20	420	400
555	0	100	20	200	180

f1

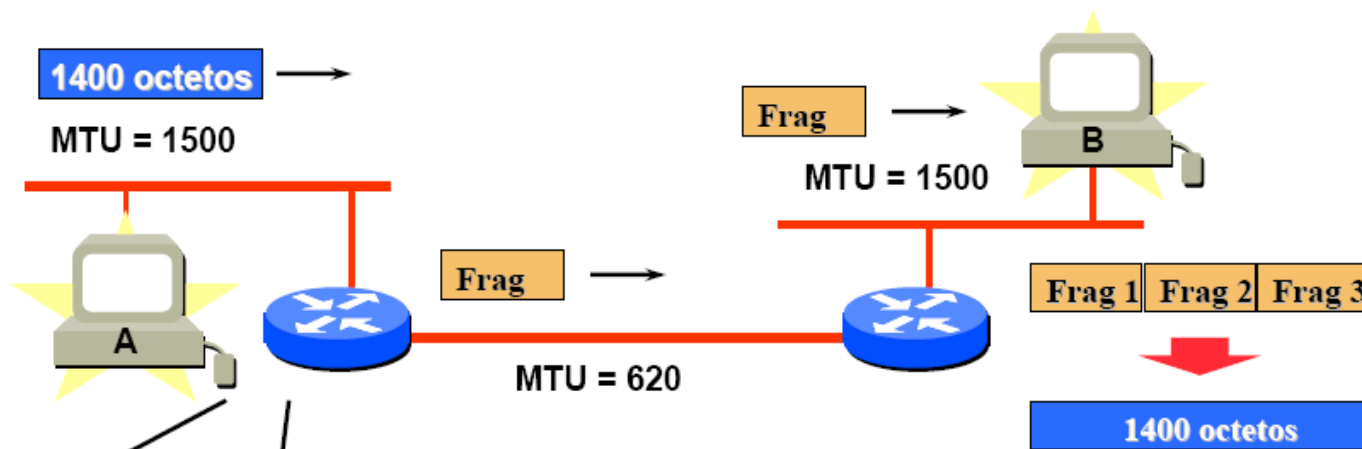
f2

f3



Protocolo IP

■ Exemplo de fragmentação

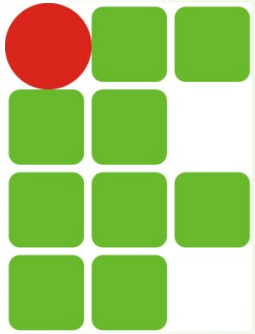


- Frag 1 600 octetos
- Frag 2 600 octetos
- Frag 3 200 octetos

Datagrama de 1400 octetos (dados) : A → B

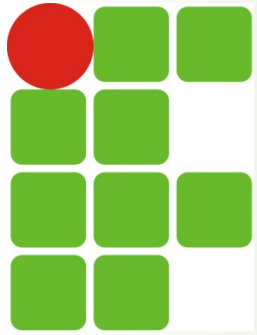
	Identification	Flags	Offset
Frag1:	xxxx	010	0
Frag2:	xxxx	010	75
Frag3:	xxxx	000	150

- Frag1 e Frag2 tem 600 bytes de comprimento
- Fragmentos só são remontados no host destino



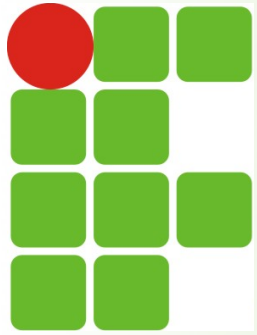
Protocolo IP

- Processo de remontagem
 - Os campos *Identification* e *Source IP Address* identificam os fragmentos de um datagrama
 - O primeiro fragmento é identificado pelo campo *Fragment OffSet = 0*
 - Fragmentos intermediários são posicionados de acordo com o seu *Fragment OffSet*
 - O último fragmento é detectado através do campo *More fragments*



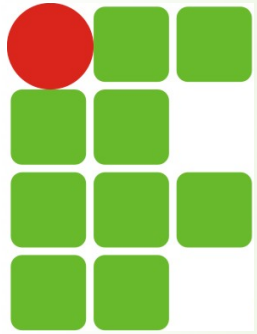
Protocolo IP

- Processamento de datagramas
 - Conjunto de regras que caracterizam o serviço de entrega de datagramas
 - As estações e os roteadores podem:
 - Enviar datagramas atuando como origens
 - Receber datagramas atuando como destino
 - Roteadores intermediários podem rotear datagramas, encaminhando-os de uma rede física para outra



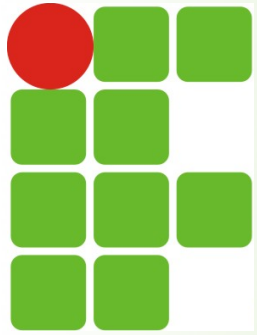
Protocolo IP

- Processamento na origem
 - Monta o datagrama
 - Preenche os campos dos cabeçalhos
 - Descobre rota para o destino
 - Envia datagramas
 - Entrega direta se origem e destino estiverem na mesma rede física
 - Entrega indireta a um roteador intermediário, se origem e destino estiverem em redes físicas diferentes



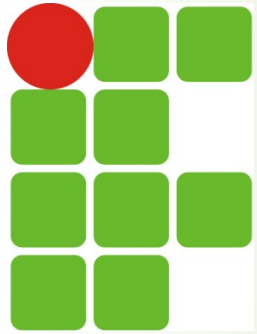
Protocolo IP

- **Processamento no destino**
 - **Recebe e armazena o datagrama**
 - Descarta datagrama e gera mensagem de erro quando não possui buffer
 - **Verifica integridade do cabeçalho**
 - Descarta o datagrama em caso de erro
 - **Identifica-se como destino do datagrama**
 - Senão, descarta-o
 - **Entrega o datagrama ao protocolo indicado**
 - Processo de remontagem pode ser necessário
 - Descarta fragmento e gera mensagem de erro em caso de falha



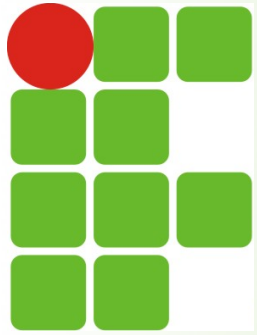
Protocolo IP

- Processamento no roteador
 - Recebe e armazena o datagrama
 - Descarta datagrama e gera mensagem de erro quando não possui buffer
 - Verifica integridade do cabeçalho
 - Descarta o datagrama em caso de erro
 - Identifica-se como roteador intermediário
 - Decrementa o TTL
 - Descarta datagrama e gera mensagem de erro quando o TTL atinge o valor zero



Protocolo IP

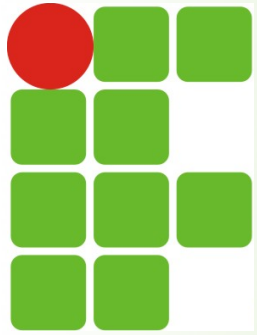
- Processamento no roteador
 - Descobre rota para o destino
 - Avalia necessidade de fragmentação
 - Descarta datagrama e gera mensagem de erro se a fragmentação for necessária e está desabilitada no datagrama (bit don't fragment = 1)
 - Envia o datagrama ou fragmentos
 - Entrega direta ao destino, se o destino estiver na mesma rede física do roteador
 - Entrega indireta a outro roteador, se o destino estiver em rede física diferente



Protocolo ICMP

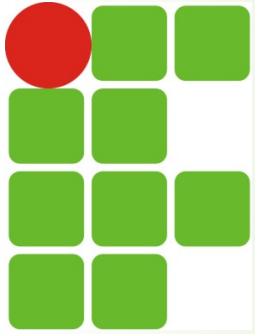
■ Fundamentos

- Permite ao protocolo IP das estações e roteadores trocarem informações de erro e controle
- Define diversos tipos de mensagem
 - Sinaliza algumas situações anormais
 - Permite a identificação de algumas informações operacionais
- Mensagens ICMP são encapsuladas em datagramas IP



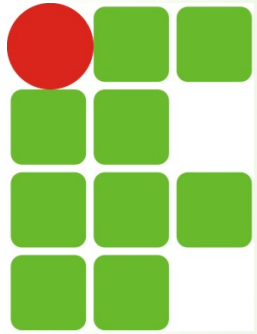
Protocolo ICMP

- Principais mensagens
 - Source quench
 - Realiza controle de congestionamento
 - Time exceeded
 - Indica que o TTL atingiu valor zero
 - Indica que nem todos os fragmentos foram recebidos durante o processo de remontagem
 - Destination unreachable
 - Sinaliza que não foi possível rotear o datagrama



Protocolo ICMP

- Principais mensagens
 - Redirect
 - Atua na otimização do roteamento
 - Echo request / Echo reply
 - Testa se o destino está operacional e pode ser alcançado através da rede



Protocolo ICMP

■ Principais mensagens

```
C:\>ping 10.1.1.1
```

```
Disparando contra 10.1.1.1 com 32 bytes de dados:
```

```
Resposta de 10.1.1.1: bytes=32 tempo<1ms TTL=64
```

```
Resposta de 10.1.1.1: bytes=32 tempo<1ms TTL=64
```

```
Resposta de 10.1.1.1: bytes=32 tempo=4ms TTL=64
```

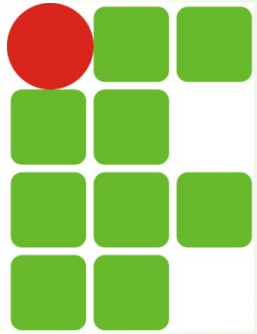
```
Resposta de 10.1.1.1: bytes=32 tempo<1ms TTL=64
```

```
Estatísticas do Ping para 10.1.1.1:
```

```
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
```

```
Aproximar um número redondo de vezes em milissegundos:
```

```
    Mínimo = 0ms, Máximo = 4ms, Média = 1ms
```



Protocolo ICMP

■ Principais mensagens

```
C:\>tracert www.cefetrn.br

Rastreando a rota para alecrim.cefetrn.br [200.137.1.15]
com no máximo 30 saltos:

 1      4 ms      <1 ms      4 ms      tirol.cefetrn.br [10.1.1.1]
 2      *        *          *          Esgotado o tempo limite do pedido.
 3      *        *          *          Esgotado o tempo limite do pedido.
 4      *        *          *          Esgotado o tempo limite do pedido.
```

- No Linux o comando é *traceroute*
- Com o comando `tcpdump` do Linux é possível verificar on-line o tráfego dos datagramas