

# Exemple de configuration de tunnels de protocoles asynchrones dans BSTUN

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Informations générales](#)

[Configuration](#)

[Diagramme du réseau](#)

[Configurations](#)

[Vérification](#)

[Dépannage](#)

[Informations connexes](#)

## [Introduction](#)

Les protocoles asynchrones dédiés et natifs ne sont pas directement pris en charge par une implémentation Cisco. Cependant, la transmission tunnel BSTUN (Block Serial Tunnel) asynchrone-générique peut fournir une capacité limitée de tunnel de ces données.

## [Conditions préalables](#)

### [Conditions requises](#)

Aucune spécification déterminée n'est requise pour ce document.

### [Components Used](#)

Les informations de ce document sont basées sur les versions de logiciel et matériel suivantes :

- Utilisez [Feature Navigator II](#) (clients [enregistrés](#) uniquement) et l'option **Rechercher par fonction**.
- Utilisez [Software Advisor](#) (clients [enregistrés](#) uniquement) pour rechercher la version logicielle minimale prise en charge requise pour votre matériel.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

## Informations générales

Les protocoles asynchrones tels que le TC500 de Diebold pour communiquer avec des distributeurs de billets ou un HyperTerminal de tunnellation d'un PC à un autre n'ont aucune prise en charge ou implémentation directe dans le logiciel Cisco IOS®. Comme son nom l'indique, il s'agit d'une implémentation générique qui a une certaine capacité à transporter ce type de données. Cette fonctionnalité est appelée BSTUN async-générique et nécessite le jeu de fonctions IBM ou Enterprise IOS.

BSTUN async-generic a été conçu à l'origine pour transporter des petits paquets unidirectionnels des périphériques de sécurité vers un périphérique de reporting. BSTUN async-generic, cependant, peut transporter du trafic interactif. Essentiellement, cette implémentation se connecte aux périphériques natifs asynchrones et reçoit les données dans l'interface série, puis dans une mémoire tampon. Périodiquement, les données mises en mémoire tampon sont ensuite encapsulées dans un paquet TCP et envoyées à l'homologue BSTUN où elles sont décapsulées et envoyées au périphérique asynchrone connecté au site distant.

BSTUN async-generic est une opération simpliste. Le routeur ne peut pas être configuré pour connaître le début de trame (SOF), la fin de trame (EOF) ou le schéma d'adressage du protocole asynchrone. Si la partie adresse de la trame se trouve dans chaque trame, comporte un octet et se trouve au même endroit dans la trame, la commande **asp address-offset** peut être exécutée pour spécifier au routeur où trouver l'adresse dans la trame, comme illustré plus loin dans ce document. Dans de nombreuses situations, cependant, il n'y aura pas de partie adresse dans le protocole. Ne pas connaître la construction du protocole asynchrone signifie que le routeur ne peut pas discerner les paquets individuels des autres s'ils ne sont pas séparés par une période de temps. Environ 40 ms sont nécessaires entre les trames à 9 600 bits/s pour fournir au routeur un temps suffisant pour discerner correctement un paquet d'un autre. Le routeur voit simplement un flux de données dans son interface série, puis encapsule ces données dans TCP. Il n'est pas possible que le routeur puisse prendre des décisions de routage en fonction de chaque aspect de la trame entrante. Ainsi, BSTUN async-generic doit être conçu physiquement de sorte qu'un seul périphérique se connecte à l'interface série du routeur. Il n'existe pas de fonction d'accusé de réception local. BSTUN prend uniquement en charge le serveur local pour le protocole BISYNC IBM3270.

## Configuration

Cette section vous fournit des informations pour configurer les fonctionnalités décrites dans ce document.

## Diagramme du réseau

Ce document utilise la configuration réseau indiquée dans le diagramme suivant.



Les deux ordinateurs utilisent le HyperTerminal de Microsoft ou, à la place de l'un des ordinateurs, ils peuvent être connectés au port de console d'un routeur Cisco. Ces exemples de configuration représentent les configurations mises en oeuvre à partir de routeurs non configurés précédemment dans un scénario de travaux pratiques et montrent les parties pertinentes de la configuration requise. Celles-ci sont configurées en supposant une connexion 8 N1 de 9 600 bits/s.

## Configurations

Ce document utilise les configurations indiquées dans cette section.

- Routeur principal (routeur Cisco 1700)
- Routeur distant (routeur Cisco 3640)
- Routeur principal (routeur Cisco 3600)
- Distant 1 (routeur Cisco 1700)
- Distant n° 2 (routeur Cisco 1700)

### **Routeur principal (routeur Cisco 1700)**

```
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
```

### **Routeur distant (routeur Cisco 3640)**

```

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

```

**Remarque :** lorsque vous émettez la commande **Physical-layer async** sur l'interface série, une ligne TTY est affectée à l'interface série. Cette définition de ligne TTY est l'emplacement où les bases de données, les arrêts, la parité et la vitesse sont configurés. Il s'agit de la formule permettant de déterminer quelle ligne correspond à l'interface série.

$line\# = (slot\# \times 32) + interface\# + 1$

La ligne show dans le résultat de la configuration du routeur distant indique dans la colonne de droite le numéro de ligne correspondant. Serial2/0 est représenté par la ligne 65 et les définitions physiques de cette liaison sont configurées sous la ligne 65

```

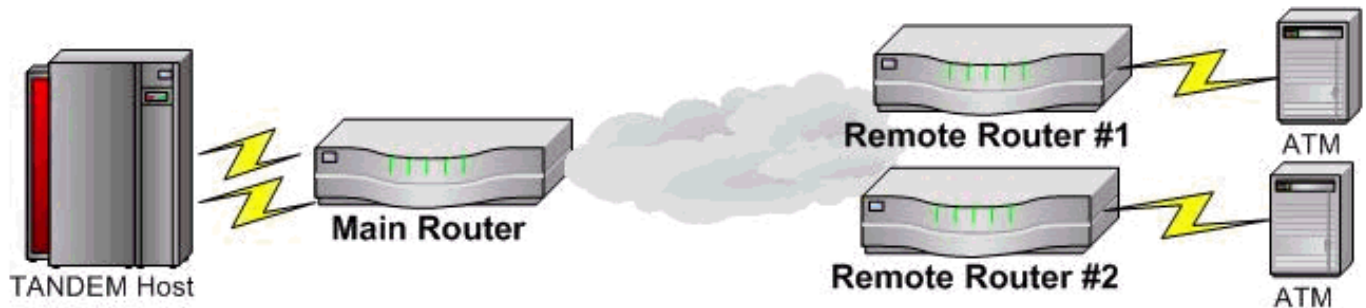
REMOTE#sh line

```

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

Dans ce scénario, un tandem communique avec des périphériques ATM distants. Dans cet exemple de configuration, le protocole asynchrone exécute un protocole 4800 7E2 et le routeur principal connecté au TANDEM est un routeur de la gamme 3600 vers les routeurs distants de la gamme 1700. Reportez-vous à ce schéma de réseau.



### Routeur principal (routeur Cisco 3600)

```
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
```

### Distant 1 (routeur Cisco 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
```

### Distant n° 2 (routeur Cisco 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
```

## Vérification

Aucune procédure de vérification n'est disponible pour cette configuration.

## Dépannage

BSTUN reçoit un paquet dans l'interface série, l'encapsule et envoie ce paquet TCP au routeur distant lorsque la commande **bstun route all tcp** est exécutée. Le paquet TCP est reçu sur le routeur distant et décapsulé. Les données sont envoyées sur l'interface série. Si cette connexion ne fonctionne pas, les données entrantes doivent d'abord être vérifiées avec le paquet **debug asp**. Vous voyez les données reçues par le routeur sur l'interface série. Comme le routeur n'a pas de construction de protocole et varie selon le protocole asynchrone, l'exemple de débogage n'est pas fourni. Le flux de données vu par le routeur doit correspondre à celui envoyé par le périphérique. Si elle ne correspond pas, plus que probablement, la vitesse, les bases de données, la parité ou les arrêts ne sont pas configurés pour correspondre au périphérique. Cela peut également être le cas si aucune donnée n'est reçue.

Si des données sont reçues sur l'interface série, émettez la commande **show bstun** afin d'afficher si la connexion est ouverte ou fermée. L'état Open avec seulement les paquets transmis indique que le TCP est envoyé à l'homologue BSTUN distant. À ce stade, un test ping de l'adresse IP du nom d'homologue BSTUN local à l'adresse IP du nom d'homologue BSTUN distant vérifie si l'adresse IP est configurée et fonctionne correctement. Si le test ping réussit, alors à distance, émettez la commande **debug asp packet** afin de déterminer si le paquet est reçu et envoyé sur l'interface série au périphérique asynchrone.

Effectuez ces étapes afin de procéder au dépannage .

1. Vérifiez que les données sont reçues dans le routeur hôte à l'aide de la commande **debug asp packet**.
2. Assurez-vous que la connectivité IP est établie avec les requêtes ping de test de l'adresse IP du meilleur nom d'homologue vers l'adresse IP distante du nom d'homologue BSTUN distant.
3. À la télécommande, vérifiez que les paquets sont envoyés au périphérique distant à l'aide de la commande **debug asp packet**.
4. Si le protocole asynchrone contient une adresse dans les paquets envoyés au routeur, il peut être utile d'émettre la commande **asp offset-address** sous l'interface avec le numéro d'octet approprié correspondant à l'emplacement où l'adresse est contenue dans le paquet. La valeur par défaut de ce paramètre est 0. Par exemple, si le paquet est 01C1ABCDEF, où C1 est l'adresse, l'interface série peut être configurée avec la commande **asp offset-address 01**. Dans certains cas, cela permet au routeur d'identifier un paquet et augmente la probabilité que le routeur traite les données en tant que paquet tramé et pas seulement en tant que flux

de données.

## Informations connexes

- [Support technique STUN \(Serial Tunnel\) et BSTUN \(Block Serial Tunnel\)](#)
- [Support technique - Cisco Systems](#)