

Esempio di configurazione del tunneling dei protocolli asincroni in BSTUN

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Premesse](#)

[Configurazione](#)

[Esempio di rete](#)

[Configurazioni](#)

[Verifica](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

[Introduzione](#)

I protocolli asincroni nativi e dedicati non sono supportati direttamente da alcuna implementazione Cisco. Tuttavia, il tunneling asincrono-generico BSTUN (Block Serial Tunnel) può fornire capacità limitate di tunneling di questi dati.

[Prerequisiti](#)

[Requisiti](#)

Nessun requisito specifico previsto per questo documento.

[Componenti usati](#)

Le informazioni fornite in questo documento si basano sulle versioni software e hardware:

- Utilizzare [Feature Navigator II](#) (solo utenti [registrati](#)) e l'opzione **Cerca per feature (Search by Feature)**.
- Utilizzare [Software Advisor](#) (solo utenti [registrati](#)) per cercare la versione software minima supportata necessaria per l'hardware.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento [Cisco sulle convenzioni nei suggerimenti tecnici](#).

Premesse

I protocolli asincroni come il TC500 di Diebold per comunicare con gli sportelli ATM o il tunneling di HyperTerminal da un PC a un altro non dispongono di supporto diretto o implementazione in Cisco IOS ®. Come indica il nome, si tratta di un'implementazione generica che dispone di una certa capacità per trasmettere questo tipo di dati. Questo processo è noto come BSTUN async-generic e richiede il set di funzionalità IBM o Enterprise IOS.

BSTUN async-generic è stato originariamente progettato per trasportare piccoli pacchetti unidirezionali da dispositivi di sicurezza a un dispositivo di reporting. BSTUN asincrono-generico, tuttavia, può trasportare traffico interattivo. In sostanza, questa implementazione si collega a dispositivi asincroni nativi e riceve i dati nell'interfaccia seriale e quindi in un buffer di memoria. Periodicamente, i dati memorizzati nel buffer vengono quindi incapsulati in un pacchetto TCP e inviati al peer BSTUN, dove vengono decapsulati e inviati al dispositivo asincrono collegato al sito remoto.

BSTUN asincrono-generico è un'operazione semplicistica. Il router non può essere configurato in modo da conoscere l'inizio del frame (SOF), la fine del frame (EOF) o lo schema di indirizzamento del protocollo asincrono. Se la parte dell'indirizzo del frame è in ogni frame, è lunga un byte e si trova nella stessa posizione nel frame, il comando **asp address-offset** può essere emesso per specificare al router dove trovare l'indirizzo nel frame, come mostrato più avanti in questo documento. In molte situazioni, tuttavia, nel protocollo non sarà contenuta una parte dell'indirizzo. Non essendo a conoscenza della costruzione del protocollo asincrono, il router non è in grado di distinguere i singoli pacchetti dagli altri se non sono separati da un intervallo di tempo. Sono necessari circa 40 ms tra i frame a 9600 bit/sec per fornire al router il tempo necessario per distinguere correttamente un pacchetto da un altro. Il router rileva semplicemente un flusso di dati nell'interfaccia seriale e quindi incapsula i dati in TCP. Non è possibile che il router sia in grado di prendere decisioni di routing in base a un singolo aspetto del frame in arrivo. Di conseguenza, i generici asincroni BSTUN devono essere progettati fisicamente in modo che solo un dispositivo si colleghi all'interfaccia seriale del router. Nessuna funzionalità di riconoscimento locale. BSTUN supporta Local-Pack solo per il protocollo IBM3270 BISYNC.

Configurazione

In questa sezione vengono presentate le informazioni necessarie per configurare le funzionalità descritte più avanti nel documento.

Esempio di rete

Nel documento viene usata l'impostazione di rete mostrata nel diagramma.



Entrambi i PC utilizzano l'HyperTerminal di Microsoft oppure uno dei due potrebbe essere una connessione alla porta console di un router Cisco. Questi esempi di configurazione rappresentano configurazioni implementate da router non configurati in precedenza in uno scenario lab e mostrano le parti pertinenti della configurazione richiesta. Queste sono configurate presupponendo una connessione a 9600 bit/sec, 8N1.

Configurazioni

Nel documento vengono usate le configurazioni mostrate in questa sezione.

- Router principale (Cisco 1700 Router)
- Router remoto (Cisco 3640 Router)
- Router principale (Cisco 3600 Router)
- Remoto 1 (Cisco 1700 Router)
- Remoto n. 2 (Cisco 1700 Router)

Router principale (Cisco 1700 Router)

```
main#show running-config
Building configuration...
.
.
.
ip subnet-zero
bstun peer-name 10.1.1.1
bstun protocol-group 1 async-generic
interface loopback0
  ip address 10.1.1.1 255.0.0.0
interface serial0
  physical-layer async
  encapsulation bstun
  asp role secondary
  bstun group 1
  bstun route all tcp 30.1.1.1
interface serial1
  ip address 20.1.1.1 255.0.0.0
ip route 0.0.0.0 0.0.0.0 20.1.1.2
line 1
  speed 9600
  databits 8
  parity none
  stopbits 1
.
.
.
!
```

Router remoto (Cisco 3640 Router)

```

REMOTE#show running-config
Building configuration...
bstun peer-name 30.1.1.1.
bstun protocol-group 1 async-generic
interface loopback 0
    ip address 30.1.1.1
interface ethernet1/0
    shutdown
interface serial 2/0
    physical-layer async
    encapsulation bstun
    asp role primary
    bstun group 1
    bstun route all tcp 10.1.1.1

interface serial 2/1
    ip address 20.1.1.2 255.0.0.0
ip route 0.0.0.0 0.0.0.0 20.1.1.1
line 65
    speed 9600
    parity none
    databits 8
    stopbits 1
.
.
!
end

```

Nota: quando si esegue il comando **physical-layer async** sull'interfaccia seriale, all'interfaccia seriale viene assegnata una riga TTY. In questa definizione di linea TTY vengono configurati i database, i bit di stop, la parità e la velocità. Questa è la formula per determinare quale linea corrisponde a quale interfaccia seriale.

$linea\# = (slot\# \times 32) + interfaccia\# + 1$

La riga show nell'output di configurazione del router remoto indica nella colonna all'estrema destra il numero di riga corrispondente. Serial2/0 è rappresentato dalla riga 65 e le definizioni fisiche per questo collegamento sono configurate nella riga 65

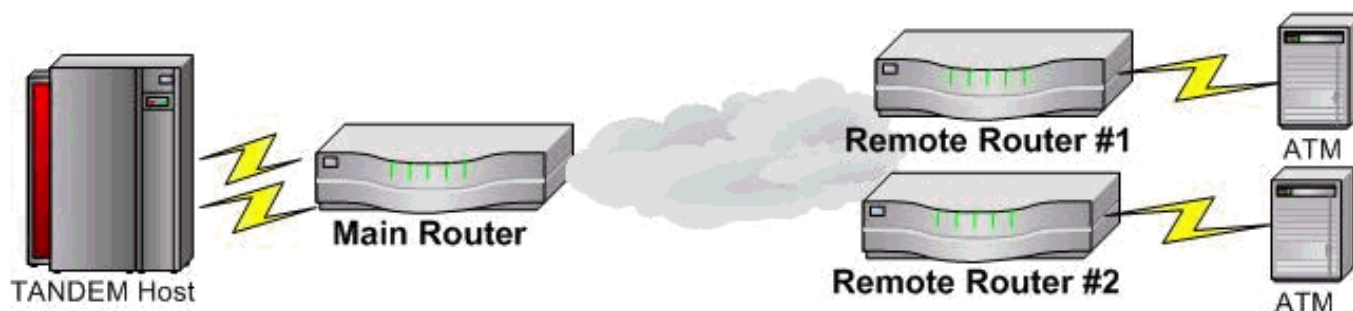
```

REMOTE#sh line
  Tty Typ      Tx/Rx      A Modem  Roty AccO AccI   Uses   Noise  Overruns  Int
*   0 CTY          -          -        -        -        -        -        -        0        0
0/0
   65 TTY    9600/9600  -          -        -        -        -        -        0        0        0/0
Se2/0
  129 AUX  9600/9600  -          -        -        -        -        -        0        0        0/0
-
  130 VTY          -          -        -        -        -        -        -        0        0
0/0
  131 VTY          -          -        -        -        -        -        -        0        0
0/0
  132 VTY          -          -        -        -        -        -        -        0        0
0/0
  133 VTY          -          -        -        -        -        -        -        0        0
0/0
  134 VTY          -          -        -        -        -        -        -        0        0
0/0

```

Line(s) not in async mode -or- with no hardware support:
1-64, 66-128

In questo scenario, un tandem comunica con dispositivi ATM remoti. In questa configurazione di esempio, il protocollo asincrono esegue un protocollo 4800 7E2 e il router principale collegato al TANDEM è un router serie 3600 per router remoti serie 1700. Vedere questo diagramma di rete.



Router principale (Cisco 3600 Router)

```
main#show running-config
Building configuration...
bstun peer-name 10.1.1.1.
bstun protocol-group 1 async-generic
bstun protocol-group 2 async-generic
interface loopback 0
  ip address 10.1.1.1
interface serial1/0
  encapsulation frame-relay
interface serial 1/0.1 point-to-point
  ip address 20.1.1.1 255.255.255.0
  frame-relay interface-dlci 100
interface serial 1/0.2 point-to-point
  ip address 20.2.1.1 255.255.255.0
  frame-relay interface-dlci 200
interface serial 2/0
  physical-layer async
  encapsulation bstun
  asp role secondary
  bstun group 1
  bstun route all tcp 30.1.1.1

interface serial 2/1
  physical-layer async
  encapsulation bstun
  asp role secondary
  bstun group 2
  bstun route all tcp 30.2.1.1

ip route 30.2.1.0 255.255.0.0 20.2.1.2
ip route 0.0.0.0 0.0.0.0 20.1.1.2
line 65
  speed 4800
  parity even
  databits 7
  stopbits 1
.
line 66
  speed 4800
```

```
parity even
databits 7
stopbits 1
.
!
end
```

Remoto 1 (Cisco 1700 Router)

```
REMOTEL#show running-config
Building configuration...
bstun peer-name 30.1.1.1
bstun protocol-group 1 async-generic
interface loopback0
  ip address 30.1.1.1 255.255.0.0
interface serial0
  physical-layer async
  encapsulation bstun
  asp role primary
  bstun group 1
  bstun route all tcp 10.1.1.1
interface serial1
  encapsulation frame-relay
interface serial1.1 point-to-point
  ip address 20.1.1.2 255.255.255.0
  frame-relay interface-dlci 100
ip route 0.0.0.0 0.0.0.0 20.1.1.1
line 1
  speed 4800
  databits 7
  parity even
  stopbits 2
.
.
.
!
end
```

Remoto n. 2 (Cisco 1700 Router)

```
REMOTEN2#show running-config
Building configuration...
bstun peer-name 30.2.1.1
bstun protocol-group 2 async-generic
interface loopback0
  ip address 30.2.1.1 255.255.0.0
interface serial0
  physical-layer async
  encapsulation bstun
  asp role primary
  bstun group 2
  bstun route all tcp 10.1.1.1
interface serial1
  encapsulation frame-relay
interface serial1.1 point-to-point
  ip address 20.2.1.2 255.255.255.0
  frame-relay interface-dlci 100
ip route 0.0.0.0 0.0.0.0 20.2.1.1
line 1
  speed 4800
  databits 7
```

```
parity even
stopbits 2
.
.
.
!
end
```

[Verifica](#)

Attualmente non è disponibile una procedura di verifica per questa configurazione.

[Risoluzione dei problemi](#)

BSTUN riceve un pacchetto nell'interfaccia seriale, lo incapsula e lo invia al router remoto quando viene emesso il comando **bstun route all tcp**. Il pacchetto TCP viene ricevuto sul router remoto e decapsulato. I dati vengono inviati tramite l'interfaccia seriale. Se la connessione non funziona, i dati in arrivo devono prima essere verificati con il pacchetto **debug asp**. I dati ricevuti dal router vengono visualizzati sull'interfaccia seriale. Poiché il router non ha una struttura di protocollo e varia in base al protocollo asincrono, non viene fornito un esempio di debug. Il flusso di dati rilevato dal router deve corrispondere a quello inviato dal dispositivo. Se non corrisponde, molto probabilmente la velocità, i database, la parità o i bit di stop non sono configurati per corrispondere al dispositivo. Ciò può verificarsi anche in assenza di dati ricevuti.

Se si ricevono dati sull'interfaccia seriale, usare il comando **show bstun** per verificare se la connessione è aperta o chiusa. Lo stato aperto con solo pacchetti trasmessi indica che il TCP viene inviato al peer BSTUN remoto. A questo punto, un test ping tra l'indirizzo IP del nome peer BSTUN locale e l'indirizzo IP del nome peer BSTUN remoto verifica se l'indirizzo IP è configurato e funziona correttamente. Se il ping ha esito positivo, sul dispositivo remoto usare il comando **debug asp packet** per determinare se il pacchetto è stato ricevuto e inviato all'interfaccia seriale del dispositivo asincrono.

Per risolvere il problema, completare la procedura seguente.

1. Verificare che i dati vengano ricevuti nel router host con il comando **debug asp packet**.
2. Verificare la connettività IP eseguendo il ping dei ping di origine tra l'indirizzo IP del nome peer bstun e l'indirizzo IP remoto del nome peer BSTUN remoto.
3. Sul dispositivo remoto, verificare che i pacchetti vengano inviati al dispositivo remoto con il comando **debug asp packet**.
4. se il protocollo asincrono non include un indirizzo nei pacchetti inviati al router, può essere utile usare il comando **asp offset-address** nell'interfaccia con il numero di byte appropriato corrispondente alla posizione in cui si trova l'indirizzo nel pacchetto. Il valore predefinito è 0. Ad esempio, se il pacchetto è 01C1ABCDEF, dove C1 è l'indirizzo, l'interfaccia seriale può essere configurata con il comando **asp offset-address 01**. In alcuni casi, ciò consente al router di identificare un pacchetto e aumenta la probabilità che i dati vengano gestiti dal router come pacchetto con frame e non solo come flusso di dati.

[Informazioni correlate](#)

- [Supporto tecnico STUN \(Serial Tunnel\) e BSTUN \(Block Serial Tunnel\)](#)

- [Supporto tecnico – Cisco Systems](#)