

# Miglioramenti dell'architettura della scala CUCM 11.5.x TFTP

## Sommario

[Introduzione](#)

[Premesse](#)

[Problema con la progettazione corrente](#)

[Ora di avvio del servizio](#)

[Panoramica delle funzionalità](#)

[Modifiche alla struttura](#)

[Miglioramenti delle prestazioni](#)

[Prestazioni](#)

[Analisi registro:](#)

[Richiesta del file di configurazione su HTTP in una versione precedente alla 11.5](#)

[Richiesta file di configurazione su HTTP in 11.5](#)

## Introduzione

Questo documento descrive l'architettura di scala TFTP (Trivial File Transfer Protocol) implementata come parte di Cisco Unified Communications Manager (CUCM) versione 11.5, l'aggiornamento più recente a CUCM. Si tratta di una funzionalità puramente tecnica per migliorare il servizio TFTP rispetto all'utilizzo della memoria e al modo in cui serve i file statici e di configurazione. La logica commerciale rimane la stessa e non vi è alcun impatto sugli altri servizi forniti dal TFTP.

## Premesse

Motivi per cui questo miglioramento è stato necessario e integrato

### Problema con la progettazione corrente

- La logica del modo in cui il protocollo TFTP serve i file di configurazione non è stata modificata per molto tempo.
- Nella versione precedente alla 11.5, il servizio TFTP crea i file di configurazione e memorizza nella cache tutti i file di configurazione.
- Con una maggiore capacità aggiunta a CUCM rispetto al numero di telefoni supportati, l'ingombro di memoria del servizio TFTP è aumentato in modo lineare.
- Le future roadmap richiedono una capacità aggiuntiva per i telefoni per essere implementate in CUCM.
- Pertanto, affrontare l'aumento di impronta di memoria del servizio TFTP diventa importante.

### Ora di avvio del servizio

- In installazioni di medie e grandi dimensioni con telefoni configurati da 20.000 a 40.000.
- Quando viene apportata una modifica che influisce su tutti i telefoni, il TFTP crea tutti i file di configurazione interessati e ricostruisce la cache.
- Ciò aumenta il tempo necessario per l'avvio del servizio TFTP.
- Al momento in cui i telefoni richiedono il file di configurazione, al telefono viene inviata una risposta occupata.

## Panoramica delle funzionalità

La nuova funzionalità implementata risolve i due problemi descritti in precedenza mediante una progettazione senza cache e crea il file di configurazione su richiesta. Quando una richiesta viene inviata dal telefono, il servizio TFTP crea il file di configurazione al volo e lo serve al telefono in tempo reale. Non memorizza nella cache il file di configurazione, il che a sua volta riduce il tempo di avvio del servizio e l'ingombro di memoria del servizio TFTP.

## Modifiche alla struttura

Le modifiche progettuali apportate rientrano in due categorie, ovvero "Gestione connessione" e "Generazione file di configurazione". Nella tabella seguente sono riportate le modifiche apportate in ciascuna categoria.

	Gestione connessione	Generazione file di configurazione
	HTTP	TFTP
	Network Service Layer è progettato per utilizzare SDL per gestire tutte le connessioni TCP	Nessuna modifica in cui i telefoni richiedono i file di configurazione su UDP
		Framework aggiunto per la compilazione su richiesta e i file di configurazione firmati

## Miglioramenti delle prestazioni

Di seguito sono riportati i miglioramenti delle prestazioni ottenuti con l'implementazione di questa nuova funzione.

- Riduzione significativa dell'ingombro di memoria del servizio TFTP
- L'ingombro della memoria è di circa 600 MB per il servizio TFTP
- L'ora di avvio del servizio è inferiore poiché i file non sono memorizzati nella cache
- L'ora di avvio del servizio è indipendente dal numero di telefoni installati nel sistema

## Prestazioni

	N. di telefoni	Tempo impiegato nella versione precedente alla 11.5	Tempo impiegato nella versione 11.5
Ora di inizio del servizio	20000	3 minuti 38 secondi	0 Minuti 19 Secondi
File forniti tramite HTTP	20000	7 minuti 24 secondi	4 Minuti 06 Secondi
File serviti tramite TFTP	20000	5 minuti 36 secondi	4 minuti 11 secondi

**Nota:** I numeri indicati sopra non si riferiscono solo a una esecuzione dei test, ma rappresentano una media di più esecuzioni dei test.

## Analisi registro:

### Dispositivi usati :

CUCM versione 11.5.1.10000-6

Cisco IP Communicator versione 8.6.2

## Richiesta del file di configurazione su HTTP in una versione precedente alla 11.5

### Richiesta telefonica per il file di configurazione

```
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
```

Poiché tutti i file vengono memorizzati nella cache dopo la generazione, il protocollo TFTP trova il file di configurazione memorizzato nella cache

```
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
```

Il file di configurazione è stato inviato al telefono

```
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
```

## Richiesta file di configurazione su HTTP in 11.5

### Richiesta telefonica per il file di configurazione

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

Il processo ServeFile invia il segnale '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
```

Poiché la progettazione senza cache è implementata, è possibile notare che il TFTP crea il file di

## configurazione

```
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]
```

Il processo ServeDynamicFile invia il segnale '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
```

Il file richiesto viene inviato al telefono

```
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
```