

TWAMP

Descriptivo técnico

Introdução

Neste documento serão apresentados: o protocolo Twamp e suas derivadas Light e Full. Descreveremos como se configura esta monitoração pela CLI, além do modo como se pode obter os resultados via SNMP. Finalmente, apresentaremos dados reais de monitoração e listaremos alguns dos equipamentos compatíveis e homologados com a solução.

Parâmetros

O comando para a criação de probes de TWAMP é:

```
probe create TWAMP name <NAME> destination <HOST> num_packets <NUMBER> light <yes|no> packet_interval <TIME> payload <BYTES> source <HOST> [port <NUMBER>] [type-p <DSCP>]
```

Os parâmetros para este comando são:

NAME – Nome da probe

HOST – IP do destino

NUMBER – Número de Pacotes

yes/no – Escolher entre o uso do TWAMP Light(yes) e o Full(no).

TIME – Intervalo (ms) entre pacotes

BYTES – Tamanho dos pacotes

HOST – IP do host (seu próprio IP)

NUMBER – Número da Porta que será usada. Caso vazio, conectará a uma porta padrão (862).

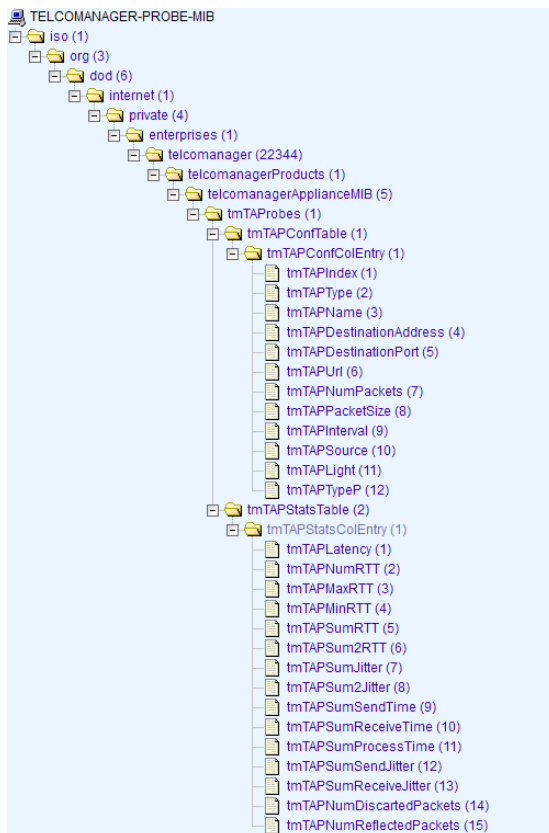
DSCP – Valor do DSCP. Caso vazio, será usado o valor padrão (000 000)

Contadores (OIDs)

OID	Nome	Descrição
1.3.6.1.4.1.22344.1.5.1.2.1.2	tmTAPNumRTT	Número de idas e voltas feitas pela probe.
1.3.6.1.4.1.22344.1.5.1.2.1.3	tmTAPMaxRTT	Valor máximo de tempos de ida e volta para a última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.4	tmTAPMinRTT	Valor mínimo de tempos de ida e volta para a última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.5	tmTAPSumRTT	Soma de todos os valores de ida e volta para a última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.6	tmTAPSum2RTT	Soma de todos os valores elevados ao quadrado de tempos de ida e volta para a última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.7	tmTAPSumJitter	Soma de todos os valores de jitter obtidos na última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.8	tmTAPSum2Jitter	Soma de todos os valores elevados ao quadrado de jitter obtidos na última probe run.

OID	Nome	Descrição
1.3.6.1.4.1.22344.1.5.1.2.1.9	tmTAPSumSendTime	Soma de todos os valores de tempo de envio da última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.10	tmTAPSumReceiveTime	Soma de todos os valores de tempo de recebimento da última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.11	tmTAPSumProcessTime	Soma de todos os valores de tempo de processo da última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.12	tmTAPSumSendJitter	Soma de todos os valores de jitter enviados obtidos da última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.13	tmTAPSumReceiveJitter	Soma de todos os valores de jitter recebidos obtidos da última probe run.
1.3.6.1.4.1.22344.1.5.1.2.1.15	tmTAPNumReflectedPackets	Número de pacotes refletidos pelo destino na última probe run.

Exemplo de Árvore



Como sumarizar KPIs

Para sumarizar as KPIs, devemos usar as OIDs mostradas anteriormente em fórmulas para calcular os valores das KPIs e, com esses valores, formar diversos gráficos.

Name	Formula
Availability	tmTAPNumRTT, 100, 0, if
Jitter	tmTAPSumJitter, tmTAPNumRTT, /, 1000, /
Jitter dst->src	tmTAPSumReceiveJitter, tmTAPNumRTT, /, 1000, /
Jitter src->dst	tmTAPSumSendJitter, tmTAPNumRTT, /, 1000, /
Latency	tmTAPSumRTT, tmTAPNumRTT, /, 1000, /
Latency dst->src	tmTAPSumReceiveTime, tmTAPNumRTT, /, 1000, /
Latency src->dst	tmTAPSumSendTime, tmTAPNumRTT, /, 1000, /
Packet loss	1, tmTAPNumRTT, tmTAPNumPackets, /, -, 100, *

New formula: Mapped object->Probe Telco TWAMP->(null)

Name

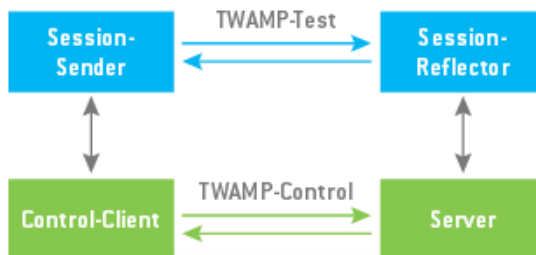
Formula

+

TWAMP

O protocolo TWAMP é um processo de monitoramento que se baseia no One-Way Active Measurement Protocol (OWAMP) com a adição da medição de performance de ida e volta e métricas two-way para redes baseadas em IP. TWAMP traz um método flexível para medição precisa de performances unidirecional entre dois endpoints que tenham suporte para TWAMP, independente do tipo de dispositivo ou vendedor.

O Control-Client inicia uma conexão TCP na porta padrão do TWAMP, e o servidor responde com sua mensagem de saudação, indicando que o(s) modo(s) de segurança/integridade está disposto a dar suporte. O Control-Client responde com o modo de comunicação escolhido e informações que suportam proteção de integridade e criptografia, se o modo exigir deles.

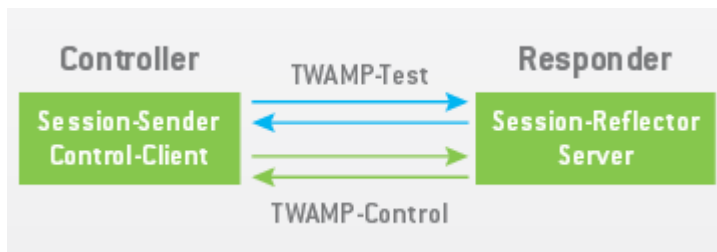


O Servidor responde para aceitar o modo e dar para ele seu tempo de início. Isso completa a configuração de controle de conexão. O Control-Client pede (e descreve) uma sessão de teste com uma mensagem única do protocolo de controle. O Servidor responde com sua aceitação e informação. Mais de uma sessão de teste pode ser requisitada com mensagens adicionais.

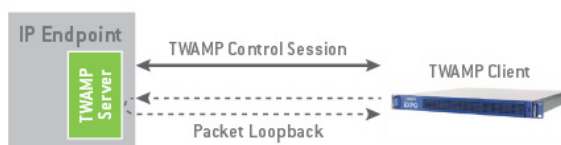
O Control-Client inicia todos os testes requisitados com a mensagem de Start-Session e o reconhecimento do Servidor. O Session-Sender e o Session-Reflector trocam pacotes de testes de acordo com o pacote de teste do protocolo TWAMP para cada sessão ativa. No momento apropriado, o Control-Client envia uma mensagem para parar todas as sessões de teste.

TWAMP Full

O teste do modo TWAMP-Full é projetado para trabalhar em uma relação Cliente-Servidor, onde o controle do teste pode ser gerenciado por dispositivos separados dos que vão enviar e receber o tráfego do teste. Existem dois protocolos ligando os dispositivos em questão: protocolo de controle e o pacote de teste do protocolo TWAMP.



Sessões de teste são inicializadas, iniciadas e paradas usando o protocolo de controle e os pacotes de testes são trocados entre os nós TWAMP por meio do pacote de teste do protocolo twamp. Como mostrado na figura abaixo, o teste é iniciado pelo controlador que faz a requisição para iniciar um teste para um servidor, que irá inicializar os processos para refletir o tráfego para o remetente. Uma vez que a sessão é aprovada, o remetente inicia o teste sob o controle do cliente. A sessão do TWAMP é estabelecida pelo Cliente via canal de controle dentro da sessão de controle. Parâmetros como as portas UDP são negociadas entre o remetente e o responder.



- Uma vez que a negociação é concluída, o cliente transmite os pacotes de teste para o Servidor.
- O Servidor faz um loopback de pacotes para o Cliente com to a cronometragem e informação de sequência embutidos.
- O Cliente computa as KPI's de IP two-way (perda, latência, jitter, duplicados, fora de ordem). Clientes sincronizados podem também computar as KPI's one-way (latency e jitter).

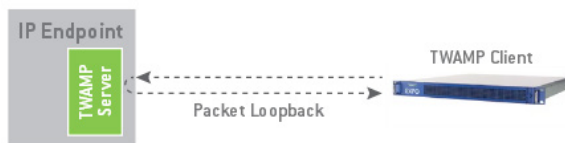
TWAMP Light

O modo TWAMP-Light é projetado para ajudar a implementar o TWAMP padrão por entre as entidades que agem como responders ativos para os controladores do TWAMP dentro da rede, assim, possibilitando a medida de performance do two-way IP de qualquer lugar dentro daquela rede.

No TWAMP-Light, o protocolo de controle é eliminado e o session-sender, control-client e o servidor se combinam em um host: o controlador. A função do session-reflector é executada por um host separado: o responder.



O responder, ou neste caso o session-reflector somente, simplesmente envia pacotes de entrada de volta para o controlador. Simultaneamente, as cópias do responder requerem informação e geram números de sequência e os valores do timestamp sem conhecimento do estado da sessão. Este é a forma mais simples de implementar o TWAMP com um session-reflector.



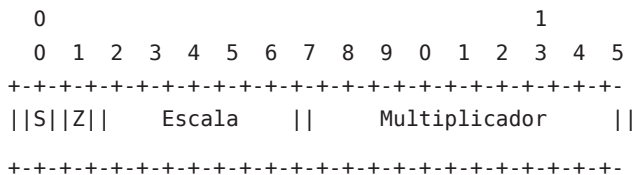
- Cliente transmite pacotes de teste para o servidor sem nenhuma negociação.
- O Servidor faz um loopback de pacotes para o Cliente com to a cronometragem e informação de sequência embutidos.
- O Cliente computa as KPI's de IP two-way (perda, latência, jitter, duplicados, fora de ordem). Clientes sincronizados podem também computar as KPI's one-way (latency e jitter).

TWAMP Exchange: TWAMP Full x Light

O TWAMP Exchange é um processo teste que é realizado entre endpoints. Este processo é similar para tanto as versões Full quanto a versão Light do TWAMP. A principal diferença é que durante a versão Full, a sessão do TWAMP é estabelecida entre o remetente e o responder através de um canal de controle dentro de uma sessão de controle que negocia a comunicação entre dispositivos. Na versão Light, não ocorrem negociações entre o endpoint e o cliente.

Determinação do Erro

A Determinação de Erro especifica a estimativa de erro e sincronização de um pacote. Ela possui o seguinte formato:

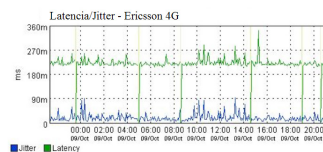
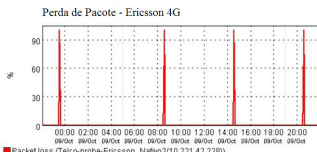
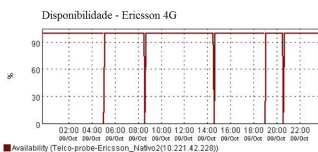


No primeiro bit, S, deve ser usado se a parte que está gerando o timestamp tem um clock que está sincronizado com o UCT usando uma fonte externa (por exemplo, o bit deve ser usado se o hardware de GPS é usado e indica que ele possui a posição e tempo atuais ou se o NTP é usado e está indicando que

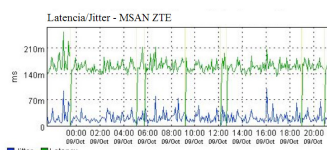
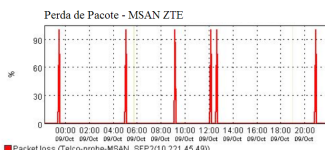
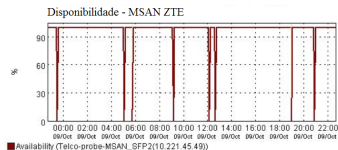
está sincronizado a uma fonte externa, na qual inclui fonte stratum 0). Se não houver nenhuma noção de sincronização externa para a origem de tempo, o bit não deve ser usado. O próximo bit tem a mesma semântica que os campos MBZ de qualquer lugar: ele deve ser usado com o valor 0 pelo remetente e ignorado por todos os outros. Os próximos seis bits, Escala, formam um inteiro não assinado; Multiplicador é um inteiro não assinado também. Eles são interpretados da seguinte forma: a estimativa de erro é igual ao $\text{Multiplicador} * 2^{-(32)} * 2^{\text{Escala}}$ (em segundos). O Multiplicador não pode ser usado com valor zero. Se o Multiplicador for zero, o pacote deve ser considerado corrompido e descartado.

Exemplos de Gráficos

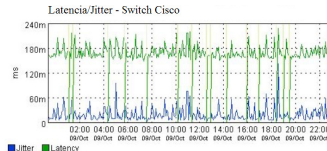
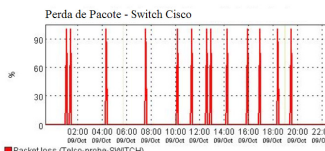
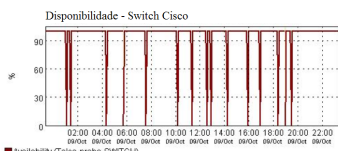
Ericsson



ZTE



Cisco



Apêndice I: Equipamentos compatíveis e homologados com a solução

Equipamentos	Fabricantes	Versão TWAMP	Modelo	Versão de Software
NodeBs	Ericsson	Light	RBS6601W	1.2.2
ENodeBs	Nokia	Light	FSME	FL16
ENodeBs	Ericsson	Light	RBS6601L	L16B
ENodeBs	Huawei	Full	BBU3900	ERAN1.1.1
MSAN	ZTE	Light	C350M	-
Routers	Cisco	Light	CSR1000v	1.5.3
Switches	Cisco	Light	6509	1.2.2



TELCO
M A N A G E R

www.telcomanager.com

+55 21 3211-2223

info@telcomanager.com