

# Tunneling von asynchronen Protokollen in BSTUN-Konfigurationsbeispiel

## Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundinformationen](#)

[Konfigurieren](#)

[Netzwerkdigramm](#)

[Konfigurationen](#)

[Überprüfen](#)

[Fehlerbehebung](#)

[Zugehörige Informationen](#)

## [Einführung](#)

Dedizierte und native async-Protokolle werden bei einer Cisco Implementierung nicht direkt unterstützt. Das async-generische Tunneling für den Block Serial Tunnel (BSTUN) kann diese Daten jedoch nur in begrenztem Maße tunneln.

## [Voraussetzungen](#)

### [Anforderungen](#)

Für dieses Dokument bestehen keine speziellen Anforderungen.

### [Verwendete Komponenten](#)

Die Informationen in diesem Dokument basieren auf den Versionen Software und Hardware:

- Verwenden Sie [Feature Navigator II](#) (nur [registrierte](#) Kunden) und die Option **Search by Feature (Nach Funktion suchen)**.
- Verwenden Sie den [Software Advisor](#) (nur [registrierte](#) Kunden), um nach der für Ihre Hardware benötigten Softwareversion zu suchen.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

## Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps zu Konventionen von Cisco).

## Hintergrundinformationen

Async-Protokolle wie z. B. Diebold's TC500 für die Kommunikation mit Geldautomaten oder das Tunneling von HyperTerminal von einem PC auf einen anderen PC haben keine direkte Unterstützung oder Implementierung in Cisco IOS ®. Wie der Name schon andeutet, ist dies eine generische Implementierung, die in der Lage ist, diese Art von Daten zu übertragen. Dies wird als async-generisch BSTUN bezeichnet und erfordert das IBM- oder Enterprise IOS-Feature-Set.

BSTUN async-generisch wurde ursprünglich entwickelt, um unidirektionale, kleine Pakete von Sicherheitsgeräten an ein Reporting-Gerät zu übertragen. BSTUN async-generisch kann jedoch interaktiven Datenverkehr übertragen. Im Wesentlichen bindet diese Implementierung an native, asynchrone Geräte und empfängt die Daten an die serielle Schnittstelle und dann an einen Speicherpuffer. In regelmäßigen Abständen werden die gepufferten Daten dann in ein TCP-Paket gekapselt und an den BSTUN-Peer gesendet, wo sie entkapselt und an das asynchrone Gerät gesendet werden, das an den Remote-Standort angeschlossen ist.

BSTUN async-generisch ist ein einfacher Vorgang. Der Router kann nicht so konfiguriert werden, dass er Kenntnis über den Start des Frames (SOF), das Ende des Frames (EOF) oder das Adressierungsschema des asynchronen Protokolls hat. Wenn sich der Adressteil des Frames in jedem Frame befindet, ein Byte lang ist und dieselbe Position im Frame hat, kann der Befehl **asp address-offset** an den Router gesendet werden, wo die Adresse im Frame gesucht werden soll, wie im weiteren Verlauf dieses Dokuments beschrieben. In vielen Fällen ist jedoch kein Adressteil im Protokoll enthalten. Wenn der Router keine Kenntnis über die async-Protokollstruktur hat, kann er einzelne Pakete von anderen nicht erkennen, wenn sie nicht durch einen bestimmten Zeitraum voneinander getrennt sind. Zwischen Frames mit 9600 Bit/s sind ca. 40 ms erforderlich, um dem Router eine angemessene Zeitspanne für die korrekte Erkennung eines Pakets von einem anderen zu bieten. Der Router erkennt einfach einen Datenstrom in seine serielle Schnittstelle und packt diese Daten dann in TCP ein. Es besteht keine Möglichkeit, dass der Router Routing-Entscheidungen basierend auf einem bestimmten Aspekt des eingehenden Frames treffen kann. Aus diesem Grund muss BSTUN async-generisch physisch konzipiert sein, sodass nur ein Gerät an die serielle Schnittstelle des Routers angeschlossen ist. Es gibt keine Funktion zur lokalen Bestätigung. BSTUN unterstützt nur das lokale Rack für das IBM3270 BISYNC-Protokoll.

## Konfigurieren

In diesem Abschnitt erhalten Sie Informationen zum Konfigurieren der in diesem Dokument beschriebenen Funktionen.

## Netzwerkdiagramm

In diesem Dokument wird die in diesem Diagramm dargestellte Netzwerkeinrichtung verwendet.



Beide PCs verwenden HyperTerminal von Microsoft. Anstelle eines der PCs kann es sich jedoch um eine Verbindung mit dem Konsolenport eines Cisco Routers handeln. Diese Beispielkonfigurationen stellen Konfigurationen dar, die von Routern implementiert wurden, die zuvor in einem Laborszenario nicht konfiguriert wurden, und zeigen die entsprechenden Teile der erforderlichen Konfiguration an. Diese werden unter der Annahme einer 9600-Bit/s-8N1-Verbindung konfiguriert.

## Konfigurationen

In diesem Dokument werden die in diesem Abschnitt beschriebenen Konfigurationen verwendet.

- Hauptrouter (Cisco 1700 Router)
- Remote-Router (Cisco Router 3640)
- Hauptrouter (Cisco 3600-Router)
- Remote Nr. 1 (Cisco Router der Serie 1700)
- Remote 2 (Cisco Router der Serie 1700)

### Hauptrouter (Cisco 1700 Router)

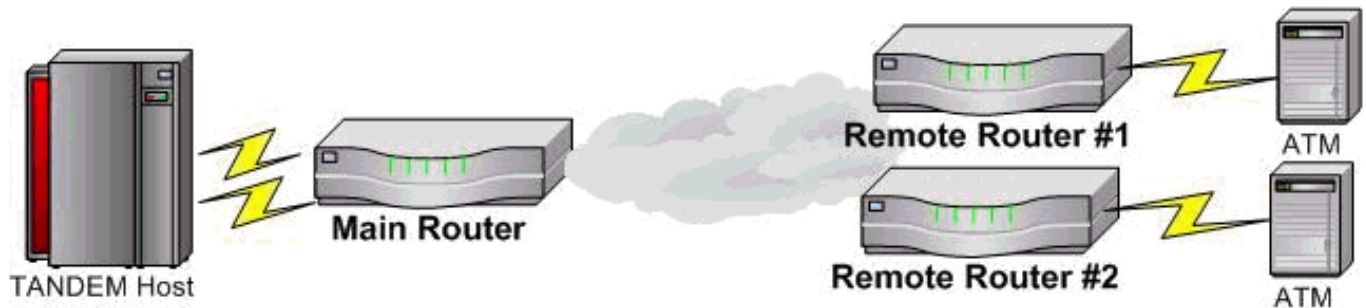
```
main#show running-config
Building configuration...
.
.
.
ip subnet-zero
bstun peer-name 10.1.1.1
bstun protocol-group 1 async-generic
interface loopback0
  ip address 10.1.1.1 255.0.0.0
interface serial0
  physical-layer async
  encapsulation bstun
  asp role secondary
  bstun group 1
  bstun route 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
.
.
.
!
end
```

### Remote-Router (Cisco Router 3640)



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

In diesem Szenario kommuniziert ein Tandem mit Remote-ATM-Geräten. In dieser Beispielformatung führt das async-Protokoll ein 4800 7E2-Protokoll aus, und der mit dem TANDEM verbundene Main-Router ist ein Router der Serie 3600 zu Remote-Routern der Serie 1700. Siehe dieses Netzwerkdiagramm.



### Hauptrouter (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
```

### Remote Nr. 1 (Cisco Router der Serie 1700)

```
REMOTE1#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
```

### Remote 2 (Cisco Router der Serie 1700)

```
REMOTE2#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
```

## Überprüfen

Für diese Konfiguration ist derzeit kein Überprüfungsverfahren verfügbar.

## Fehlerbehebung

BSTUN empfängt ein Paket an die serielle Schnittstelle, kapselt es und sendet dieses TCP-Paket an den Remote-Router, wenn der Befehl **bstun route all tcp** ausgegeben wird. Das TCP-Paket wird am Remote-Router empfangen und entkapselt. Die Daten werden an die serielle Schnittstelle gesendet. Wenn diese Verbindung nicht funktioniert, müssen die eingehenden Daten zuerst mit dem **debug asp**-Paket überprüft werden. Die vom Router empfangenen Daten werden auf der seriellen Schnittstelle angezeigt. Da der Router keine Protokollstruktur aufweist und je nach asynchronem Protokoll variiert, wird kein Beispieldebug bereitgestellt. Der vom Router angezeigte Datenstrom muss mit dem übereinstimmen, was vom Gerät gesendet wird. Wenn sie nicht übereinstimmt, werden die Geschwindigkeit, Datenbanken, Parität oder Stopbits höchstwahrscheinlich nicht so konfiguriert, dass sie mit dem Gerät übereinstimmen. Dies kann auch der Fall sein, wenn keine Daten empfangen werden.

Wenn Daten auf der seriellen Schnittstelle empfangen werden, geben Sie den Befehl **show bstun** ein, um anzuzeigen, ob die Verbindung offen oder geschlossen ist. Der offene Zustand, bei dem nur Pakete übertragen werden, weist darauf hin, dass das TCP an den entfernten BSTUN-Peer gesendet wird. An diesem Punkt überprüft ein Ping-Test von der IP-Adresse des lokalen BSTUN-Peernamens bis zur IP-Adresse des entfernten BSTUN-Peer-Namens, ob die IP-Adresse konfiguriert ist und ordnungsgemäß funktioniert. Wenn der Ping-Test erfolgreich war, geben Sie am Remote-Standort den Befehl **debug asp packet** aus, um festzustellen, ob das Paket empfangen und an die serielle Schnittstelle an das async-Gerät gesendet wird.

Führen Sie diese Schritte aus, um eine Fehlerbehebung durchzuführen.

1. Überprüfen Sie, ob Daten mit dem Befehl **debug asp packet** im Host-Router empfangen werden.
2. Stellen Sie sicher, dass die IP-Verbindung mit Ping-Test-Sourcing-Pings von der IP-Adresse des internen Peer-Namens zur Remote-IP-Adresse des entfernten BSTUN-Peer-Namens hergestellt wird.
3. Überprüfen Sie am Remote-Standort, ob Pakete mit dem Befehl **debug asp packet** an das Remote-Gerät gesendet werden.
4. Wenn das async-Protokoll über eine Adresse verfügt, die in den an den Router gesendeten Paketen enthalten ist, kann es sinnvoll sein, den Befehl **asp offset-address** unter der Schnittstelle mit der entsprechenden Bytnummer auszugeben, die dem Speicherort der Adresse im Paket entspricht. Der Standardwert hierfür ist 0. Wenn das Paket beispielsweise 01C1ABCDEF ist, wobei C1 die Adresse ist, kann die serielle Schnittstelle mit dem Befehl

**asp offset-address 01** konfiguriert werden. In einigen Fällen kann der Router so ein Paket identifizieren und erhöht die Wahrscheinlichkeit, dass der Router die Daten als Framed-Paket und nicht nur als Datenstrom behandelt.

## Zugehörige Informationen

- [STUN \(Serieller Tunnel\) & BSTUN \(Serieller Tunnel blockieren\) Technischer Support](#)
- [Technischer Support - Cisco Systems](#)