

CUCM 11.5.x: Verbesserungen bei der TFTP-Skalierung und der Architektur

Inhalt

[Einführung](#)

[Hintergrundinformationen](#)

[Probleme mit dem aktuellen Design](#)

[Startzeit des Service](#)

[Übersicht der Funktionen](#)

[Design-Änderungen](#)

[Leistungsverbesserungen](#)

[Leistungskennzahlen](#)

[Protokollanalyse:](#)

[Anforderung von Konfigurationsdateien über HTTP in einer Version vor 11.5](#)

[Konfigurationsdateianforderung über HTTP in 11.5](#)

Einführung

Dieses Dokument enthält Informationen zur Architektur der Trivial File Transfer Protocol (TFTP)-Skalierung, die als Teil von Cisco Unified Communication Manager (CUCM) Version 11.5 als neueste Erweiterung für CUCM implementiert wurde. Dies ist eine rein technische Funktion, um den TFTP-Dienst hinsichtlich der Speichernutzung und der Art und Weise, wie er die Konfigurations- und statischen Dateien bedient, zu verbessern. Die Geschäftslogik bleibt unverändert, und es ergeben sich keine Auswirkungen auf andere von TFTP bereitgestellte Services.

Hintergrundinformationen

Gründe, weshalb diese Verbesserung erforderlich und integriert war

Probleme mit dem aktuellen Design

- Die Logik, wie TFTP die Konfigurationsdateien bereitstellt, wurde lange Zeit nicht geändert.
- Vor 11.5 erstellt der TFTP-Dienst die Konfigurationsdateien und speichert alle Konfigurationsdateien im Speicher.
- Da der CUCM mehr Kapazität in Bezug auf die Anzahl der unterstützten Telefone hinzugefügt wurde, stieg der Platzbedarf für den TFTP-Service linear an.
- Zukünftige Roadmaps erfordern zusätzliche Kapazitäten für Telefone, um in CUCM implementiert zu werden.
- Daher ist es wichtig, die Erhöhung der Speicherkapazität des TFTP-Service zu bewältigen.

Startzeit des Service

- In mittelgroßen bis großen Bereitstellungen mit konfigurierten Telefonen von 20.000 bis 40.000
- Wenn eine Änderung vorgenommen wird, die alle Telefone betrifft, erstellt TFTP alle betroffenen Konfigurationsdateien und erstellt den Cache neu.
- Dadurch wird die Zeit verlängert, die zum Starten des TFTP-Services benötigt wird.
- Wenn Telefone Konfigurationsdateien anfordern, wird eine Besetzzeichen-Antwort an das Telefon gesendet.

Übersicht der Funktionen

Die implementierte neue Funktion behebt die beiden oben genannten Probleme durch ein Cache-loses Design und die On-Demand-Erstellung der Konfigurationsdatei. Wenn eine Anfrage vom Telefon gesendet wird, erstellt der TFTP-Dienst die Konfigurationsdatei sofort und stellt sie in Echtzeit für das Telefon bereit. Die Konfigurationsdatei wird nicht im Arbeitsspeicher zwischengespeichert, wodurch wiederum die Servicestartzeit und der Arbeitsspeicherbedarf des TFTP-Services reduziert werden.

Design-Änderungen

Die vorgenommenen Designänderungen fallen in zwei Kategorien: "Connection Management" (Verbindungsverwaltung) und "Configuration File Generation" (Konfigurationsdateigenerierung). In der Tabelle unten sind die in den einzelnen Kategorien vorgenommenen Änderungen aufgeführt.

Verbindungsmanagement		Erstellen von Konfigurationsdateien
HTTP	TFTP	Framework hinzugefügt für Demand-Build- und signierte Konfigurationsdateien
Network Service Layer ist für die Verwendung von SDL zur Verarbeitung aller TCP-Verbindungen konzipiert.	Keine Änderungen, wenn Telefone Konfigurationsdateien über UDP anfordern	

Leistungsverbesserungen

Im Folgenden finden Sie die Leistungsverbesserungen, die mit der Implementierung dieser neuen Funktion erzielt wurden.

- Deutliche Reduzierung des Arbeitsspeicherbedarfs des TFTP-Service
- Die Speicherkapazität des TFTP-Service beträgt ca. 600 MB.
- Die Startzeit für den Dienst ist geringer, da die Dateien nicht zwischengespeichert werden.
- Die Startzeit des Service ist von der Anzahl der im System bereitgestellten Telefone unabhängig.

Leistungskennzahlen

	Anzahl der Telefone	Zeit bis zur Version vor 11.5	Zeitaufwand in Version 11.5
Service-Startzeit	20000	3 Minuten, 38 Sekunden	0 Minuten 19 Sekunden
Über HTTP bereitgestellte	20000	7 Minuten, 24 Sekunden	4 Minuten, 06 Sekunden

Dateien Über TFTP bereitgestellte Dateien

20000

5 Minuten, 36 Sekunden

4 Minuten, 11 Sekunden

Hinweis: Die obigen Zahlen stammen nicht nur aus einem Testlauf, sondern sind im Durchschnitt aus mehreren Testläufen.

Protokollanalyse:

Verwendete Geräte:

CUCM-Version 11.5.1.1000-6

Cisco IP Communicator Version 8.6.2

Anforderung von Konfigurationsdateien über HTTP in einer Version vor 11.5

Anforderung der Konfigurationsdatei vom Telefon

```
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
```

Da alle Dateien nach dem Erstellen zwischengespeichert werden, sucht TFTP die zwischengespeicherte Konfigurationsdatei

```
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
```

Die Konfigurationsdatei wird dem Telefon erfolgreich bereitgestellt.

```
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
```

Konfigurationsdateianforderung über HTTP in 11.5

Anforderung der Konfigurationsdatei vom Telefon

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

ServeFile-Prozess sendet das Signal 'FileRequest' an 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
```

Da das cachless-Design implementiert ist, sehen Sie, dass TFTP die Konfigurationsdatei erstellt.

```
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]
```

ServerDynamicFile-Prozess sendet das Signal 'FileResponse' an 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
```

Die angeforderte Datei wird an das Telefon gesendet.

```
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
```