

Mejoras arquitectónicas de la escalabilidad TFTP de CUCM 11.5.x

Contenido

[Introducción](#)

[Antecedentes](#)

[Problema con el diseño actual](#)

[Tiempo de inicio del servicio](#)

[Descripción general de características](#)

[Cambios de diseño](#)

[Mejoras de rendimiento](#)

[Valores de rendimiento](#)

[Análisis de registro:](#)

[Solicitud de archivo de configuración sobre HTTP en la versión anterior a 11.5](#)

[Solicitud de archivo de configuración sobre HTTP en 11.5](#)

Introducción

Este documento trata sobre la función de arquitectura de escala del protocolo trivial de transferencia de archivos (TFTP) implementada como parte de Cisco Unified Communication Manager (CUCM) versión 11.5, la versión más reciente de CUCM. Esta es puramente una función de ingeniería para mejorar el servicio TFTP con respecto al uso de memoria y cómo sirve a la configuración y a los archivos estáticos. La lógica empresarial sigue siendo la misma y no hay impacto con respecto a otros servicios proporcionados por TFTP.

Antecedentes

Razones por las que esta mejora fue necesaria e incorporada

Problema con el diseño actual

- La lógica de cómo TFTP atiende los archivos de configuración no ha cambiado desde hace mucho tiempo.
- Antes de 11.5, el servicio TFTP construye los archivos de configuración y almacena en memoria caché todos los archivos de configuración.
- Con más capacidad añadida a CUCM con respecto al número de teléfonos soportados, la huella de memoria del servicio TFTP aumentó linealmente.
- Las hojas de ruta futuras requieren capacidad adicional para los teléfonos con el fin de implementarlas en CUCM.
- Por lo tanto, abordar el aumento de la huella de memoria del servicio TFTP se vuelve importante.

Tiempo de inicio del servicio

- En implementaciones medianas a grandes con teléfonos de 20 000 a 40 000 configurados.
- Cuando se realiza un cambio que afecta a todos los teléfonos, el TFTP genera todos los archivos de configuración afectados y reconstruye la memoria caché.
- Esto aumenta el tiempo que tarda el servicio TFTP en iniciarse.
- En el momento en que los teléfonos solicitan el archivo de configuración, se envía una respuesta de ocupado al teléfono.

Descripción general de características

La nueva función implementada aborda los dos problemas anteriores mediante un diseño sin caché y genera el archivo de configuración a demanda. Cuando se envía una solicitud desde el teléfono, el servicio TFTP genera el archivo de configuración sobre la marcha y lo envía al teléfono en tiempo real. No guardará en memoria caché el archivo de configuración, lo que a su vez reduce el tiempo de inicio del servicio y el espacio de memoria del servicio TFTP.

Cambios de diseño

Los cambios realizados en el diseño se clasifican en dos categorías, a saber, "Gestión de conexiones" y "Generación de archivos de configuración". En la tabla siguiente se detallan los cambios realizados en cada categoría.

Gestión de conexiones		Generación de archivos de configuración
HTTP	TFTP	
La capa de servicio de red está diseñada para utilizar SDL para gestionar todas las conexiones TCP	No hay cambios cuando los teléfonos solicitan los archivos de configuración a través de UDP	Marco agregado para la generación a demanda de archivos de configuración firmados

Mejoras de rendimiento

A continuación se muestran las mejoras de rendimiento logradas con la implementación de esta nueva función.

- Reducción significativa del espacio de memoria del servicio TFTP
- La huella de memoria es de aproximadamente 600 MB para el servicio TFTP
- La hora de inicio del servicio es menor, ya que los archivos no están almacenados en caché
- La hora de inicio del servicio es independiente del número de teléfonos implementados en el sistema

Valores de rendimiento

	Nº de teléfonos	Tiempo de espera en la versión anterior a 11.5	Tiempo empleado en la versión 11.5
Hora de inicio del servicio	20000	3 minutos 38 segundos	0 minutos 19 segundos
Archivos almacenados en HTTP	20000	7 minutos 24 segundos	4 minutos 06 segundos

Archivos servidos a través
de TFTP

20000

5 minutos 36 segundos

4 minutos 11 segundos

Nota: Los números anteriores no son sólo de una ejecución de prueba, sino que son una media de varias ejecuciones de prueba.

Análisis de registro:

Dispositivos utilizados:

CUCM, versión 11.5.1.10000-6

Cisco IP Communicator versión 8.6.2

Solicitud de archivo de configuración sobre HTTP en la versión anterior a 11.5

Solicitud desde el teléfono para el archivo de configuración

```
00593088.000 |21:58:11.698 |AppInfo | TID[da900b70] HTTPEngine::getRequest(),  
[0xa0d6c90~7~10.65.64.132~54462] INFO:: socket(12), ReqTimeout[60],  
Request[GET /SEP000C29ED3D88.cnf.xml HTTP/1.1
```

Dado que todos los archivos se almacenan en la memoria caché después de su compilación, TFTP encuentra el archivo de configuración en la memoria caché

```
00593097.000 |21:58:11.698 |AppInfo  
|CReqContext::FindAndServe(1)[0xa0d6c90~7~10.65.64.132~54462]  
,[ (SEP000C29ED3D88.cnf.xml), (6779), (0xf388c2a8)] found in config cache
```

El archivo de configuración se envía correctamente al teléfono

```
00593102.000 |21:58:11.698 |AppInfo |  
HTTPEngine::sendResponse[0xa0d6c90~7~10.65.64.132~54462]  
FileName[SEP000C29ED3D88.cnf.xml], Version[HTTP/1.1], Size[6779] 00593103.000 |21:58:11.698  
|AppInfo | HTTPEngine::sendResponse[0xa0d6c90~7~10.65.64.132~54462]  
INFO:: [85][HTTP/1.1 200 OK
```

Solicitud de archivo de configuración sobre HTTP en 11.5

Solicitud desde el teléfono para el archivo de configuración

```
00000510.003 |21:47:40.683 |AppInfo | HTTPConnection::wait_SdlDataInd Printing the  
HTTPRequest :  
msgBuffer size [148] --: GET /SEP000C29ED3D88.cnf.xml HTTP/1.1
```

El proceso ServeFile envía la señal 'FileRequest' a ServeDynamicFile

```
00000511.010 |21:47:40.683 |AppInfo | ServeFile::wait_FileRequest Sending the  
FileRequest signal to ProcessServeDynamicFile process
```

```
00000511.011 |21:47:40.683 |AppInfo |<--ServeFile::wait_FileRequest
```

```
00000512.000 |21:47:40.683 |SdlSig | FileRequest |wait
|ServeDynamicFile(1,600,25,1) |ServeFile(1,600,24,1) |1,600,14,4.3^*^*
|*TraceFlagOverrode
```

Dado que se implementa el diseño de caché, verá que el TFTP genera el archivo de configuración

```
00000512.027 |21:47:40.684 |AppInfo |TFTPList::GetSupportsFMT(), Pkid[9e9cb809-df9f-4bce-8a41-
37cd5f7e4d21] Name[SEP000C29ED3D88] Class[1] Product[30041] Model[30016] Protocol[0],
DevProfile[0] SUPPORTs[2], Value[2]
```

```
00000512.028 |21:47:40.684 |AppInfo |<--TFTPList::SelectByDeviceID[0,0]
```

```
00000512.029 |21:47:40.684 |AppInfo | ServeDynamicFile::wait_FileRequest
Build Config file for Device [SEP000C29ED3D88]
```

El proceso ServeDynamicFile envía la señal 'FileResponse' a ServeFile

```
00000512.091 |21:47:40.686 |AppInfo |<--ServeDynamicFile::wait_FileRequest
00000513.000 |21:47:40.686 |SdlSig | FileResponse |wait
|ServeFile(1,600,24,1) |ServeDynamicFile(1,600,25,1) |1,600,14,4.3^*^*
|*TraceFlagOverrode
```

```
00000513.002 |21:47:40.686 |AppInfo | ServeFile::wait_FileResponse File
Response signal received by ServeFile process
```

El archivo solicitado se envía al teléfono

```
00000514.001 |21:47:40.686 |AppInfo |-->HTTPConnection::wait_FileResponse
00000514.002 |21:47:40.686 |AppInfo | HTTPConnection::wait_FileResponse Requested
file FOUND... Sending file Response
00000514.003 |21:47:40.686 |AppInfo |<--HTTPConnection::wait_FileResponse
```