Solucionar problemas de desempenho de TCP no Nexus 9000 (NX-OS)

Contents

Introdução

Este documento descreve os fundamentos do TCP, a análise profunda de pacotes do Wireshark e a solução prática de problemas para otimizar o desempenho de ponta a ponta.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- IP/TCP

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Escala de nuvem do Cisco Nexus 9000 com o Cisco NX-OS 10.6(X).

Note: Qualquer dúvida sobre a configuração e a interoperabilidade de software ou hardware de terceiros está fora do suporte da Cisco. O uso de ferramentas de terceiros é o melhor esforço para demonstrar sua configuração e operação com o equipamento da Cisco.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

O que é TCP

O Transmission Control Protocol (TCP) é um protocolo fundamental da camada de transporte que opera na camada 4 do modelo OSI e fornece entrega confiável, ordenada e verificada por erro de um fluxo de bytes entre aplicativos que se comunicam por uma rede IP.

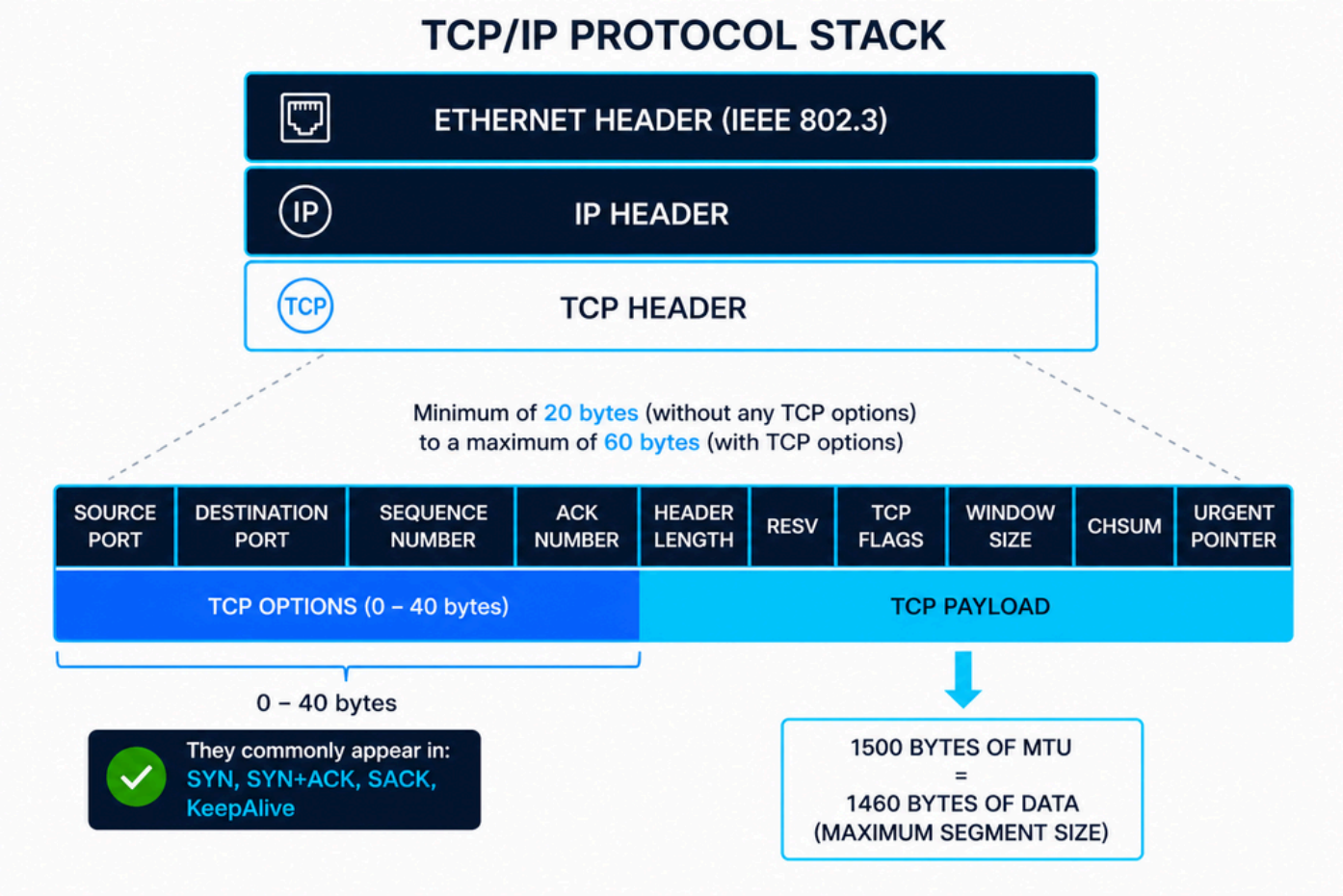
Três benefícios principais

1. Confiabilidade: o TCP é orientado a conexão e garante a entrega exigindo confirmações do receptor. Se um pacote for perdido ou corrompido durante a transmissão, o TCP retransmite automaticamente os dados para garantir que eles cheguem ao destino.
2. Entrega ordenada: Como os pacotes de rede podem chegar fora de ordem, o TCP atribui números de sequência a cada segmento. Isso permite que o sistema receptor remonte os dados na ordem exata em que foram enviados originalmente.
3. Controle de fluxo e de congestionamento: O TCP gerencia dinamicamente a taxa de transmissão de dados para corresponder à capacidade de processamento dos receptores e às condições atuais da rede, evitando a perda de dados causada por estouros de buffer ou congestionamento da rede.

Visão geral do encapsulamento TCP/IP

O diagrama representa a pilha TCP/IP onde um segmento TCP (Camada 4) é encapsulado dentro de um pacote IP (Camada 3) e depois dentro de um quadro Ethernet (Camada 2) definido pelo IEEE 802.3. Essa abordagem em camadas garante a comunicação modular, onde cada camada adiciona suas próprias informações de controle (cabeçalhos) para garantir a entrega, o

roteamento e a integridade dos dados.



Conector Ethernet (IEEE 802.3)

O cabeçalho Ethernet é normalmente de 14 bytes, composto por:

- Endereço MAC de Destino (6 bytes)
- Endereço MAC de Origem (6 bytes)
- EtherType/Length (2 bytes)

Além disso, os quadros Ethernet incluem um trailer FCS (Frame Check Sequence) de 4 bytes para detecção de erros na Camada 2. O IEEE 802.3 define o enquadramento, os tamanhos mínimo/máximo de quadros e as restrições de entrega física que afetam diretamente os protocolos das camadas superiores, como o TCP.

Cabeçalho IP (IPv4)

O cabeçalho IPv4 tem um tamanho mínimo de 20 bytes, extensível até 60 bytes com opções. Os

principais campos incluem:

- Endereços IP de origem e de destino
- Vida Útil (TTL)
- Protocolo (identifica o TCP como payload)

A camada IP é responsável pelo endereçamento lógico e pelo roteamento através das redes, mas não garante a confiabilidade.

Estrutura do cabeçalho TCP

O cabeçalho TCP varia de 20 a 60 bytes, dependendo das opções. Os principais campos incluem:

- Portas de origem/destino
- Número de seqüência
- Número de confirmação
- Sinalizadores (SYN, ACK, FIN, RST, etc.)
- Tamanho da janela
- Checksum

O TCP adiciona entrega confiável, sequenciamento adequado e controle de fluxo à comunicação IP.

Opções de TCP (Comum 10)

As opções de TCP estendem o protocolo base. Os mais comuns incluem:

1. Maximum Segment Size (MSS) - Define o maior payload de TCP que um host pode aceitar.
2. Escala de janela - Estende a janela de recebimento além de 65.535 bytes.
3. Confirmação seletiva permitida (SACK permitido) - habilita o recurso de confirmação seletiva.
4. Confirmação seletiva (SACK) - Especifica blocos de dados recebidos para evitar retransmissões completas.
5. Carimbos de data/hora - Usados para cálculo de RTT e Proteção contra números de sequência empacotados (PAWS).
6. Sem Operação (NOP) - Preenchimento para alinhamento de opções.
7. End of Option List (EOL) - Marca o fim das opções TCP.
8. TFO (TCP Fast Open) - Permite a troca de dados durante o handshake inicial.
9. Multipath TCP (MPTCP) - Ativa vários caminhos de rede para uma única sessão TCP.

10. User Timeout Option (UTO) - Controla por quanto tempo os dados transmitidos podem permanecer sem confirmação.

Sequência TCP e comportamento de confirmação (incluindo SYN/FIN)

As flags SYN e FIN consomem um número de sequência, mesmo quando não há payload presente. O TCP opera usando um modelo de sequenciamento orientado a bytes, onde cada byte transmitido - e flags de controle específicos - avançam o espaço de sequência. Esse comportamento é essencial para a análise precisa do TCP em capturas de pacotes e para diagnosticar inconsistências de sequência ou de confirmação.

ACK = SEQ + Comprimento de payload + (SYN ? 1: 0) + (FIN ? 1: 0)

Where:

- SEQ = Número de sequência inicial
- Comprimento do payload = Tamanho dos dados em bytes
- SYN ? 1: 0 = Adiciona 1 se o sinalizador SYN estiver definido, caso contrário 0
- FIN ? 1: 0 = Adiciona 1 se o sinalizador FIN estiver definido; caso contrário, 0
- ACK = Próximo byte esperado

Exemplo 1: SYN com Dados (TCP Fast Open)

- SEQ = 1000
- SYN = 1
- Comprimento de payload = 200 bytes

Cálculo ACK:

- ACK = 1000 + 200 + 1 + 0 = 1201

Isso reflete um cenário em que os dados são enviados durante o handshake TCP. O payload e o flag SYN consomem espaço de sequência.

Exemplo 2: FIN com Dados (Terminação de Conexão)

- SEQ = 3000
- FIN = 1

- Comprimento de payload = 150 bytes

Cálculo ACK:

- ACK = 3000 + 150 + 0 + 1 = 3151

Isso mostra que o TCP pode incluir dados durante a desativação da conexão e a carga útil e o flag FIN incrementam o número de sequência.

MSS e seu relacionamento com MTU

O Tamanho Máximo de Segmento (MSS) define o payload máximo que o TCP pode enviar em um segmento.

- MTU Ethernet típica = 1500 bytes
- MSS = MTU − cabeçalho IP − cabeçalho TCP
- MSS padrão = 1460 bytes (1500 − 20 − 20)

Se as opções TCP estiverem presentes, o MSS é reduzido de acordo. O MSS é negociado durante o handshake triplo do TCP e evita a fragmentação na camada IP.

Como funciona a negociação MSS no handshake triplo do TCP

O Maximum Segment Size (MSS) é trocado durante o handshake triplo do TCP usando a opção MSS em pacotes SYN:

- Host A → Host B (SYN): anuncia seu MSS (por exemplo, 1460)
- Host B → Host A (SYN-ACK): anuncia seu MSS (por exemplo, 1380)

Cada lado está dizendo efetivamente:

Esta é a maior carga TCP aceita.

Regra-chave: MSS é Direcional

O MSS não é negociado como um valor único acordado.

Em vez disso:

- Cada host usa o MSS anunciado pelo outro lado.
- Isso cria dois limites independentes, um por direção.

Portanto:

- A envia dados usando o MSS de B.
- B envia dados usando o MSS de A.

A origem pode enviar mais payload de TCP do que o MSS de destino?

Em uma pilha TCP funcionando corretamente: No.

- O remetente deve respeitar o MSS anunciado pelo receptor.
- O envio de segmentos maiores poderia arriscar:
  - Fragmentação de IP (se a MTU for excedida)
  - Descartes de pacotes (se a fragmentação estiver bloqueada ou não for suportada)
- Isso leva a:
  - Retransmissões
  - Degradação do desempenho
  - Problemas como buracos negros do PMTUD (Path MTU Discovery)

Percepção prática para solução de problemas

- Sempre verifique os valores de MSS no handshake triplo TCP (pacotes SYN/SYN-ACK).
- Verificar incompatibilidades causadas por:
  - Túneis (VXLAN, GRE, IPsec)
  - Firewalls modificando MSS (fixação de MSS)
- Em plataformas como o Cisco NX-OS, o ajuste de MSS é frequentemente usado para evitar a fragmentação em caminhos encapsulados

Tamanho da Janela (Controle de Fluxo)

O Tamanho da janela define a quantidade de dados que o receptor pode aceitar sem confirmação.

O que é:

- Um mecanismo de controle de fluxo para evitar estouro de buffer.

Propósito:

- Garante que o remetente não sobrecarregue o receptor.

Onde obtê-lo:

- Visível em capturas de pacotes (por exemplo, Wireshark).
- Derivado da configuração da pilha TCP do SO e do tamanho do buffer.

Variabilidade de fornecedor/SO:

- Diferentes implementações (Linux, Windows, Cisco NX-OS) usam ajuste dinâmico de escala e buffer, levando a tamanhos de janela variados.

Condição zero de janela:

- Quando Tamanho da janela = 0, o buffer do receptor está cheio.
- O remetente pausa a transmissão e envia sondas periódicas.

Mecanismos variáveis do Windows

- Controle de fluxo baseado em taxa
    - Ele atribui ao remetente uma taxa de dados fixa e garante que os dados nunca excedam essa alocação.
    - Ideal para aplicativos de streaming.
    - Entrega de broadcast e multicast
- Controle de fluxo baseado em janela
    - O tamanho da janela varia com o tempo.
    - O receptor obtém o controle de fluxo sinalizando a janela permitida para as atualizações da janela do remetente.

Solução de problemas de uso:

- Janelas pequenas ou zero → afunilamento do lado do receptor (CPU, memória, aplicativo).
- Janelas grandes, mas baixa taxa de transferência → problemas de rede (latência, congestionamento).
- A análise do comportamento da janela é crítica para diagnosticar problemas de desempenho em sessões TCP.

Solução de problemas de plano de dados TCP no Cisco Nexus 9000 (NX-OS)

Esta seção descreve uma metodologia prática para diagnosticar se um switch Cisco Nexus executando NX-OS está afetando o encaminhamento de tráfego TCP ou apresentando problemas de desempenho. A abordagem é apresentada através de um cenário hipotético.

Quando a latência ou a degradação de desempenho do TCP é observada, é comum suspeitar inicialmente que a rede está causando isso. No entanto, essa suposição deve ser validada por meio da análise orientada por dados. O método oficial para a identificação e solução de problemas de TCP é a captura de pacotes, executada de maneira ideal:

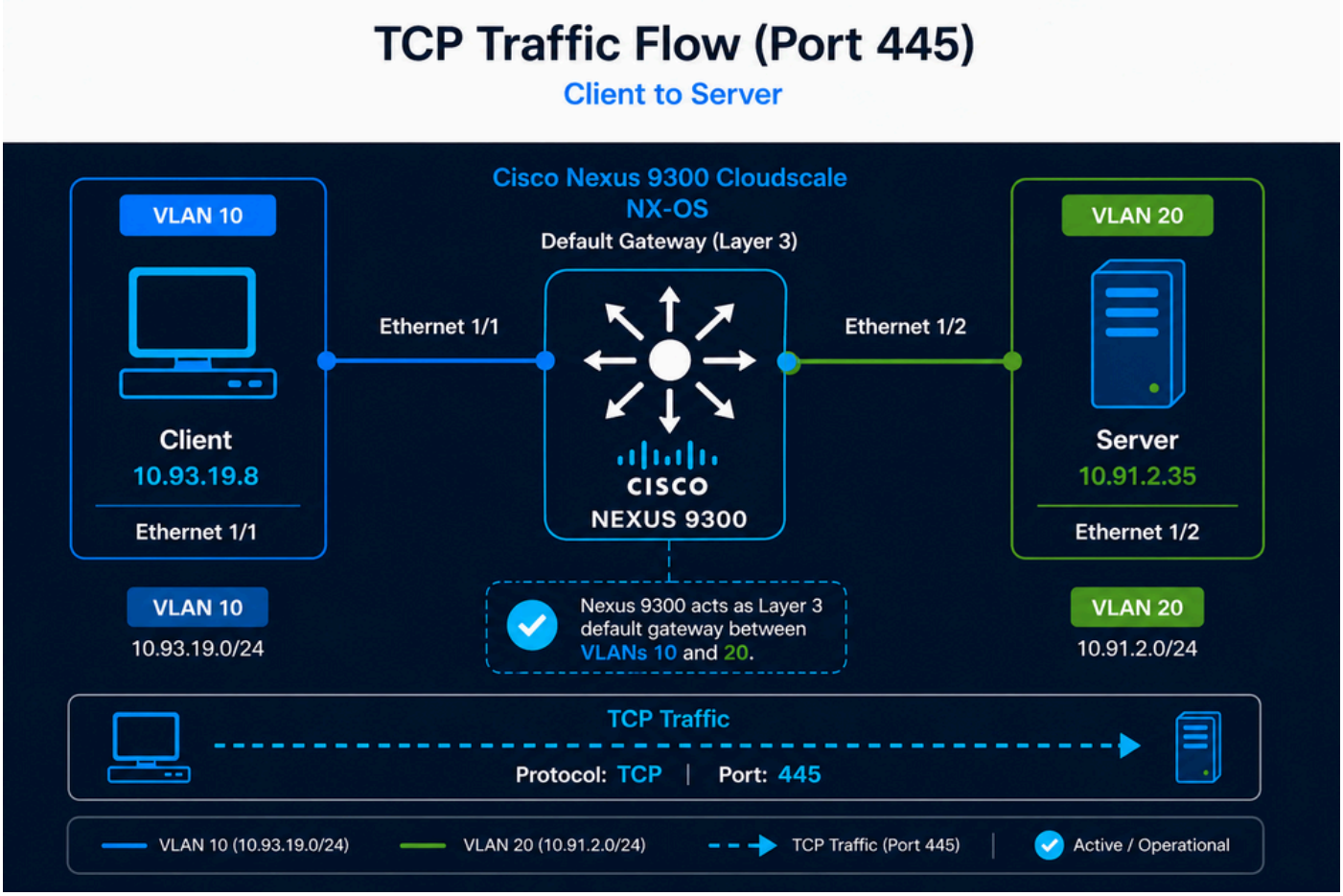
- Simultaneamente na origem e no destino
- Antes do início do tráfego

Isso garante visibilidade no handshake triplo TCP, onde parâmetros críticos como MSS, Escala de Janela e SACK são negociados e não repetidos posteriormente na sessão. Se capturas simultâneas não forem possíveis, a análise poderá prosseguir com uma única captura, mas as conclusões serão limitadas.

Definição de Cenário

Um usuário identificou que o processo de backup de um conjunto de dados de aplicativo de aproximadamente 7,5 TB, que antes era concluído em cerca de 9 horas, agora leva quase 21 horas. Embora as sessões de TCP entre o cliente e o servidor ainda sejam estabelecidas com êxito, o aumento significativo na duração do backup sugere uma possível degradação no throughput ou no desempenho geral do TCP. Como o switch Nexus é o único dispositivo de rede no caminho e também fornece funcionalidade de gateway de Camada 3, o administrador de rede suspeita que o switch Nexus seja a causa do problema.

- Cliente: 10.93.19.8 (VLAN 10)
- Servidor: 10.91.2.35 (VLAN 20)
- Nexus 9300 atuando como gateway padrão
- Porta TCP 445



Validação inicial (acessibilidade)

- Esses comandos são usados para validar a MTU de Caminho (PMTU) entre uma origem e um destino, enviando pacotes ICMP com o conjunto de bits Não Fragmentar (DF). Isso ajuda a determinar o tamanho máximo do pacote que pode atravessar a rede sem fragmentação. Esse processo deve ser executado na origem e no destino.
- Sempre verifique a MTU da interface física na origem e no destino.
- Neste cenário, o acesso está disponível apenas para o host de origem, onde um MTU de 1500 foi identificado.

```
Linux: ping -c 10 -I 10.93.19.8 -s 1472 -M do 10.91.2.35
```

- -c 10 → Envia 10 solicitações de eco ICMP
- -I 192.168.10.10 → Usa esse IP/interface de origem específico
- -s 1472 → Define o tamanho de payload ICMP para 1472 bytes
- -M do → Define o bit DF (Não Fragmentar)
- 192.168.20.20 → IP de destino

```
Windows: ping -n 10 -l 1472 -f 10.91.2.35
```

- -n 10 → Envia 10 solicitações de eco ICMP
- -l 1472 → Define o tamanho de payload ICMP para 1472 bytes
- -f → Define o sinalizador Não Fragmentar (DF)
- 192.168.20.20 → IP de destino

Por que 1472 bytes?

- payload de ICMP = 1472 bytes
- cabeçalho IP = 20 bytes
- cabeçalho ICMP = 8 bytes
- Tamanho total do pacote: 1472 + 20 + 8 = 1500 bytes (MTU padrão)
- Isso testa se o caminho suporta um MTU de 1.500 bytes sem fragmentação. Se você tentar enviar 1500 bytes de payload ICMP, o ping poderá falhar porque o tamanho total do pacote excederia o MTU padrão após a adição dos cabeçalhos IP e ICMP.

O que pode ser concluído

- Se o ping for bem-sucedido (sem perda de pacotes), o caminho suportará pelo menos uma MTU de 1.500 bytes e nenhuma fragmentação será necessária.
    - Limpar resultados ICMP → prosseguir para a análise TCP
    - Êxito do ping intermitente → possível perda de pacotes, congestionamento transitório, limitação de taxa ou um problema de encaminhamento; prossiga para a análise de perda de pacotes, já que o TCP requer um caminho livre de perdas para operar com eficiência.
- Se o ping falhar com o erro "Fragmentation needed" (Fragmentação necessária) ou expirar, se houver um link no caminho com MTU inferior a 1500 bytes, o pacote não poderá ser encaminhado devido ao bit DF e isso indica um problema de MTU de caminho.

Como usar isso para solução de problemas

- Reduza gradualmente o tamanho do payload (por exemplo, 1472 → 1400 → 1300) para identificar o maior tamanho bem-sucedido.
- Uma vez identificado, calcule o MTU usando a fórmula MTU = payload + 28 bytes (cabeçalhos IP + ICMP).

Relevância prática para o TCP

- Se o MTU for menor que o esperado, os segmentos TCP podem ser fragmentados ou

descartados.
- Isso leva a retransmissões, maior latência e throughput reduzido, com impacto direto no desempenho do aplicativo.

Identificando o caminho de tráfego (interfaces)

Para solucionar com eficiência problemas de desempenho do TCP em um switch Cisco Nexus 9000, é essencial determinar quais interfaces estão recebendo e encaminhando o tráfego entre a origem e o destino.

Em topologias simples, isso pode ser inferido diretamente das conexões físicas. Por exemplo, se o cliente estiver conectado à Ethernet1/1 e o servidor à Ethernet1/2, o caminho do tráfego será direto. No entanto, em ambientes reais com várias interfaces ativas, canais de porta ou configurações de vPC, essa identificação nem sempre é trivial.

Nesses casos, a abordagem recomendada é usar o Embedded Logic Analyzer Module (ELAM), que fornece visibilidade no nível ASIC (data-plane hardware).

O ELAM permite capturar um pacote à medida que ele é processado pelo pipeline de encaminhamento e revela informações críticas como:

- Interface de entrada
- Interface de saída
- Decisão de encaminhamento (resultado da pesquisa de L2/L3)

Esse método é significativamente mais preciso do que contar com ferramentas de plano de controle, pois reflete o caminho real de encaminhamento de hardware.

É importante observar que o ELAM captura apenas um pacote por vez, portanto, os critérios de filtragem devem ser definidos precisamente para corresponder ao tráfego desejado (por exemplo, IP de origem, IP de destino, porta TCP). Se os filtros forem muito amplos, há um risco de capturar tráfego não relacionado, como ICMP ou UDP, em vez do fluxo TCP pretendido.

Além disso, esse processo deve ser repetido para ambos os sentidos de tráfego:

- Origem → Destino
- Destino → Origem

Em ambientes que usam vPC ou ECMP, o tráfego pode ter a carga balanceada através de vários caminhos. Como resultado, o tráfego de encaminhamento e de retorno pode atravessar diferentes

switches ou interfaces. Nesses cenários, o ELAM deve ser executado em cada switch Nexus relevante para garantir visibilidade completa.

Ao identificar com precisão as interfaces de entrada e saída, o escopo da solução de problemas é reduzido significativamente, permitindo a validação focada de contadores de interface, políticas de QoS, configurações de MTU e possíveis pontos de congestionamento ao longo do caminho de encaminhamento exato.

Configuração do ELAM (escala de nuvem do Nexus 9300)

Este exemplo filtra o tráfego com o IP origem 10.93.19.8, o IP destino 10.91.2.35 e a porta destino TCP 445.

Configuração do ELAM

<#root>

```
switch#

debug platform internal tah elam


switch(TAH-elam)#

trigger init


Slot 1: param values: start asic 0, start slice 0, lu-a2d 1, in-select 6, out-select 0
switch(TAH-elam-insel6)#

set outer ipv4 src_ip 10.93.19.8


switch(TAH-elam-insel6)#

set outer ipv4 dst_ip 10.91.2.35


switch(TAH-elam-insel6)#

set outer l4 l4-type 0


switch(TAH-elam-insel6)#

set outer l4 dst-port 445
```

```
switch(TAH-elam-insel6)#

start
```

Depois de gerar o tráfego, recupere o resultado:

<#root>

```
switch(TAH-elam-insel6)#

report
```

Captura de tráfego reverso (obrigatório para visibilidade total)

Para validar o caminho de retorno, repita a configuração trocando os endereços IP origem e destino:

<#root>

```
switch#

debug platform internal tah elam


switch(TAH-elam)# trigger init
Slot 1: param values: start asic 0, start slice 0, lu-a2d 1, in-select 6, out-select 0
switch(TAH-elam-insel6)#

set outer ipv4 dst_ip 10.93.19.8


switch(TAH-elam-insel6)#

set outer ipv4 src_ip 10.91.2.35


switch(TAH-elam-insel6)#

set outer l4 l4-type 0


switch(TAH-elam-insel6)#
```

```
set outer l4 dst-port 445
```

```
switch(TAH-elam-insel6)#
```

```
start
```

Notas operacionais

- O ELAM captura apenas um pacote, portanto, assegure-se de que o tráfego esteja fluindo ativamente ao iniciar a captura.
- Os filtros devem ser precisos para evitar a captura de tráfego não relacionado.
- Em ambientes vPC, execute o ELAM em ambos os switches, já que o tráfego pode ter hash diferente em cada direção.
- A saída exibe a interface de entrada, a interface de saída e a decisão de encaminhamento no hardware, fornecendo visibilidade autorizada no plano de dados.

Referência

Guia de ELAM do Cisco Nexus 9000 Cloud Scale ASIC

Validação no nível da interface

A validação no nível da interface garante que o switch Nexus não esteja introduzindo nenhuma restrição ou anomalia que afete o tráfego TCP. O foco é confirmar se a configuração, o estado operacional e os contadores de hardware estão consistentes com o comportamento esperado para o encaminhamento de plano de dados de alto desempenho.

Validação da configuração

- Verifique se nenhuma ACL restritiva está aplicada às interfaces:

<#root>

```
switch#
```

```
show running-config interface ethernet1/1-2 | include access-group
```

- Valide se nenhuma política de QoS não intencional está afetando o tráfego (nível de interface e QoS global, incluindo enfileiramento, policiamento e modelagem):

<#root>

switch#

show running-config interface ethernet1/1-2 | include service-policy

<#root>

switch#

show policy-map interface ethernet1/1-2

<#root>

switch#

show policy-map

<#root>

switch#

show class-map

<#root>

switch#

show class-map type network-qos

<#root>

switch#

show policy-map type network-qos

<#root>

switch#

show policy-map system type network-qos

<#root>

switch#

show queuing interface ethernet1/1-2

<#root>

switch#

show policy-map type queuing

- Confirme a configuração da Camada 2 ou Camada 3 (switchport versus interface roteada), incluindo associação de VLAN, estado de STP e endereçamento IP:

<#root>

switch#

show running-config interface ethernet1/1-2

<#root>

switch#

show interface ethernet1/1-2 switchport

<#root>

switch#

show spanning-tree interface ethernet1/1-2

<#root>

switch#

show ip interface ethernet1/1-2

Validação do Estado Operacional

- Verifique a consistência da MTU e certifique-se de que ela corresponda à configuração esperada (por exemplo, 1500 ou 9000 bytes):

<#root>

switch#

show interface ethernet1/1-2 | include MTU

- Confirme as configurações de velocidade e duplex da interface:

<#root>

switch#

show interface ethernet1/1-2 | include speed|duplex

- Validar a estabilidade da interface (sem oscilação ou transições frequentes de link):

<#root>

switch#

```
show interface ethernet1/1-2 | include rate|flap
```

Validação do Contador de Erros

- Limpar contadores antes do teste:

<#root>

```
switch#
```

```
clear counters interface all
```

- Monitorar contadores de erro (somente valores diferentes de zero):

<#root>

```
switch#
```

```
show interface counters errors non-zero | include Port|Eth1/1|Eth1/2
```

Validação pós-teste

- Execute novamente o teste de tráfego TCP e observe os contadores novamente:

<#root>

```
switch#
```

```
show interface counters errors non-zero | include Port|Eth1/1|Eth1/2
```

- Os contadores não devem incrementar; qualquer aumento indica possíveis problemas relacionados à Camada 1 ou ao hardware, como erros de link físico, erros de CRC/FCS ou sobrecargas/quedas de buffer.

Roteamento e estabilidade ARP

Garantir o roteamento e a estabilidade do ARP é fundamental para confirmar se o switch Nexus

tem acessibilidade consistente da Camada 3 e não está introduzindo problemas de resolução intermitente que possam afetar o desempenho do TCP. A instabilidade nas entradas de roteamento ou na resolução ARP pode levar à perda de pacotes, aumento da latência ou blackholing de tráfego.

Critérios de validação

- As entradas de roteamento para origem e destino devem estar presentes, estáveis e não devem ser alteradas com frequência.
- As entradas ARP devem ser resolvidas e não continuamente atualizadas ou ausentes.

<#root>

```
switch#

show ip route 10.93.19.8
```

<#root>

```
switch#

show ip route 10.91.2.35
```

<#root>

```
switch#

show ip arp detail | include 10.93.19.8
```

<#root>

```
switch#

show ip arp detail | include 10.91.2.35
```

Verificando se o tráfego não é enviado para a CPU

Nos switches Cisco Nexus 9000, o encaminhamento é executado no hardware (ASIC) e a CPU não está envolvida em operações normais de plano de dados. Portanto, observar o tráfego TCP host a host no plano de controle é anormal e indica que os pacotes estão sendo lançados devido a exceções ou configurações incorretas. Uma vez que o tráfego deve ser processado pela CPU, ele fica sujeito à Política de Plano de Controle e espera-se que quedas possam ser observadas se o tráfego exceder a taxa de plano de controle permitida.

Método de validação

- Capturar o tráfego que chega ao plano de controle usando o Ethanalyzer:

<#root>

```
switch#

ethanalyzer local interface inband display-filter "ip.addr==10.93.19.8 and ip.addr==10.91.2.35" limit-ca
```

Comportamento esperado

- Nenhum tráfego de plano de dados TCP host a host pode ser observado na CPU.

Comportamento inesperado

- Se os pacotes correspondentes ao fluxo estiverem visíveis, o tráfego será lançado, o que pode ser causado por:
  - Tratamento de pacotes excepcional (expiração de TTL, registro de ACL, redirecionamentos)
  - Configuração incorreta ou recursos sem suporte
  - Programação de hardware incorreta

Determinando a latência de encaminhamento de pacotes

A latência de encaminhamento de pacotes nos switches Nexus 9000 depende do tamanho do pacote, do modo de encaminhamento e dos recursos ativados. As especificações da Cisco normalmente fazem referência à latência para pacotes de 64 bytes em encaminhamento cut-through.

```
+----------------------+----------------------+-------------------------+------------------------------
| Switch Model         | ASIC / Architecture  | Ports (example config)  | Typical Forwarding Latency
|                      |                      |                         | (64B packet)
```

Nexus 93180YC-EX	Cloud Scale (EX)	48x25G + 6x100G	~1.0 - 1.2 microseconds
Nexus 93180YC-FX	Cloud Scale (FX)	48x25G + 6x100G	~0.9 - 1.0 microseconds
Nexus 93180YC-FX2	Cloud Scale (FX2)	48x25G + 6x100G	~0.8 - 0.9 microseconds
Nexus 9364C	Cloud Scale	64x100G	~1.0 microsecond
Nexus 9336C-FX2	Cloud Scale (FX2)	36x100G	~0.8 microseconds
Nexus 93240YC-FX2	Cloud Scale (FX2)	48x25G + 12x100G	~0.8 - 0.9 microseconds
Nexus 92300YC	Broadcom Trident II	48x10/25G + 6x40/100G	~2 - 3 microseconds
Nexus 92160YC-X	Broadcom Tomahawk	48x25G + 6x100G	~2 microseconds

- Encaminhamento cut-through (padrão no Nexus 9000):
  - Inicia o encaminhamento antes do recebimento do pacote completo.
  - Minimiza a latência (submicrossegundo a ~1 µs).
- Armazenamento e encaminhamento:
  - O pacote inteiro deve ser recebido antes de ser encaminhado.
  - Adiciona latência proporcional ao tamanho do pacote.

Recursos adicionais podem introduzir latência incremental:

- Encapsulamento/desencapsulamento de VXLAN
- Pesquisas de ACL (processamento de TCAM)
- Classificação e enfileiramento de QoS
- Telemetria (NetFlow, ERSPAN, sFlow)
- Buffer durante congestionamento

No entanto:

- Essas operações são executadas em pipelines de hardware.

O único cenário realista onde a latência aumenta visivelmente é o congestionamento:

- Os pacotes são colocados em buffer nas filas de saída.
- O atraso depende de:
  - Profundidade de Fila
  - Utilização da interface
  - Políticas de QoS

Mesmo nestes casos:

- A latência geralmente está no intervalo de microssegundos a centenas baixas de microssegundos.
- O atraso de nível de milissegundo sustentado implicaria em:

- Congestionamento grave
- Sobrescrita
- QoS ou buffering configurados incorretamente

SPAN para CPU (captura de pacote para plano de dados)

Isso permite o espelhamento do tráfego do plano de dados no plano de controle para a captura e exportação de pacotes para um arquivo .pcapng, permitindo análise detalhada no Wireshark.

Configuração

```
monitor session 1
 source interface ethernet1/1 both
 source interface ethernet1/2 both
 destination interface sup-eth0
 no shut
```

Execução de Captura

<#root>

```
switch#
```

ethanalyzer local interface inband mirror capture-filter "tcp port 445" limit-capture 0 write bootflash:

Considerações técnicas

- O tráfego espelhado para a CPU está sujeito à Política de Plano de Controle (CoPP).
- Se o tráfego exceder CoPP:
  - Os pacotes só podem ser descartados no plano de controle.
  - Isso cria falsos positivos durante a análise.
- O SPAN para a CPU é recomendado para cenários de tráfego de baixa a moderada.
- Para ambientes de alto throughput:
  - Usar SPAN local (analisador externo)
  - Usar ERSPAN para captura remota

Método	Vantagem	Limitação
SPAN	Precisa, sem encapsulamento	Requer conexão física.

Método	Vantagem	Limitação
ERSPAN	Recurso de captura remota	Susceptível a congestionamento na rede.

Validação da limitação da taxa do plano de controle

Para garantir que as capturas de SPAN para CPU sejam confiáveis, é necessário validar se o plano de controle não está descartando pacotes espelhados devido à limitação de taxa.

Comando de validação

```
switch(config)# show hardware rate-limiter | i Allowed|span

Allowed, Dropped & Total: aggregated bytes since last clear counters

R-L Class     Config Allowed Dropped Total

span          50           0       0     0 <<<

span-egress   disabled     0       0     0
```

Metodologia de validação

- Execute o comando em intervalos de ~3 segundos.
- Observe os contadores de queda relacionados ao SPAN.

Interpretação

- Nenhum incremento nos contadores de queda da linha SPAN indica uma captura confiável.
- Contadores de queda crescentes indicam perda de pacotes no plano de controle, tornando a captura não confiável.

Se forem observadas quedas, o método de captura deverá ser alterado para SPAN ou ERSPAN.

Validação baseada em ICMP antes do TCP

O teste ICMP fornece uma validação de linha de base da integridade do plano de dados antes de executar uma análise TCP complexa. Como o ICMP é stateless e mais simples, ele permite a detecção rápida de perda de pacotes, duplicação ou inconsistências de caminho.

Comportamento esperado na captura de SPAN

- Cada pacote ICMP pode aparecer duas vezes:
  - Uma vez na entrada
  - Uma vez na saída
- Para um ping padrão:
  - Solicitação de Eco → 2 pacotes
  - Resposta de Eco → 2 pacotes

Isso confirma o encaminhamento correto e a ausência de perda de pacotes no plano de dados.

Comportamento Anormal

- A ausência de duplicatas ou contagens assimétricas de pacotes indicam a possível perda de pacotes ou limitações de captura.
- Os intervalos intermitentes sugerem problemas de Camada 1, congestionamento ou problemas de upstream.

Se o tráfego ICMP for consistentemente encaminhado sem perda, há uma alta probabilidade de que o tráfego TCP também esteja sendo encaminhado corretamente na Camada 2/3.

Determinando a latência de encaminhamento do switch Nexus usando a captura de pacotes

Quando o tráfego é capturado usando SPAN para a CPU (ou SPAN/ERSPAN), cada pacote pode ser observado duas vezes: once on ingress e once on egress. Essa duplicação pode ser usada para estimar a latência de encaminhamento introduzida pelo switch Nexus, calculando a diferença de tempo entre as duas instâncias do mesmo pacote.

Na prática, essa latência pode ser medida usando o tráfego ICMP capturado anteriormente, comparando o delta de tempo entre pacotes duplicados de Solicitação de Eco e de Resposta de Eco. Isso fornece uma linha de base simples e eficaz para o desempenho de encaminhamento de switch. Se uma análise mais profunda for necessária, a mesma metodologia pode ser aplicada ao tráfego TCP, capturando o fluxo e medindo a diferença de tempo entre os pacotes TCP duplicados.

Metologia

- Identificar um pacote e sua duplicata (mesmo número de sequência).
- Meça o delta de tempo entre as cópias de entrada e saída.
- Esse delta representa uma estimativa de limite superior da latência de encaminhamento de

switch, pois pode incluir o espelhamento e a sobrecarga de registro de tempo.

Configuração do Wireshark

- Habilitar exibição de delta de tempo:

```
View > Time Display Format > Seconds Since Previous Displayed Packet
```

- Adicionar uma coluna personalizada para o intervalo de tempo:

```
Right-click on "Time Delta from Previous Displayed Packet" → Apply as Column
```

- Filtrar tráfego relevante (exemplo):

```
ip.addr==10.93.19.8 and ip.addr==10.91.2.35 and tcp
```

- Classificar pacotes por número de sequência ou fluxo TCP:

```
Right-click packet → Follow → TCP Stream
```

Interpretação

- O delta de tempo entre pacotes duplicados pode estar no intervalo de microssegundos.
  - Se esse for o caso, o switch Nexus não está introduzindo latência para o encaminhamento de pacotes.
- Os deltas baixos consistentes confirmam o desempenho de encaminhamento baseado em hardware.
- Deltas maiores ou inconsistentes podem indicar:
  - Congestionamento ou buffering

Referências

- Dados técnicos do Cisco Nexus 9000 Series
- Guias de design dos switches Cisco Nexus 9000 Series

- White paper Gerenciamento inteligente de buffer em switches Cisco Nexus 9000 Series

Análise de tráfego TCP a partir da captura do pacote de host de origem

Esta seção fornece uma metodologia detalhada para analisar uma captura de pacote TCP no Wireshark, incluindo a configuração de perfil, através do caso hipotético descrito anteriormente. As imagens mostradas foram tiradas diretamente do Wireshark. Como lembrete, o cenário é:

Um usuário identificou que o processo de backup de um conjunto de dados de aplicativo de aproximadamente 6,5 TB, que antes era concluído em cerca de 9 horas, agora leva quase 21 horas. O único dispositivo de rede acessível é um switch Cisco Nexus 9300 conectado ao servidor de origem (10.93.19.8). O MTU configurado na interface do switch é de 9000 bytes (quadros jumbo), enquanto o MTU no servidor é desconhecido. Uma captura de pacote do servidor de origem está disponível e todas as etapas de validação do Nexus anteriores já foram concluídas sem nenhuma anomalia detectada.

Principais observações e restrições

- O switch Nexus foi excluído:
  - Nenhum descarte de pacote
  - Sem erros de interface
  - Sem impacto na QoS ou na ACL
  - Encaminhamento de hardware confirmado
- Configuração da interface:
  - Porta de acesso
  - MTU: 9000 bytes
- Dados disponíveis:
  - Captura de pacotes na origem
  - Conhecimento de MTU de ponta a ponta
    - O ping foi concluído com êxito sem fragmentação usando um pacote de 1.500 bytes com 1.472 bytes de dados.
- Dados ausentes:
  - Visibilidade de destino
  - Nenhuma captura de pacote está disponível no servidor de destino.

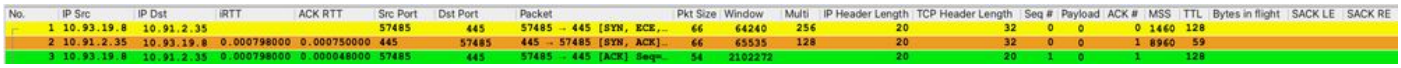
No Wireshark, você pode criar perfis personalizados adaptados ao tipo específico de análise que deseja executar.

Descrição da coluna

- tcp.analysis.initial_rtt (iRTT): Estima o tempo de ida e volta inicial com base no handshake triplo do TCP.
- tcp.analysis.ack_rtt (ACK RTT): Mede o tempo entre um segmento TCP e sua confirmação correspondente.
- tcp.window_size (Janela): Indica o tamanho da janela TCP anunciada do receptor antes da aplicação do dimensionamento.
- tcp.options.wscale.multiplier (Vários): Representa o fator de escala da janela usado para calcular a janela de recebimento efetivo.
- tcp.seq (Seq#): Exibe o número sequencial do primeiro byte no segmento TCP.
- tcp.len (Payload): Mostra o tamanho do payload TCP em bytes para esse segmento.
- tcp.ack (Nº ACK): Indica o próximo byte esperado do remetente (confirmação cumulativa).
- tcp.options.mss_val (MSS): Exibe o Tamanho máximo de segmento anunciado durante o handshake TCP.
- ip.ttl (TTL): Mostra o valor de Vida Útil, útil para identificar a contagem de saltos e o comportamento de roteamento.
- tcp.analysis.bytes_in_flight (Bytes em Voo): Representa a quantidade de dados não confirmados em trânsito no momento.

Análise do handshake triplo do TCP

A captura do handshake triplo TCP é obrigatória porque contém parâmetros críticos, como MSS, Escala de Janela e SACK, que definem como a sessão se comporta.
Sem essas informações, qualquer análise de TCP é incompleta e pode levar a conclusões incorretas sobre o desempenho ou a causa raiz.



No.	IP Src	IP Dst	iRTT	ACK RTT	Src Port	Dst Port	Packet	Pkt Size	Window	Multi	IP Header Length	TCP Header Length	Seq #	Payload	ACK #	MSS	TTL	Bytes in flight	SACK LE	SACK RE
1	10.93.19.8	10.91.2.35			57485	445	57485 → 445 [SYN, ECE,…	66	64240	256	20	32	0	0	0	1460	128			
2	10.91.2.35	10.93.19.8	0.000798000	0.000750000	445	57485	445 → 57485 [SYN, ACK]…	66	65535	128	20	32	0	0	1	8960	59			
3	10.93.19.8	10.91.2.35	0.000798000	0.000048000	57485	445	57485 → 445 [ACK] Seq=…	54	2102272		20	20	1	0	1		128			

Identificação de tráfego

A partir da captura de pacotes:

- Endereço IP origem: 10.93.19.8
- Endereço IP de destino: 10.91.2.35

Análise do tempo de ida e volta inicial (iRTT)

O RTT inicial (iRTT) é calculado do seguinte modo:

- iRTT = 798 microssegundos

Este valor é derivado de:

- Pacote 2 (SYN-ACK) ACK RTT: 750 µ → Tempo para que o destino responda ao SYN.
- Pacote 3 (ACK) ACK RTT: 48 µs → Tempo para a origem confirmar o SYN-ACK.

A maioria da latência (~94%) está no caminho de encaminhamento (cliente → servidor → cliente), enquanto o tempo de resposta da origem é mínimo, indicando que não há atraso de CPU ou aplicativo no cliente.

Identificação da porta TCP

- Porta TCP de destino: 445

A porta 445 corresponde ao Microsoft Server Message Block (SMB), normalmente usado para compartilhamento de arquivos, unidades de rede e serviços de autenticação do Windows. Esse protocolo é sensível à latência e ao throughput, tornando-o altamente dependente da eficiência do TCP e da estabilidade da rede.

Análise do Tamanho da Janela TCP

- Janela de Origem (dimensionada): 64,240 bytes
- Janela de destino: 65,535 bytes

A janela TCP representa a quantidade de dados que pode ser enviada antes de aguardar a confirmação. Nesse caso, a origem é um pouco mais restritiva que o destino. Esses valores são relativamente pequenos para ambientes modernos e podem limitar o throughput, especialmente à medida que o RTT aumenta.

O rendimento teórico máximo pode ser estimado utilizando:

Throughput = Tamanho da Janela TCP / RTT

Substituindo os valores observados:

- Tamanho da Janela TCP = 64.240 bytes
- RTT = 798 microssegundos = 0,000798 segundos

Rendimento ≈ 64.240 / 0,000798 ≈ 80,5 MB/s (~644 Mbps)

Isso representa o throughput do limite superior, considerando:

- Sem perda de pacotes
- Sem retransmissões
- Condições ideais de rede

Análise de throughput, tempo de transferência e condições necessárias

Com o rendimento atual de 644 Mbps, transferir um arquivo de 6,5 TB leva aproximadamente 23,5 horas, o que se alinha com a degradação observada. Para obter uma janela de transferência de 9 horas, o throughput deve aumentar para aproximadamente 1,68 Gbps, exigindo uma janela TCP maior (aumento de~2,7x) ou um RTT significativamente mais baixo (~291 µs).

Com as condições atuais (janela de 64 KB e RTT de ~798 µs), não é possível atingir o objetivo de 9 horas, pois o throughput do TCP é restringido pelo produto de atraso de largura de banda. Sem aumentar o tamanho da janela ou reduzir a latência, o protocolo não pode utilizar uma largura de banda disponível mais alta, tornando o destino inatingível.

Cenário	Transferência	Tempo de transferência estimado (6,5 TB)	Janela TCP Necessária	RTT obrigatório
Estado atual	644 Mbps (~80,5 MB/s)	Aproximadamente 23,5 horas	64 KB	798 µs
Meta (9 horas)	~1683 Mbps (~210 MB/s)	9 horas	~172 KB	Aproximadamente 291 µs

Isso funcionou anteriormente, indicando que ocorreu uma alteração na rede, no aplicativo, na origem ou no destino. É importante notar que, apenas com base nesta análise inicial, já se pode chegar a uma conclusão significativa: nas condições atuais de tamanho da janela TCP e RTT, não é possível atingir o objetivo de 9 horas.

As tabelas mostram uma comparação de como o throughput varia à medida que o tamanho da janela do RTT e do TCP aumenta ou diminui.

Impacto de RTT no throughput (tamanho de janela fixo = 64.240 bytes)

RTT	Rendimento (MB/s)	Rendimento (Mbps)
200 µ (0,0002 s)	~321 MB/s	~2.568 Mbps
798 µs (0,000798 s)	~80,5 MB/s	~644 Mbps
2 ms (0,002 s)	~32,1 MB/s	~257 Mbps
10 ms (0,01 s)	~6,4 MB/s	~51 Mbps

Impacto no tamanho da janela TCP (RTT fixo = 798 µs)

Tamanho da Janela TCP	Rendimento (MB/s)	Rendimento (Mbps)
16 KB (16.384 KB)	~20,5 MB/s	~164 Mbps
64 KB (64.240 KB)	~80,5 MB/s	~644 Mbps
256 KB (262.144 KB)	~328 MB/s	~2.624 Mbps
1 MB (1.048.576 KB)	~1.314 MB/s	~10,5 Gbps

Interpretação técnica

- O throughput é inversamente proporcional ao RTT → a maior latência reduz o desempenho.
- O throughput é diretamente proporcional ao tamanho da janela TCP → janelas maiores aumentam a capacidade.
- Tamanhos de janela pequenos limitam severamente o throughput, mesmo em ambientes de baixa latência.
- Redes de alta velocidade (10G+) exigem escalonamento de janela para utilizar totalmente a largura de banda.

Isso demonstra que o tamanho da janela do RTT e do TCP são fatores críticos no desempenho do TCP e devem ser analisados juntos ao solucionar problemas de throughput.

Comprimento do Cabeçalho IP e TCP

- Comprimento do cabeçalho IP: 20 bytes
- Comprimento do cabeçalho TCP: 32 bytes

Um cabeçalho IP de 20 bytes indica que não há opções de IP presentes. O cabeçalho TCP de 32 bytes confirma que as opções TCP estão sendo usadas, adicionando 12 bytes além do cabeçalho base. Essas opções normalmente incluem MSS, Escala de janela e SACK permitido.

Análise de opções TCP e TTL

O SACK (Selective Acknowledgment) é habilitado em ambos os endpoints. Isso não está visível na imagem. O SACK permite que o receptor confirme blocos de dados não contíguos, informando ao remetente exatamente quais segmentos foram recebidos com sucesso.

Por exemplo, se os segmentos 1000-2000 e 3000-4000 forem recebidos, mas 2000-3000 estiver faltando, o receptor pode indicar isso explicitamente. Sem o SACK, o remetente retransmitiria todos os dados após o intervalo; com SACK, somente a parte ausente é retransmitida. Isso melhora significativamente o desempenho em ambientes com perda de pacotes.

Análise do pacote 1 (SYN)

- Seq. nº: 0 (Wireshark normalizado)
- Carga útil: 0 bytes
- Nº ACK: 0
- MSS: 1460 bytes
- TTL: 128

O Wireshark normaliza o número de sequência para zero para legibilidade, embora na prática seja um grande valor aleatório. A ausência de payload é esperada durante o estabelecimento da conexão. O valor de MSS de 1460 bytes indica um MTU de 1500 bytes (cabeçalho IP de 20 bytes + cabeçalho TCP de 20 bytes). Um TTL de 128 pode ser um host baseado em Windows, e ver esse valor na captura indica que a captura provavelmente foi feita na origem ou muito perto dela através da Camada 2.

Análise do pacote 2 (SYN-ACK)

- Nº ACK: 1

O valor ACK é 1 porque o flag SYN consome um número de sequência, mesmo quando não há payload presente. Portanto, ACK = SEQ + 1.

- TTL: 59

O TTL observado de 59 sugere um TTL inicial de 64, significando que o pacote atravessou

aproximadamente 5 saltos de roteamento (64 − 59 = 5). Cada salto roteado diminui o TTL em um.

Risco de fragmentação e impacto na rede

A presença de aproximadamente cinco saltos de roteamento apresenta riscos potenciais de desempenho, particularmente relacionados a incompatibilidades e fragmentação de MTU.

Se qualquer link intermediário tiver uma MTU menor do que o tamanho do pacote original, a fragmentação poderá ocorrer. Isso leva a várias consequências:

- Maior latência devido à sobrecarga de fragmentação e remontagem.
- Maior probabilidade de perda de pacotes, já que a perda de um único fragmento requer a retransmissão do pacote inteiro.
- Throughput reduzido, pois o TCP interpreta a perda como congestionamento e reduz sua taxa de envio.
- Maior utilização da CPU em dispositivos de rede que lidam com fragmentação.
- Risco de falhas de Path MTU Discovery (PMTUD) se o ICMP for bloqueado, resultando em quedas silenciosas de pacotes.

Considerando esses fatores, é essencial garantir MTU consistente no caminho ou implementar o aperto de MSS quando necessário.

Análise TCP RTT: ACK RTT vs RTT inicial

Quando o ACK RTT é maior que o iRTT, ele indica que a latência aumentou em comparação com a linha de base estabelecida durante o handshake TCP.

Isso significa que a rede ou os endpoints estão introduzindo um atraso adicional durante a sessão, geralmente devido a:

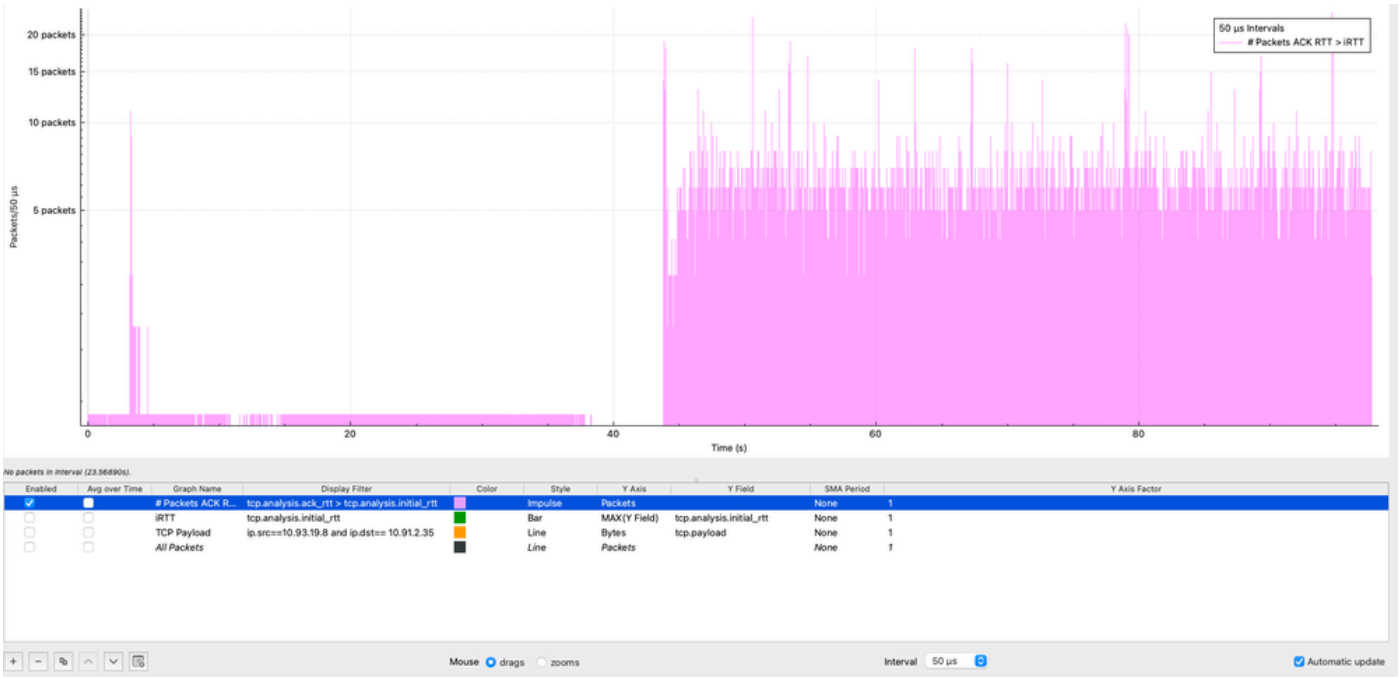
- Congestionamento ou enfileiramento da rede
- Atrasos de processamento de aplicativo ou receptor
- Dispositivos intermediários (firewalls, balanceadores de carga)
- Retransmissões

Se essa condição persistir por toda a sessão TCP, ela levará a:

- Taxa de transferência de TCP reduzida
- Utilização ineficiente da janela
- Desempenho degradado do aplicativo

No Wireshark, é possível visualizar a frequência com que a condição ACK RTT > iRTT ocorre usando o recurso Gráficos de E/S em: Estatísticas → Gráficos de E/S, aplicando o filtro de exibição (tcp.analysis.ack_rtt > tcp.analysis.initial_rtt), selecionando estilo Impulso, definindo o Eixo Y como Pacotes e usando um intervalo de 50 microssegundos.

No gráfico, os impulsos roxos representam o número de pacotes que atendem a essa condição dentro de cada intervalo de 50 microssegundos. Como observado, essa condição persiste em toda a captura de pacotes, indicando que a latência durante a sessão é consistentemente mais alta que a linha de base inicial. Esse comportamento sugere fortemente a degradação sustentada do desempenho, em vez de uma condição transitória, reforçando a necessidade de investigar fontes potenciais, como congestionamento, buffer ou atrasos de processamento de endpoint no caminho de ponta a ponta.



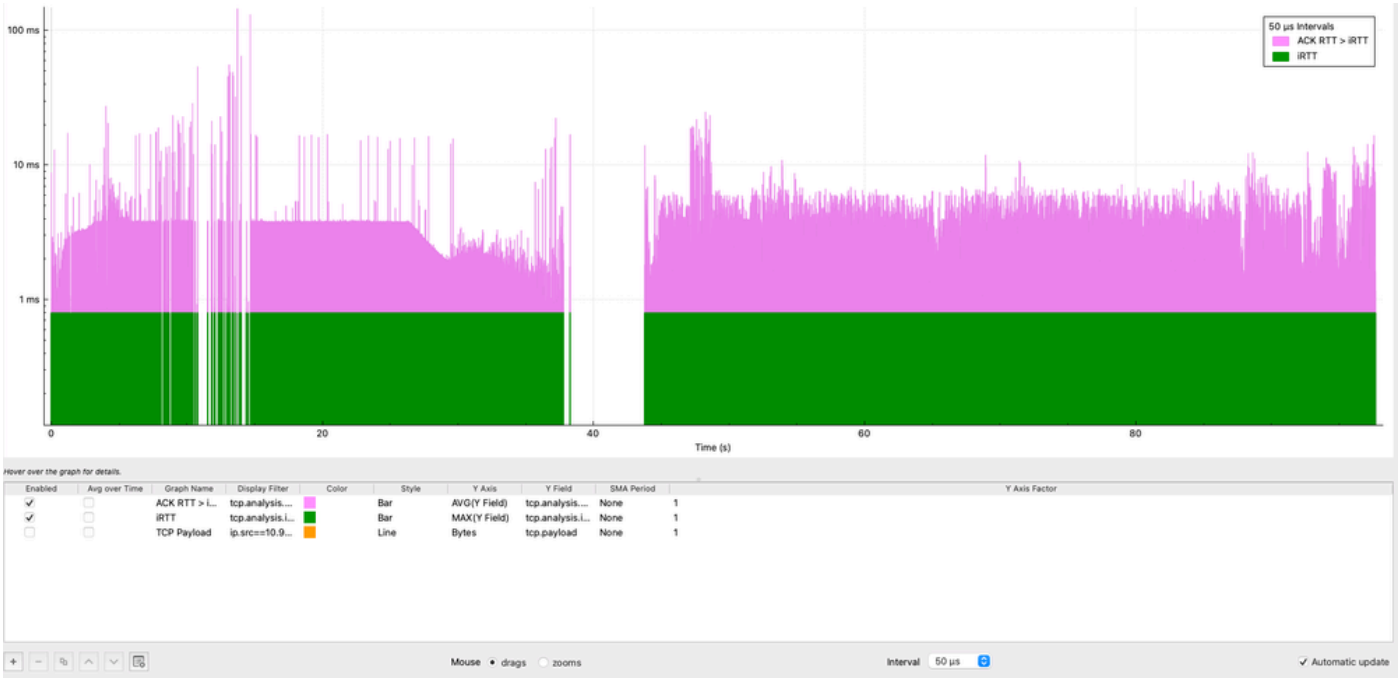
Também é importante determinar por quanto tempo o iRTT está sendo excedido, não apenas com que frequência. Embora o Wireshark não permita diretamente a subtração entre campos, uma comparação visual pode ser obtida usando Gráficos de E/S:

- Navegue até Estatísticas → Gráficos de E/S
- Gráfico 1:
  - Filtro de exibição: tcp.analysis.ack_rtt > tcp.analysis.initial_rtt
  - Estilo: Barra
  - Eixo Y: MÉDIA
  - Campo Y: tcp.analysis.ack_rtt
  - Intervalo: 50 microssegundos
- Gráfico 2:
  - Filtro de exibição: tcp.analysis.initial_rtt
  - Estilo: Barra

    - Eixo Y: MAX
    - Campo Y: tcp.analysis.initial_rtt
- Em seguida, clique com o botão direito do mouse no gráfico e ative Log scale.

Nesta visualização, o gráfico roxo representa a condição ACK RTT > iRTT, que está consistentemente presente em toda a sessão TCP. Os dados mostram inflação de latência sustentada, com vários picos atingindo 11 milissegundos e um pico máximo de mais de 100 milissegundos, representando 11x a 100x o iRTT da linha de base.

Esse comportamento confirma que o aumento de latência não é transitório, mas persistente, indicando um problema sistêmico que afeta a sessão ao longo do tempo. Tal desvio sustentado sugere fatores como congestionamento de rede, buffering (bufferbloat) ou atrasos de processamento de ponto final.



Retransmissões TCP e Análise de Retransmissões Artificiais

Esta seção avalia a confiabilidade do TCP analisando retransmissões ao longo do tempo, permitindo a validação da contribuição da perda de pacotes para a degradação do desempenho.

Retransmissões de TCP com o Tempo

O gráfico mostra a distribuição das retransmissões TCP ao longo do tempo. Foram observadas 42 retransmissões, representando apenas 0,00125% do tráfego total.

Esse nível de retransmissões é insignificante e indica claramente que a perda de pacotes não é um fator contribuinte nesse cenário.

Configuração do Wireshark (Retransmissões TCP)

```
Statistics → I/O Graphs
```

- Filtro de exibição:

```
tcp.analysis.retransmission and !tcp.analysis.spurious_retransmission
```

- Estilo: Impulso ou Barra
- Eixo Y: Pacotes
- Intervalo: 1 s

Retransmissões artificiais de TCP

O gráfico mostra o número de retransmissões artificiais TCP em intervalos de 1 s geradas pela origem 10.93.19.8.

No Wireshark, uma retransmissão espúria TCP indica que um host retransmitiu um segmento que não foi realmente perdido. O pacote original atingiu com êxito o receptor, mas o remetente assumiu incorretamente a perda devido a uma estimativa de tempo imprecisa. Esse comportamento não indica perda real de pacotes, mas sim lógica de retransmissão ineficiente no remetente.

Nesta captura:

- A origem 10.93.19.8 retransmite pacotes após apenas ~8 microssegundos.
- Enquanto os temporizadores típicos de retransmissão são da ordem de ~200 milissegundos.

Isso confirma que o comportamento de retransmissão é inteiramente controlado pela pilha TCP origem, não pela rede.

O número total de retransmissões artificiais observado é 1.112, representando 0,032% do tráfego total capturado.

Configuração do Wireshark (Retransmissões artificiais de TCP)

`Statistics → I/O Graphs`

- Filtro de exibição:

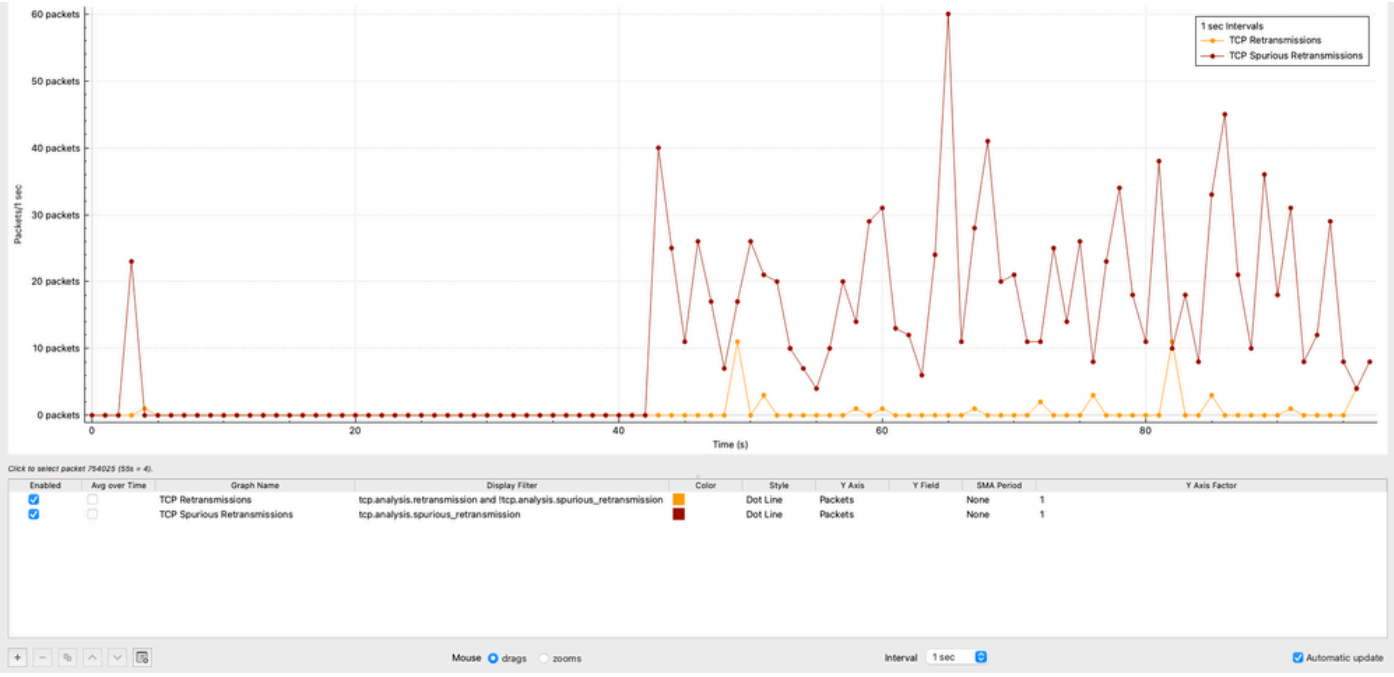
```
tcp.analysis.spurious_retransmission and ip.src==10.93.19.8
```

- Estilo: Impulso ou Barra
- Eixo Y: Pacotes
- Intervalo: 1 s

Interpretação técnica

- A porcentagem extremamente baixa de retransmissões reais confirma que a perda de pacotes não está presente na rede.
- A presença de retransmissões artificiais indica decisões prematuras de retransmissão pelo host de origem.
- Esse comportamento pode impactar um pouco a eficiência, mas não é a principal causa de uma grave degradação do throughput.

Essa análise reforça ainda mais que o problema não está relacionado à confiabilidade da rede, mas ao comportamento do TCP, latência ou desempenho do endpoint.

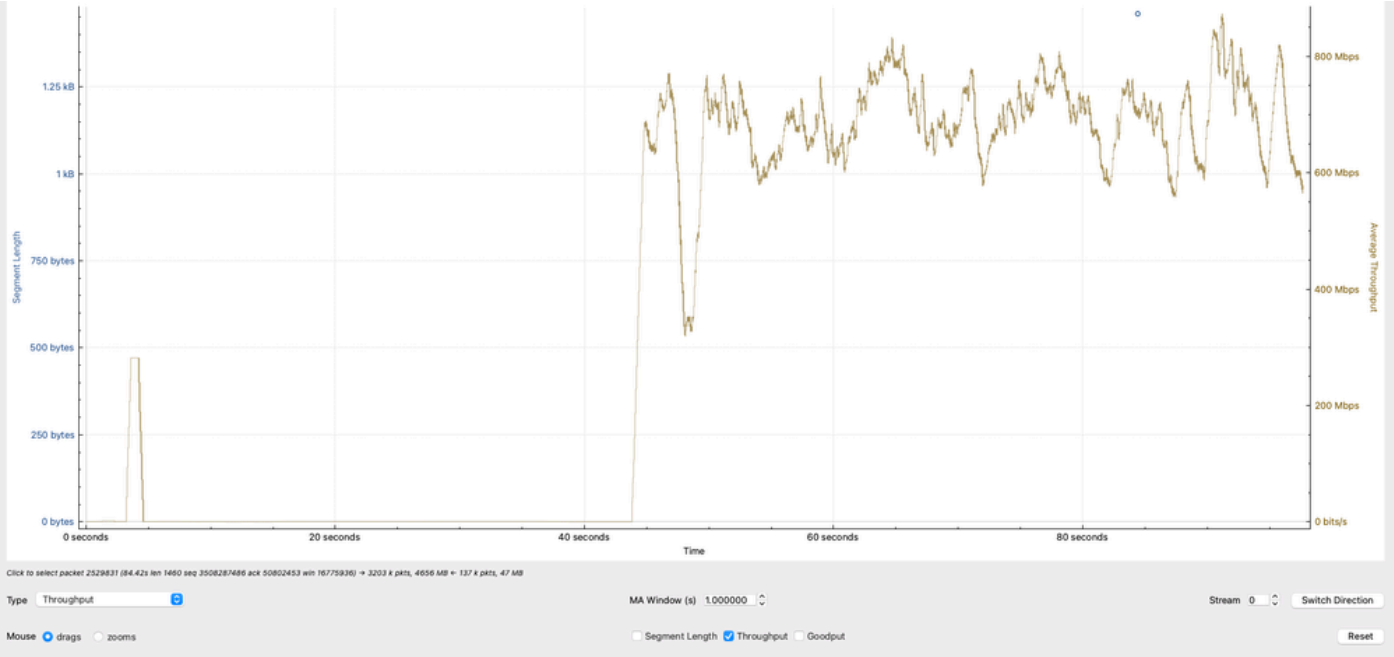


Análise de throughput efetivo

O gráfico mostra o throughput efetivo, calculado com base no payload do TCP (dados reais transferidos) em Megabits por segundo. O throughput observado oscila principalmente entre 600 Mbps e 800 Mbps, indicando que, embora a rede esteja transferindo dados ativamente, não está atingindo um potencial de largura de banda maior.

Configuração do Wireshark (Taxa de Transferência Efetiva)

`Statistics → TCP Streams Graphs → Throughout`



Interpretação técnica

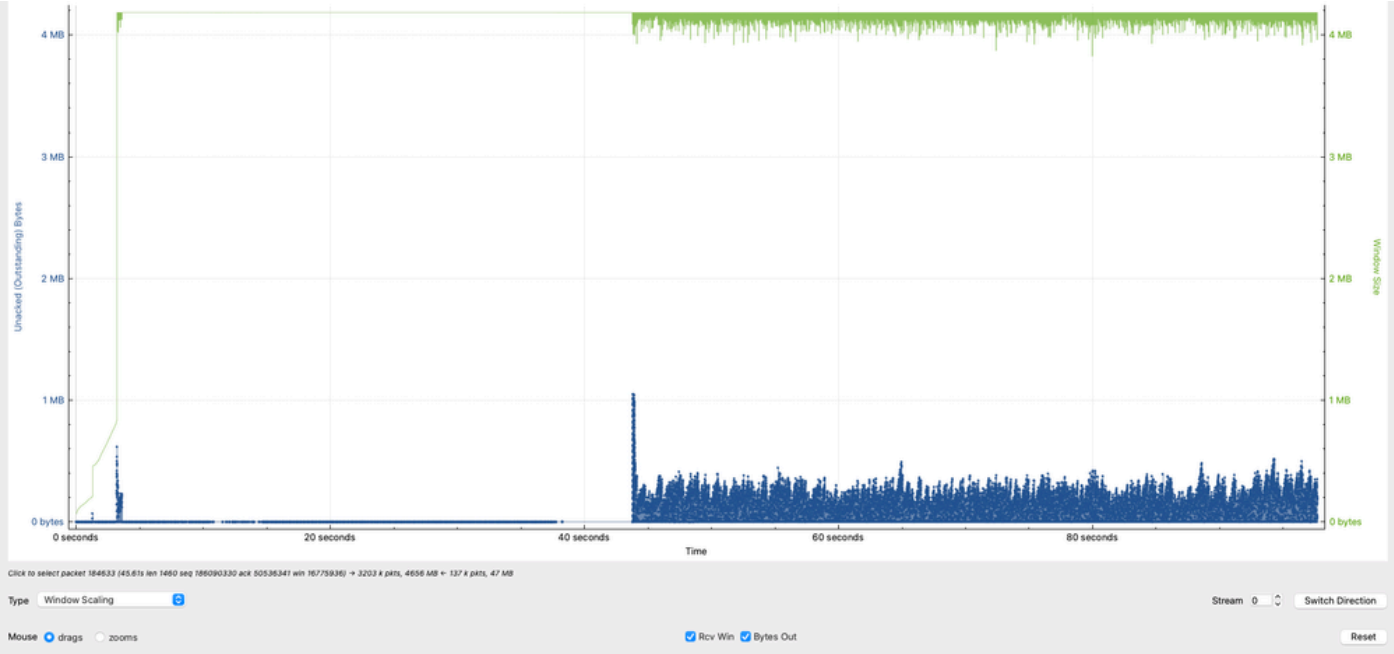
- O intervalo de throughput de 600-800 Mbps se alinha com cálculos anteriores baseados no tamanho da janela TCP e no RTT.
- A variabilidade no throughput reflete:
  - Flutuações de RTT
  - Ajustes de controle de congestionamento TCP
  - Ritmo ou colocação em buffer do aplicativo
- Como o throughput não se aproxima da taxa de linha (por exemplo, 10G), a limitação não é a largura de banda física, mas sim as restrições de eficiência do TCP.
- Essa análise confirma que o throughput observado é consistente com as limitações de TCP (tamanho da janela e latência), reforçando que o gargalo não se deve à perda de pacotes ou à capacidade da interface, mas ao comportamento da camada de transporte e às

condições do endpoint.

Análise de dados em trânsito (janela TCP)

O gráfico destaca um comportamento crítico na sessão TCP comparando a capacidade do receptor versus os dados reais em trânsito (bytes em trânsito).

- A linha verde representa a quantidade de dados TCP que 10.91.2.35 (receptor) pode aceitar (janela de recebimento efetiva).
- A linha azul representa a quantidade de dados TCP atualmente em trânsito de 10.93.19.8 (remetente).



Os dados observados em voo atingem picos de aproximadamente 1 MB, com picos adicionais em torno de 8 KB e 5 KB, mas concentram-se principalmente entre 1 KB e 250 KB.

Isso indica que, embora o receptor seja capaz de lidar com volumes maiores de dados, o remetente não está utilizando consistentemente a janela disponível.

Configuração do Wireshark (dados em voo versus janela)

```
Statistics → TCP Streams Graphs → Throughout
```

Interpretação técnica

- O receptor (10.91.2.35) anuncia uma janela significativamente maior, indicando que é capaz de receber mais dados.
- O remetente (10.93.19.8) está subutilizando a janela disponível, conforme mostrado pelos valores de dados em voo inferiores e inconsistentes.
    - O remetente pode, idealmente, manter os valores de Dados em voo mais próximos da janela anunciada dos receptores (~1 MB) para maximizar o throughput.
    - A incapacidade de sustentar altos níveis de dados em trânsito limita diretamente o throughput e é um forte indicador de ineficiência do TCP na origem, não um problema de capacidade da rede.

Payload de TCP vs MSS na Análise de Tempo

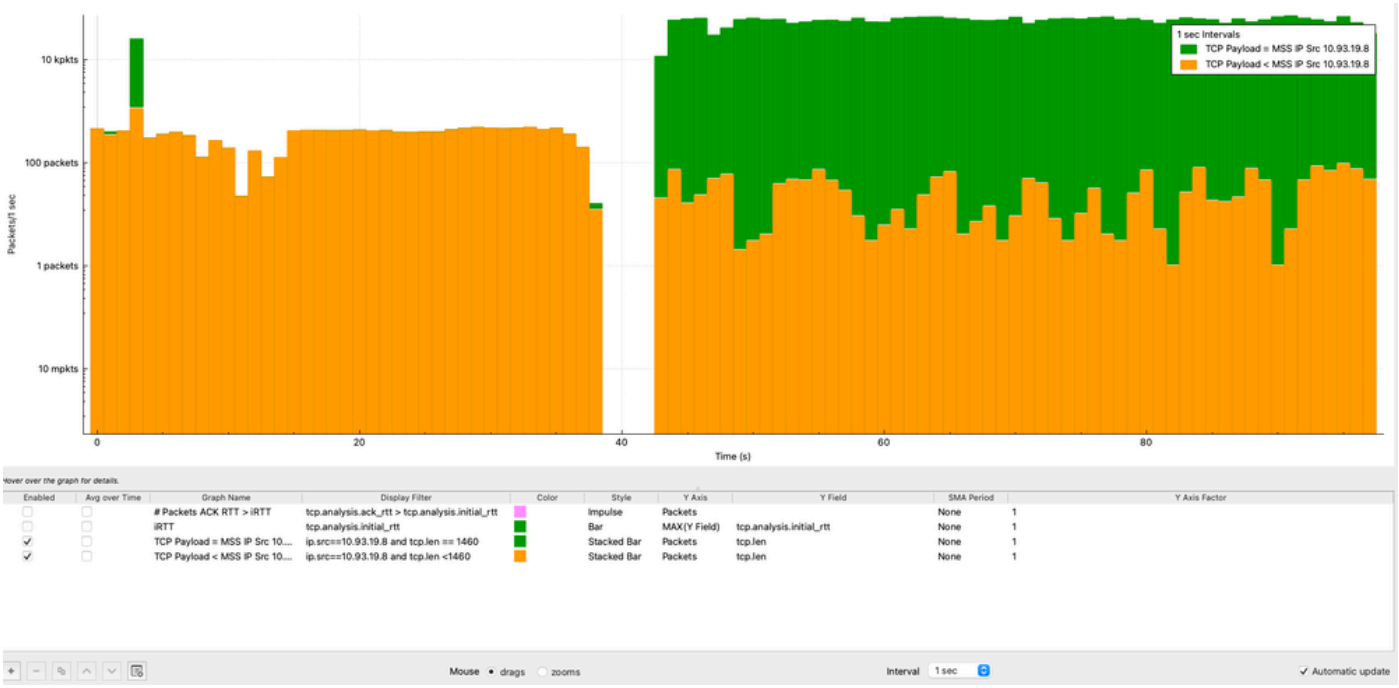
A análise do tamanho do payload TCP em relação ao MSS ao longo do tempo ajuda a determinar se o remetente está utilizando cada segmento TCP de forma eficiente. Essa análise é realizada da perspectiva do endereço IP de origem (10.93.19.8).

No Wireshark, os gráficos são configurados da seguinte forma:

- Gráfico 1 (pacotes do tamanho MSS):
    - Filtro de exibição: ip.src==10.93.19.8 e tcp.len == 1460
    - Estilo: Barra Empilhada
    - Eixo Y: Pacotes
    - Intervalo: 1 segundo
- Gráfico 2 (todos os pacotes ≤ MSS):
    - Filtro de exibição: ip.src==10.93.19.8 e tcp.len <= 1460
    - Estilo: Barra Empilhada
    - Eixo Y: Pacotes
    - Intervalo: 1 segundo
- Aplicar escala logarítmica para melhor visualização

Da análise:

- A maioria dos pacotes (>10.000 pacotes por segundo) alcança consistentemente o valor de MSS de 1460 bytes.
- Uma porção menor de pacotes carrega menos payload devido ao comportamento normal do TCP (ACKs, segmentação ou dados de fim de fluxo).



Análise de causa raiz (RCA): Degradação de Desempenho TCP

Essa análise demonstra que a identificação da causa raiz dos problemas de desempenho do TCP requer uma abordagem holística de ponta a ponta, em vez de assumir que a rede é a principal fonte de degradação.

Uma validação extensiva foi realizada no switch Cisco Nexus 9300, incluindo contadores de interface, políticas de QoS, roteamento e estabilidade ARP, verificação de punt de CPU, captura de pacotes baseada em SPAN e validação de encaminhamento de nível ASIC usando ELAM. Todos os resultados confirmaram consistentemente que o switch estava operando dentro dos parâmetros esperados:

- Nenhum descarte de pacote
- Sem latência anormal (intervalo de microssegundos)
- Sem QoS ou impacto no plano de controle
- Encaminhamento correto de hardware

Além disso, a análise do TCP revelou:

- Retransmissões insignificantes (0,00125%)
- Nenhuma evidência de perda de pacotes
- Utilização consistente do MSS na origem
- Throughput alinhado com a janela TCP e restrições RTT
- Subutilização da janela TCP disponível (análise de Dados em Voo)
- A rede não é o gargalo

- O servidor de origem está limitando o desempenho

Conclusão

A degradação de desempenho é causada pelo servidor de origem operando com MTU 1500 em um ambiente com capacidade jumbo, impedindo o uso eficiente da capacidade de rede disponível.

Solução

Aumente o MTU no servidor de origem de 1500 para 9000 bytes para alinhar com a infraestrutura de rede e de destino. Os benefícios:

- Habilitar segmentos TCP maiores
- Reduzir a sobrecarga de pacotes
- Melhorar o throughput geral

Reflexão técnica

Uma conclusão importante dessa análise é a importância de evitar conclusões prematuras ao solucionar problemas de desempenho da rede. Embora seja comum inicialmente atribuir problemas à rede, este caso demonstra claramente que a rede estava funcionando corretamente em todo o caminho do plano de dados. Somente executando uma análise profunda do TCP a partir das perspectivas de origem e destino—incluindo parâmetros de handshake, comportamento de RTT, utilização de janela, retransmissões e eficiência de payload—foi possível identificar com precisão o verdadeiro gargalo.

Dedicar um tempo para analisar o comportamento do TCP em detalhes evita diagnósticos errados, reduz alterações desnecessárias na rede e garante que os esforços de remediação sejam direcionados à causa raiz real.

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.