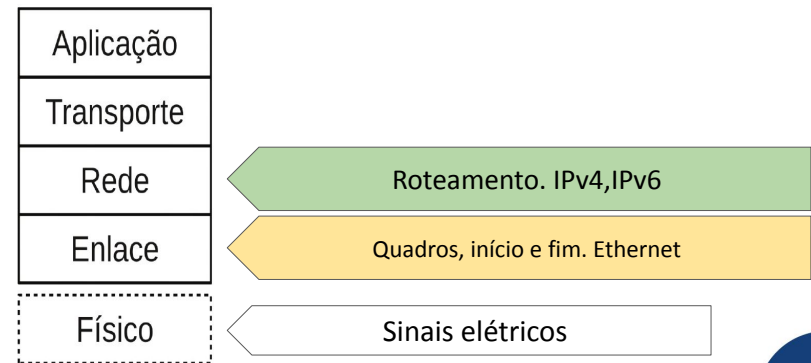


Redes de Computadores

Aula 5

Elgio Schlemer
elgio.schlemer@unilasalle.edu.br

Camada de rede



Nível de rede

- Responsável pelo Roteamento
- Como chegar ao destino?
 - Se mesmo domínio de broadcast:
 - destino é atingível, basta colar MAC destino no quadro
 - E obtém-se o MAC por ARP
 - se destino não for mesmo domínio de broadcast
 - necessário repassar quadro para o gateway
 - quadro vai para o MAC ADDRESS do gateway
 - necessário ARP para obter o MAC do gateway

ARP

- Address Resolution Protocol
- Protocolo para obter o MAC address de uma máquina.
- Basicamente um "Quem é a máquina X"
 - mas não é pelo nome (ai é DNS)
 - é pelo número IP
- Cada máquina tem uma tabela ARP
- Associando um IP à um Mac address
- Demonstração

Ipv4

- número IP representado por 32 bits
 - um "unsigned long int" do C
- Exemplo:

11000000 10101000 00000000 00001010
C0 A8 00 0A

3232235530 (decimal)

Ipv4

- Números binários e Hexa são de difícil compreensão para nós.
 - uso em forma Decimal

11000000 10101000 00000000 00001010
192 168 0 10

- usando pontos: 192.168.0.10

Cabeçalho IPv4

| | | | | | | | |
|---------------|-----|-----------------|----|-----------------|-------------------------------|----|----|
| 0 | 4 | 8 | 12 | 16 | 20 | 24 | 28 |
| Versão | IHL | Tipo de serviço | | Tamanho Total | | | |
| Identificação | | | | Flags | Fragmentação Offset (13 bits) | | |
| TTL | | Protocolo | | Header Checksum | | | |
| IP de Origem | | | | | | | |
| IP de Destino | | | | | | | |
| DADOS | | | | | | | |

Cabeçalho IPv4

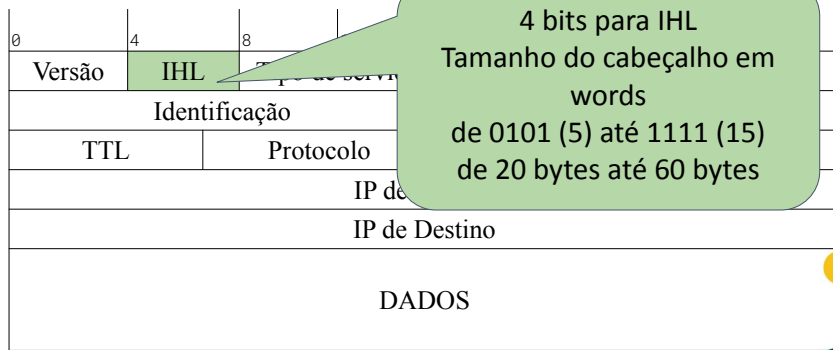
| | | | | | | | |
|---------------|---|-----------|----|-----------------|----|-------------------------------|----|
| 0 | 4 | 8 | 12 | 16 | 20 | 24 | 28 |
| Versão | | IHL | | Tipo de serviço | | Tamanho Total | |
| Identificação | | | | Flags | | Fragmentação Offset (13 bits) | |
| TTL | | Protocolo | | Header Checksum | | IP de Origem | |
| IP de Origem | | | | IP de Destino | | | |
| DADOS | | | | | | | |

4 bits para versão

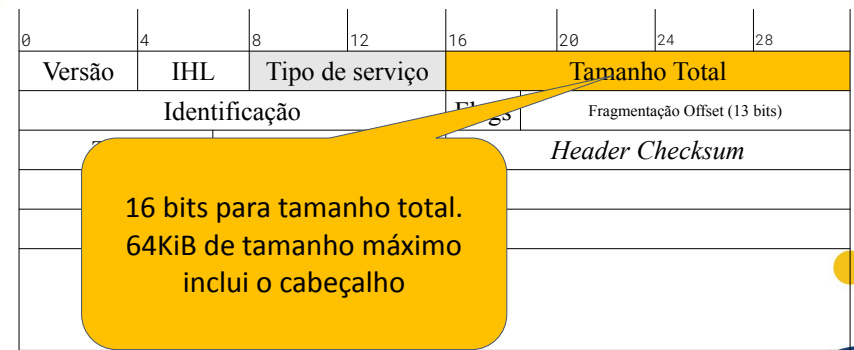
No caso do IPv4 sempre será 0100

4 bits para versão
No caso do IPv4 sempre
será
0100

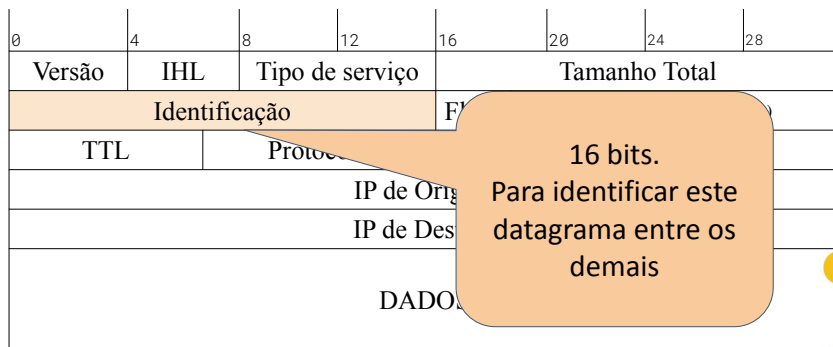
Cabeçalho IPv4



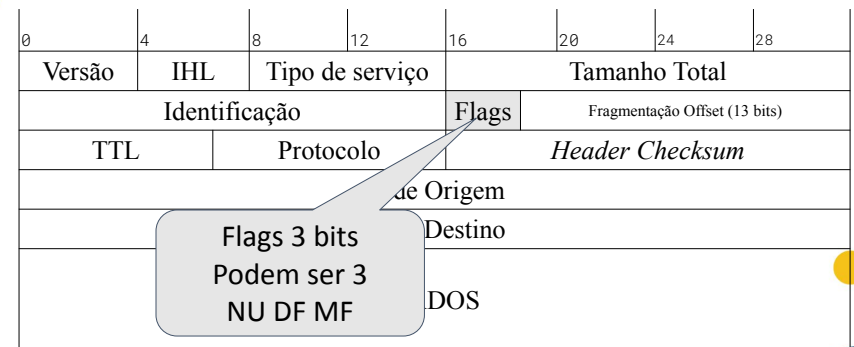
Cabeçalho IPv4



Cabeçalho IPv4



Cabeçalho IPv4



Cabeçalho IPv4

| | | | | | | | |
|---------------|-----------|-----------------|---------------|-----------------|-------------------------------|----|----|
| 0 | 4 | 8 | 12 | 16 | 20 | 24 | 28 |
| Versão | IHL | Tipo de serviço | Tamanho Total | | | | |
| Identificação | | | | Flags | Fragmentação Offset (13 bits) | | |
| TTL | Protocolo | | | Header Checksum | | | |
| IP de Origem | | | | | | | |
| Destino | | | | | | | |
| DADOS | | | | | | | |

TTL 8 bits
Time To Live

Cabeçalho IPv4

| | | | | | | | |
|---------------|---|-----------|----|-----------------|-------------------------------|---------------|----|
| 0 | 4 | 8 | 12 | 16 | 20 | 24 | 28 |
| Versão | | IHL | | Tipo de serviço | | Tamanho Total | |
| Identificação | | | | Flags | Fragmentação Offset (13 bits) | | |
| TTL | | Protocolo | | Header Checksum | | | |
| IP de origem | | | | | | | |
| IP de destino | | | | | | | |
| DADOS | | | | | | | |

Protocolo de transporte

Cabeçalho IPv4

| | | | | |
|---------------|-----------|-----------------|----|----------|
| 0 | 4 | 8 | 12 | 16 |
| Versão | IHL | Tipo de serviço | | |
| Identificação | | | | Flag |
| TTL | Protocolo | | | Checksum |
| IP de Origem | | | | |
| IP de Destino | | | | |
| DADOS | | | | |

Ips de origem (quem envia)
e de destino (para quem)

Fragmentação Ipv4

- Quando um datagrama é maior que o MTU
 - Ethernet tem MTU de 1500
 - mas um datagrama pode ter até 64KiB de dados
- necessário realizar a fragmentação do datagrama
 - qualquer roteador no meio do caminho pode fragmentar
 - apenas o destino monta os fragmentos

Fragmentação Ipv4

- Usa os campos do cabeçalho:
 - identificação: todos os fragmentos de um mesmo datagrama IP tem o mesmo identificador
 - Flags:
 - DF para Don't Fragment (indica que não pode fragmentar)
 - MF para More Fragment (indica que tem mais fragmentos)
 - Tamanho total: terá o tamanho deste fragmento
 - Fragmentação Offset
 - onde este fragmento se encaixa
 - informação em múltiplos de 8

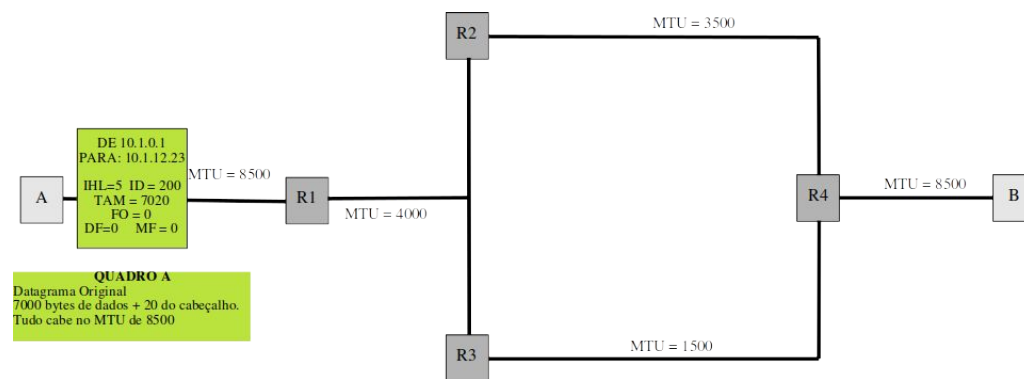
Exemplo Fragmentação Ipv4

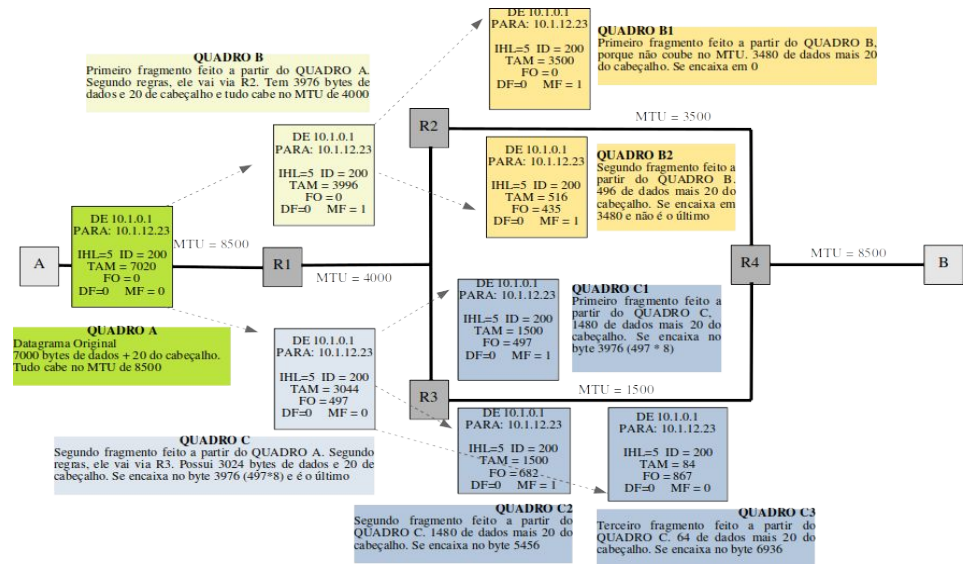
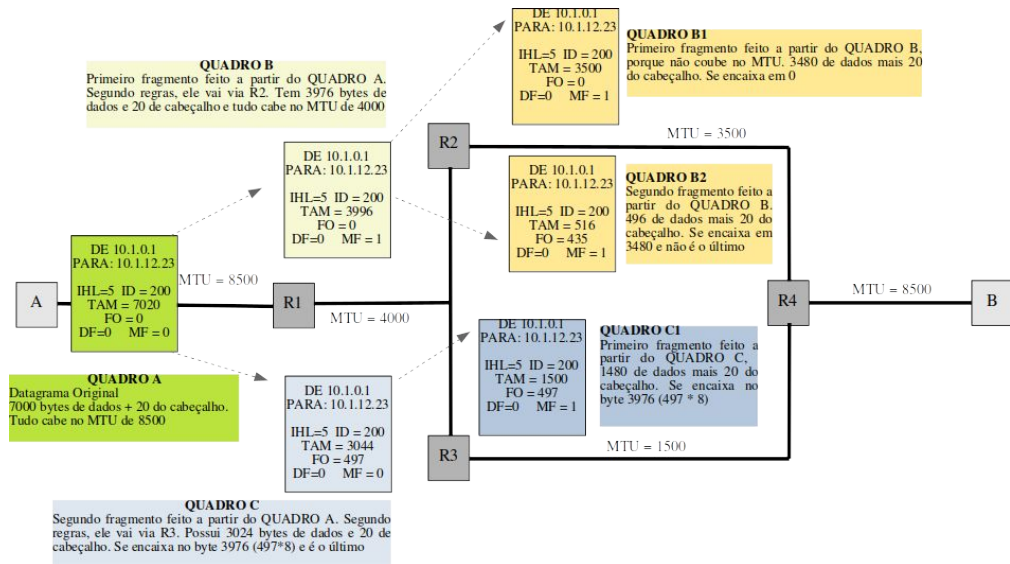
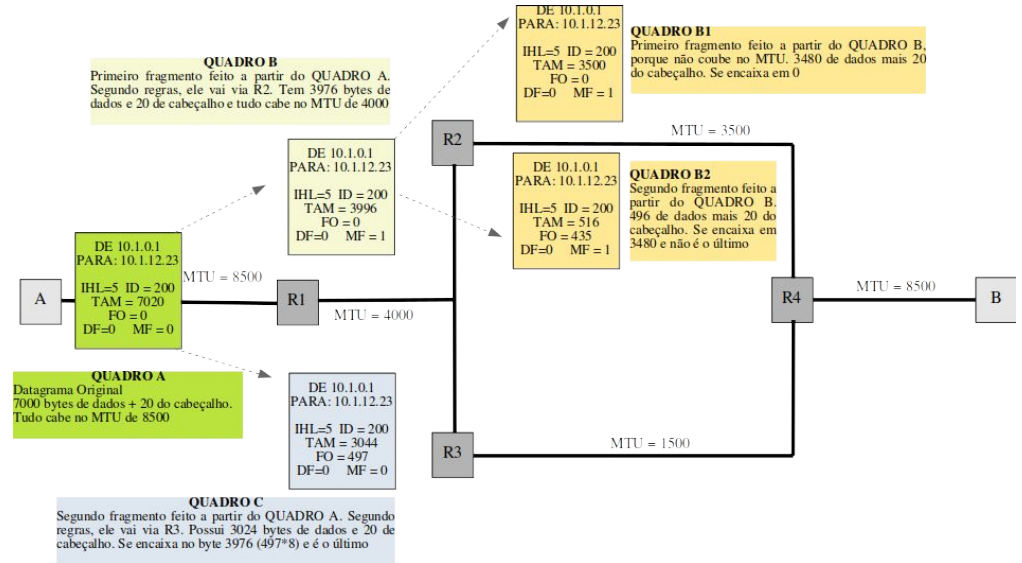
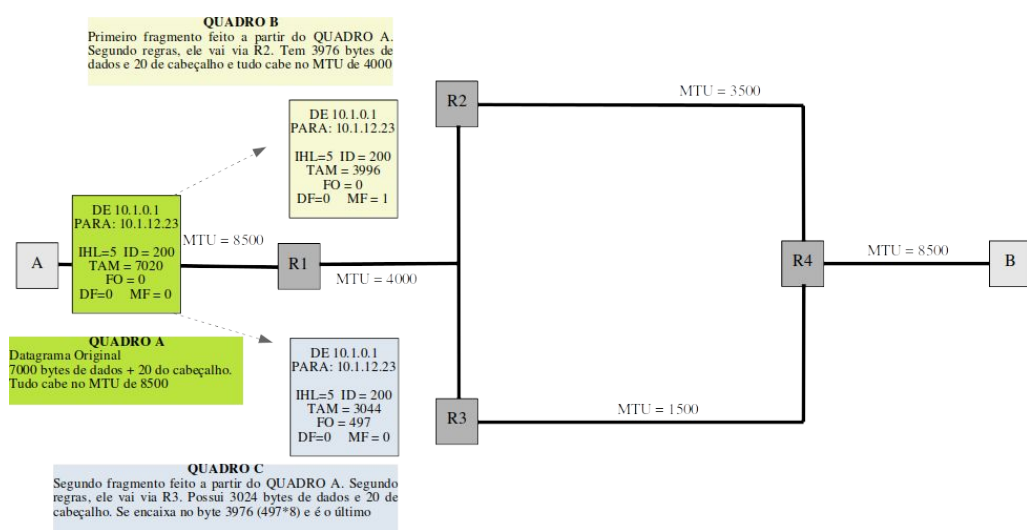
- Datagrama ID = 345 de 2000 bytes precisou ser fragmentado
- Primeiro fragmento:
 - ID = 345
 - FO=0 (se encaixa no byte 0 do datagrama original)
 - MF=1 (não é o último)
 - Tamanho = 1220 (1200 de dados + 20 cabeçalho)
- Segundo fragmento
 - ID = 345
 - FO= 150 ($150 * 8 = 1200$. Se encaixa no byte 1200)
 - MF=0 (é o último)
 - Tamanho = 820 (800 de dados + 20 cabeçalho)

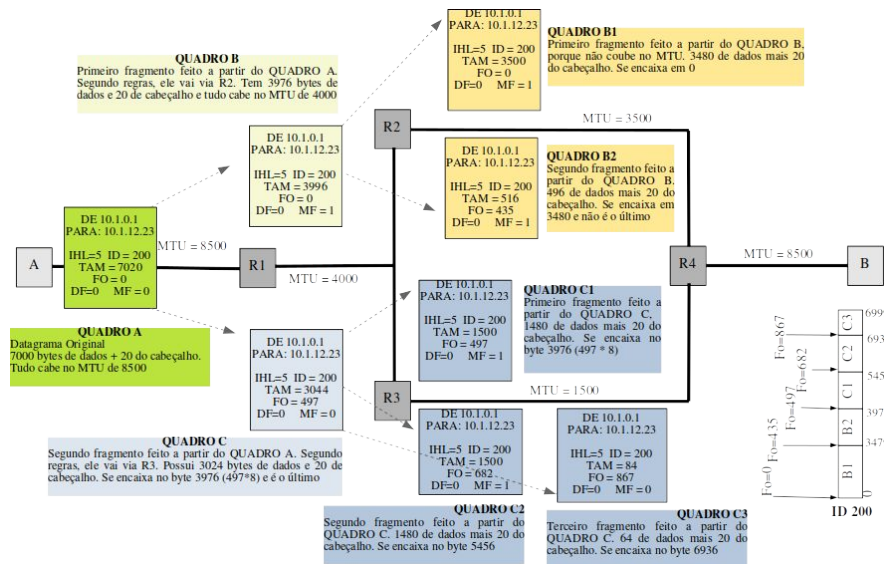
Fragmentação Ipv4

- Suporta chegada dos fragmentos fora de ordem
 - O primeiro é identificado por ter FO=0
 - O último é identificado por ter MF=0
- Cada fragmento é independente, pode ser roteado de forma individual
- Apenas o destino remonta os fragmentos
 - Em um buffer do nível de rede (IP)
 - Se chegou todos, envia para transporte
 - Se não chegou e time out: DESCARTA tudo

Exemplo







Roteamento no IPv4

- Problema: como rotear
 - De alguma forma a máquina sabe que destino não é local e repassa ao gateway
 - através da padronização dos números Ips alocados
- Classes de Ip (não mais usada)
 - Ip dividido em duas partes: Rede e máquina
 - Na parte de máquina, sempre:
 - todos os bits de host em ZERO determina o número da rede
 - todos os bits de host em UM determina *broadcast* de IP

Classe A

- 0RRRRRRR HHHHHHHH HHHHHHHH HHHHHHHH
- Primeiro bit em ZERO
 - outros 7 bits identificam a REDE
 - Total de 128 redes (rede 00000000 a 01111111)
 - Ips: 0.x.x.x até 127.x.x.x são classe A
 - 24 bits identificam máquinas (total de 16.777.216 - 2 máquinas)
 - Exemplo: Classe A 5.x.x.x
 - Endereço de rede: 5.0.0.0
 - Endereço de *broadcast*: 5.255.255.255

Classe B

- 10RRRRRR RRRRRRRR HHHHHHHH HHHHHHHH
- Primeiros 2 bits 10 identificam classe
 - Outros 14 bits identifica a rede
 - total de 16384 redes
 - rede de 128.x.x.x até 191.x.x.x
 - 16 bits máquinas (65536 - 2 máquinas em cada rede)

Classe C

110RRRRR RRRRRRRR RRRRRRRR HHHHHHHH

- Primeiros 3 bits 110 id classe
- Outros 21 bits o número da rede (2097152 redes)
 - de 192.x.x.x até 223.x.x.x
- 8 bits para máquinas (256 máquinas - 2 = 254 hosts)

Classe D e E

1110XXXX XXXXXXXX XXXXXXXX XXXXXXXX

- Reservado para multicast
- não usado para máquinas na Internet
- de 224.x.x.x até 239.x.x.x

1111XXXX XXXXXXXX XXXXXXXX XXXXXXXX

- Reservado para uso futuro

Roteamento por classes

- Uso de classes facilita roteamento
 - diminui os custos
 - não necessita tabelas muito grandes
- em pouco tempo:
 - todos classe B usados
 - classe A muito grande e C muito pequeno para maioria das empresas
 - Apenas classe C disponíveis
- necessário redividir Ips

CIDR - Classless Inter-Domain Routing

- Introduzido em 1993
- Não fixar o número de bits para rede.
- Qualquer ip, não importa de qual classe seria, pode ter quantos bits quiser para rede.
- O que é rede e o que é host?
 - R bits mais significativos são para rede
 - H bits restantes menos significativos são para host
- Quantos bits?
 - sem classes, necessário especificar

Máscara de rede

- IP + Máscara
- Roteamento por tabelas
- Cada equipamento deve saber Seu IP e sua rede (pela máscara)
- Rede expressa em máscara: números de 1's usados
- Exemplo:
 - Ip 192.168.10.10 pertence a rede 192.168.10.0 (como um classe C).
 - Dos 32 bits, 24 designam a rede e 8 a host
 - diz-se que é 192.168.10/24 (24 bits para rede)

```
IP: 11000000 10101000 00001010 00001010
    [----- Rede ----->] [<Host>]
```

Máscara de rede

- Exemplo:
 - diz-se que é 192.168.10/24 (24 bits para rede)
- ```
IP: 11000000 10101000 00001010 00001010
 [----- Rede ----->] [<Host>]
```
- na máscara representa os bits de rede em 1 e os de host em 0
- ```
IP: 11000000 10101000 00001010 00001010
BITS 11111111 11111111 11111111 00000000
      255      255      255      0
```
- E olha ai de onde vem o 255.255.255.0

Porque se chama máscara?

- Facilita operações
- cálculo da rede e de broadcast são feitos por operações binárias
- Exemplo:
 - Qual rede pertence o IP 192.168.10.50/24
 - isto e, 24 bits iniciais designam a rede) ?

```
IP: 11000000 10101000 00001010 00110010 (192.168.10.50)
REDE: 11111111 11111111 11111111 00000000 (255.255.255.0)
AND: 11000000 10101000 00001010 00000000 (192.168.10.0)
```