

Travail en équipe de deux

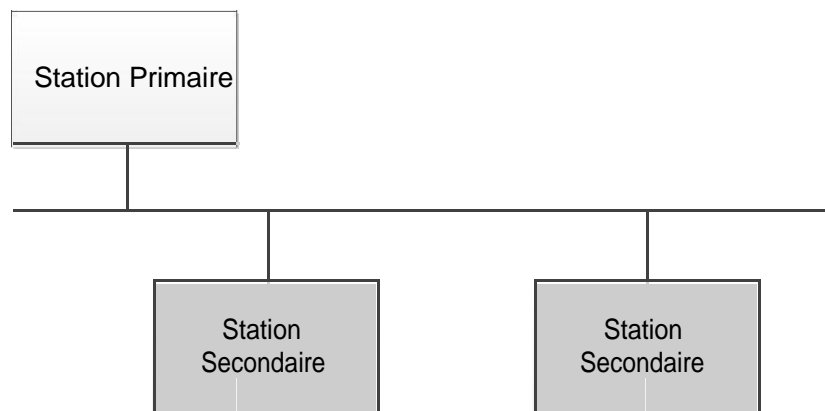
Dû le 25 mars 2023

CEG3585 – Lab 4

Le HDLC et le réseau multipoint

Objectifs: (1) se familiariser avec le réseau multipoint qui contient une station primaire et plusieurs stations secondaires; (2) mieux comprendre l'importance et rôle des protocoles; (3) maîtriser le mécanisme de la fenêtre d'anticipation; et (4) étudier un protocole concret de la couche liaison de données, le HDLC.

Le réseau de ce laboratoire est réalisé en simulant la couche physique avec la programmation socket dans les classes `PhysicalLayer` et `PhysicalLayerServer`. Les entités de la couche liaison de données sont réalisées par les classes `PrimaryHDLCDataLink` et `SecondaryHDLCDataLink`. Les classes `PrimaryStation` et `SecondaryStation` représentent les applications qui utilisent le service de la couche liaison de données pour échanger un message. Le réseau comprend 3 stations, une station primaire et deux stations secondaires branchées par une liaison multipoint.



Votre tâche est de compléter la méthode `dlDataReques()` de la classe `SecondaryHDLCDataLink`. Toutes les autres méthodes des classes sont complètes. La méthode `dlDataRequest()` de la station secondaire permet d'envoyer un message à la station primaire avec plusieurs trames-I du HDLC. Voici des consignes pour compléter votre tâche:

- 1) La reprise d'erreur n'est pas considérée. Donc le champ FCS n'est pas inclus dans la trame-I et les trames REJ et SREJ ne sont pas utilisées.

- 2) Le champ de données (information) de la trame-I est limité à 32 octets (défini par la constante `HdlcDefs.MAX_DATA_SIZE_BYTES`). Le message à transmettre doit donc être divisé et transmis en plusieurs trames-I. La méthode `BitString.splitString()` permet de séparer une chaîne de caractères (`String`) en morceaux (produit un tableau de `String`).
- 3) La variable `vs` détermine le numéro de séquence pour la prochaine trame-I à transmettre. Elle est incrémentée après la création/transmission de chaque nouvelle trame-I. Pour circuler à travers les numéros de séquences, utilisez l'opération modulo avec `HdlcDefs.SNUM_DATA_SIZE` qui donne le nombre de numéros de séquences.
- 4) Une fenêtre d'anticipation est utilisée avec une taille de fenêtre 4 (définie dans la variable `windowSize`). Le bord droit de la fenêtre est défini par `rhsWindow` qui donne le numéro de séquence juste à droite de la fenêtre ; donc la variable `vs` (le prochain numéro de séquence valide) ne doit pas être égale à `rhsWindow`. Le bord gauche de la fenêtre (le premier numéro de séquence dans la fenêtre) peut être déterminé à partir de `rhsWindow` :
 - a. Si $rhsWindow - windowSize \geq 0$, alors le bord gauche = `rhsWindow - windowSize`
 - b. Autrement le bord gauche = `rhsWindow - windowSize + HdlcDefs.SNUM_SIZE_COUNT`.
- 5) La classe `HdlcDefs` contient des constantes pour faire la mise en trame ainsi que faire l'extraction des champs de trames reçues.
- 6) Pour transmettre le message (référé par `sdu`), la station secondaire devrait :
 - a. Attendre une sonde (trame-RR avec le bit P/F = 1) avant de transmettre les trames-I.
 - b. Divisez le message en morceaux de 32 octets (voir le point 2).
 - c. Traduire chaque morceau en un `bitString` (voir la méthode `BitString.stringToBitString()`).
 - d. Dans une boucle :
 - i. S'il reste un morceau de message à transmettre et qu'un numéro de séquence est disponible, créer et transmettre une trame-I pour transmettre le prochain morceau de message (en bits), ajouter la trame au tampon de trames `frameBuffer`, et mettre à jour la variable `vs`;
 - ii. Si une trame RR est reçue (utilisez la méthode `getFrame(true)` pour interroger la couche physique pour une trame), extraire le numéro d'acquittement et ajuster la fenêtre d'anticipation (la variable `rhsWindow`), et enlever le nombre approprié de trames du tampon `frameBuffer`.
 - iii. Après la transmission d'une trame-I et la réception d'une trame-RR, exécuter la méthode `displayDataXchgngState` qui affichera les valeurs de `vs`, `vr`, les bordures de la fenêtre d'anticipation et le nombre de messages dans le tampon `frameBuffer`. Les appels appropriés sont déjà présents dans le code fourni. Voir la sortie des stations secondaires dans l'annexe.
 - iv. Rester dans la boucle sous les conditions suivantes : il existe un morceau de message à transmettre, il existe une trame-I à acquitter (le tampon de trames n'est pas vide).
- 7) Quelques indices pour réaliser l'étape 6:
 - a. Créer une méthode séparée pour détecter l'arrivée d'une trame-RR. La méthode devrait ignorer les trames autres que les trames-RR. La méthode devrait avoir un paramètre booléen qui a la valeur vraie si la méthode bloque jusqu'à l'arrivée d'une trame-RR (voir 6a) et faux pour interroger la couche physique l'arrivée de la trame-RR (voir 6d-ii). Un gabarit `getFrame()` est fourni.
 - b. Créer une méthode séparée pour déterminer à partir de `nr` (numéro d'acquittement reçu dans une trame-RR) le nombre de trames acquittées. Ce nombre est ensuite utilisé pour mettre à jour la fenêtre d'anticipation (`rhsWindow`) et enlever les trame-I du tampon `frameBuffer`. Un gabarit `checkNr()` est fourni.

Annexe A – Sortie des programmes

Pour exécuter le logiciel fournie, ouvrez 4 fenêtres « cmd » et dans chaque fenêtre exécuter les commandes suivantes dans l'ordre donné et chacune dans une fenêtre différente :

```
java PhysicalLayerServer
java SecondaryStation 1
java SecondaryStation 2
java PrimaryStation
```

Vous pouvez examiner la sortie des programmes soit à l'écran, soit dans des fichiers si vous redirigez les sorties à des fichiers (e.g. java PrimaryStation >PrimLog.txt).

Sortie de la Station Primaire (PrimaryStation)

```
-----Connection to Station 1-----
Primary Station: Requesting connection to station 1
Data Link Layer: prepared SNRM frame >01111110 00000001 11001001 01111110<
Physical layer: transmitted frame >01111110 00000001 11001001 01111110<
Physical layer: received frame >01111110 00000001 11001110 01111110<
Data Link Layer: received UA frame >01111110 00000001 11001110 01111110<
Primary Station: Received connect confirmation from station 1
-----
-----Connection to Station 2-----
Primary Station: Requesting connection to station 2
Data Link Layer: prepared SNRM frame >01111110 00000010 11001001 01111110<
Physical layer: transmitted frame >01111110 00000010 11001001 01111110<
Physical layer: received frame >01111110 00000010 11001110 01111110<
Data Link Layer: received UA frame >01111110 00000010 11001110 01111110<
Primary Station: Received connect confirmation from station 2
-----
-----Get Message from Station 2-----
Data Link Layer: sending RR frame (poll) >01111110 00000010 10001000 01111110<
Physical layer: transmitted frame >01111110 00000010 10001000 01111110<
Physical layer: received frame >01111110 00000010 00000000 01010011 ... 01101111 01111110<
Data Link Layer: received I frame >01111110 00000010 00000000 01010011 ... 01101111 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000001 01111110<
Physical layer: transmitted frame >01111110 00000010 10000001 01111110<
Physical layer: received frame >01111110 00000010 00010000 01110010 ... 01001000 01111110<
Data Link Layer: received I frame >01111110 00000010 00010000 01110010 ... 01001000 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000010 01111110<
Physical layer: transmitted frame >01111110 00000010 10000010 01111110<
Physical layer: received frame >01111110 00000010 00100000 01000100 ... 01110000 01111110<
Data Link Layer: received I frame >01111110 00000010 00100000 01000100 ... 01110000 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000011 01111110<
Physical layer: transmitted frame >01111110 00000010 10000011 01111110<
Physical layer: received frame >01111110 00000010 00110000 01101111 ... 01101001 01111110<
Data Link Layer: received I frame >01111110 00000010 00110000 01101111 ... 01101001 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000100 01111110<
Physical layer: transmitted frame >01111110 00000010 10000100 01111110<
Physical layer: received frame >01111110 00000010 01000000 01100011 ... 01101110 01111110<
Data Link Layer: received I frame >01111110 00000010 01000000 01100011 ... 01101110 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000101 01111110<
Physical layer: transmitted frame >01111110 00000010 10000101 01111110<
Physical layer: received frame >01111110 00000010 01010000 01100101 ... 01110000 01111110<
Data Link Layer: received I frame >01111110 00000010 01010000 01100101 ... 01110000 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000110 01111110<
Physical layer: transmitted frame >01111110 00000010 10000110 01111110<
Physical layer: received frame >01111110 00000010 01100000 01110010 ... 01110100 01111110<
Data Link Layer: received I frame >01111110 00000010 01100000 01110010 ... 01110100 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000111 01111110<
Physical layer: transmitted frame >01111110 00000010 10000111 01111110<
Physical layer: received frame >01111110 00000010 01110000 01110111 ... 01110011 01111110<
Data Link Layer: received I frame >01111110 00000010 01110000 01110111 ... 01110011 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000000 01111110<
Physical layer: transmitted frame >01111110 00000010 10000000 01111110<
Physical layer: received frame >01111110 00000010 00000000 01110011 ... 01100100 01111110<
Data Link Layer: received I frame >01111110 00000010 00000000 01110011 ... 01100100 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000001 01111110<
Physical layer: transmitted frame >01111110 00000010 10000001 01111110<
```

Physical layer: received frame >01111110 00000010 00010000 01100001 ... 01101110 01111110<
Data Link Layer: received I frame >01111110 00000010 00010000 01100001 ... 01101110 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000010 01111110<
Physical layer: transmitted frame >01111110 00000010 10000010 01111110<
Physical layer: received frame >01111110 00000010 00100000 01111001 ... 01101111 01111110<
Data Link Layer: received I frame >01111110 00000010 00100000 01111001 ... 01101111 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000011 01111110<
Physical layer: transmitted frame >01111110 00000010 10000011 01111110<
Physical layer: received frame >01111110 00000010 00111000 00100000 ... 01100101 01111110<
Data Link Layer: received I frame >01111110 00000010 00111000 00100000 ... 01100101 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000010 10000100 01111110<
Physical layer: transmitted frame >01111110 00000010 10000100 01111110<
Primary Station: Received from Station 2 >Station 2 to Primary: Message for testing data transfer. The HDLC protocol is designed to support communication over a physical link between physically connected stations.

In this lab a primary station is connected to two secondary stations.
This message will be sent by each Secondary station to the Primary.
Many HDLC I frames shall be used to send this message< -----

-----Get Message from Station 1-----

Data Link Layer: sending RR frame (poll) >01111110 00000001 10001100 01111110<
Physical layer: transmitted frame >01111110 00000001 10001100 01111110<
Physical layer: received frame >01111110 00000001 00000000 01010011 ... 01101111 01111110<
Data Link Layer: received I frame >01111110 00000001 00000000 01010011 ... 01101111 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000001 01111110<
Physical layer: transmitted frame >01111110 00000001 10000001 01111110<
Physical layer: received frame >01111110 00000001 00010000 01110010 ... 01001000 01111110<
Data Link Layer: received I frame >01111110 00000001 00010000 01110010 ... 01001000 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000010 01111110<
Physical layer: transmitted frame >01111110 00000001 10000010 01111110<
Physical layer: received frame >01111110 00000001 00100000 01000100 ... 01110000 01111110<
Data Link Layer: received I frame >01111110 00000001 00100000 01000100 ... 01110000 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000011 01111110<
Physical layer: transmitted frame >01111110 00000001 10000011 01111110<
Physical layer: received frame >01111110 00000001 00110000 01101111 ... 01101001 01111110<
Data Link Layer: received I frame >01111110 00000001 00110000 01101111 ... 01101001 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000100 01111110<
Physical layer: transmitted frame >01111110 00000001 10000100 01111110<
Physical layer: received frame >01111110 00000001 01010000 01100101 ... 01110000 01111110<
Data Link Layer: received I frame >01111110 00000001 01010000 01100101 ... 01110000 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000110 01111110<
Physical layer: transmitted frame >01111110 00000001 10000110 01111110<
Physical layer: received frame >01111110 00000001 01100000 01110010 ... 01110100 01111110<
Data Link Layer: received I frame >01111110 00000001 01100000 01110010 ... 01110100 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000111 01111110<
Physical layer: transmitted frame >01111110 00000001 10000111 01111110<
Physical layer: received frame >01111110 00000001 01110000 01110111 ... 01110011 01111110<
Data Link Layer: received I frame >01111110 00000001 01110000 01110111 ... 01110011 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000000 01111110<
Physical layer: transmitted frame >01111110 00000001 10000000 01111110<
Physical layer: received frame >01111110 00000001 00000000 01110011 ... 01100100 01111110<
Data Link Layer: received I frame >01111110 00000001 00000000 01110011 ... 01100100 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000001 01111110<
Physical layer: transmitted frame >01111110 00000001 10000001 01111110<
Physical layer: received frame >01111110 00000001 00010000 01100001 ... 01101110 01111110<
Data Link Layer: received I frame >01111110 00000001 00010000 01100001 ... 01101110 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000010 01111110<
Physical layer: transmitted frame >01111110 00000001 10000010 01111110<
Physical layer: received frame >01111110 00000001 00100000 01111001 ... 01101111 01111110<
Data Link Layer: received I frame >01111110 00000001 00100000 01111001 ... 01101111 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000011 01111110<
Physical layer: transmitted frame >01111110 00000001 10000011 01111110<
Physical layer: received frame >01111110 00000001 00111000 00100000 ... 01100101 01111110<
Data Link Layer: received I frame >01111110 00000001 00111000 00100000 ... 01100101 01111110<
Data Link Layer: prepared RR frame(ack) >01111110 00000001 10000100 01111110<
Physical layer: transmitted frame >01111110 00000001 10000100 01111110<

Primary Station: Received from Station 2 >Station 1 to Primary: Message for testing data transfer. The HDLC protocol is designed to support communication over a physical link between physically connected stations.

In this lab a primary station is connected to two secondary stations.
This message will be sent by each Secondary station to the Primary.
Many HDLC I frames shall be used to send this message< -----

----- Disconnect Station 1-----

Primary Station: Requesting disconnect from station 1
Data Link Layer: prepared DISC frame >01111110 00000001 11000010 01111110<
Physical layer: transmitted frame >01111110 00000001 11000010 01111110<

----- Disconnect Station 2-----

Primary Station: Requesting disconnect from station 2
Data Link Layer: prepared DISC frame >01111110 00000010 11000010 01111110<
Physical layer: transmitted frame >01111110 00000010 11000010 01111110<

Sortie de la Station Secondaire 1 (SecondaryStation)

-----Connection to Primary-----

Physical layer: received frame >01111110 00000001 11001001 01111110<
Data Link Layer: received SNRM frame >01111110 00000001 11001001 01111110<
Secondary Station (1): Received conenct indication
Secondary Station (1): Issuing connect confirmation
Data Link Layer: prepared UA frame >01111110 00000001 11001110 01111110<
Physical layer: transmitted frame >01111110 00000001 11001110 01111110<

-----Send Message To Primary-----

Secondary Station (1): Issuing data request
Physical layer: received frame >01111110 00000010 11001001 01111110<
Physical layer: received frame >01111110 00000010 11001110 01111110<
Physical layer: received frame >01111110 00000010 10001000 01111110<
Physical layer: received frame >01111110 00000010 00000000 01010011 ... 01101111 01111110<
Physical layer: received frame >01111110 00000010 00010000 01110010 ... 01001000 01111110<
Physical layer: received frame >01111110 00000010 00100000 01000100 ... 01110000 01111110<
Physical layer: received frame >01111110 00000010 00110000 01101111 ... 01101001 01111110<
Physical layer: received frame >01111110 00000010 10000001 01111110<
Physical layer: received frame >01111110 00000010 10000010 01111110<
Physical layer: received frame >01111110 00000010 01000000 01100011 ... 01101110 01111110<
Physical layer: received frame >01111110 00000010 01010000 01100101 ... 01110000 01111110<
Physical layer: received frame >01111110 00000010 10000011 01111110<
Physical layer: received frame >01111110 00000010 10000100 01111110<
Physical layer: received frame >01111110 00000010 10000101 01111110<
Physical layer: received frame >01111110 00000010 01100000 01110010 ... 01110100 01111110<
Physical layer: received frame >01111110 00000010 01110000 01110111 ... 01110011 01111110<
Physical layer: received frame >01111110 00000010 00000000 01110011 ... 01100100 01111110<
Physical layer: received frame >01111110 00000010 10000110 01111110<
Physical layer: received frame >01111110 00000010 00010000 01100001 ... 01101110 01111110<
Physical layer: received frame >01111110 00000010 10000111 01111110<
Physical layer: received frame >01111110 00000010 10000000 01111110<
Physical layer: received frame >01111110 00000010 10000001 01111110<
Physical layer: received frame >01111110 00000010 00100000 01111001 ... 01101111 01111110<
Physical layer: received frame >01111110 00000010 00111000 00100000 ... 01100101 01111110<
Physical layer: received frame >01111110 00000010 10000010 01111110<
Physical layer: received frame >01111110 00000010 10000011 01111110<
Physical layer: received frame >01111110 00000010 10000100 01111110<
Physical layer: received frame >01111110 00000001 10001100 01111110<
Data Link Layer: Station 1: Data Link Layer: prepared and buffered I frame >01111110 00000001 00000000 01010011 ... 01101111 01111110<
v(s) = 1, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 1
Physical layer: transmitted frame >01111110 00000001 00000000 01010011 ... 01101111 01111110<
Data Link Layer: Station 1: Data Link Layer: prepared and buffered I frame >01111110 00000001 00010000 01110010 ... 01001000 01111110<
v(s) = 2, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 2
Physical layer: transmitted frame >01111110 00000001 00010000 01110010 ... 01001000 01111110<
Data Link Layer: Station 1: Data Link Layer: prepared and buffered I frame >01111110 00000001 00100000 01000100 ... 01110000 01111110<
v(s) = 3, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 3
Physical layer: transmitted frame >01111110 00000001 00100000 01000100 ... 01110000 01111110<

[illegible]

Data Link Layer: Station 1: received an RR frame (ack) >01111110 00000001 10000100 01111110<
v(s) = 4, v(r) = 0, Window: lhs=4 rhs=0, Number frames buffered = 0

-----Disconnection-----

Physical layer: received frame >01111110 00000001 11000010 01111110<
Data Link Layer: received DISC frame >01111110 00000001 11000010 01111110<
Secondary Station (1): Received disconnect indication

Sortie de la Station Secondaire 2 (SecondaryStation)

-----Connection to Primary-----

Physical layer: received frame >01111110 00000001 11001001 01111110<
Physical layer: received frame >01111110 00000001 11001110 01111110<
Physical layer: received frame >01111110 00000010 11001001 01111110<
Data Link Layer: received SNRM frame >01111110 00000010 11001001 01111110<
Secondary Station (2): Received connect indication
Secondary Station (2): Issuing connect confirmation
Data Link Layer: prepared UA frame >01111110 00000010 11001110 01111110<
Physical layer: transmitted frame >01111110 00000010 11001110 01111110<

-----Send Message To Primary-----

Secondary Station (2): Issuing data request
Physical layer: received frame >01111110 00000010 10001000 01111110<
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110 00000010 00000000 01010011 ... 01101111 01111110<
v(s) = 1, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 1
Physical layer: transmitted frame >01111110 00000010 00000000 01010011 ... 01101111 01111110<
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110 00000010 00010000 01110010 ... 01001000 01111110<
v(s) = 2, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 2
Physical layer: transmitted frame >01111110 00000010 00010000 01110010 ... 01001000 01111110<
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110 00000010 00100000 01000100 ... 01110000 01111110<
v(s) = 3, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 3
Physical layer: transmitted frame >01111110 00000010 00100000 01000100 ... 01110000 01111110<
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110 00000010 00110000 01101111 ... 01101001 01111110<
v(s) = 4, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 00110000 01101111 ... 01101001 01111110<
Physical layer: received frame >01111110 00000010 10000001 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000001 01111110<
v(s) = 4, v(r) = 0, Window: lhs=1 rhs=5, Number frames buffered = 3
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110 00000010 01000000 01100011 ... 01101110 01111110<
v(s) = 5, v(r) = 0, Window: lhs=1 rhs=5, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 01000000 01100011 ... 01101110 01111110<
Physical layer: received frame >01111110 00000010 10000010 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000010 01111110<
v(s) = 5, v(r) = 0, Window: lhs=2 rhs=6, Number frames buffered = 3
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110 00000010 01010000 01100101 ... 01110000 01111110<
v(s) = 6, v(r) = 0, Window: lhs=2 rhs=6, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 01010000 01100101 ... 01110000 01111110<
Physical layer: received frame >01111110 00000010 10000011 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000011 01111110<
v(s) = 6, v(r) = 0, Window: lhs=3 rhs=7, Number frames buffered = 3
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110 00000010 01100000 01110010 ... 01110100 01111110<
v(s) = 7, v(r) = 0, Window: lhs=3 rhs=7, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 01100000 01110010 ... 01110100 01111110<
Physical layer: received frame >01111110 00000010 10000100 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000100 01111110<
v(s) = 7, v(r) = 0, Window: lhs=4 rhs=0, Number frames buffered = 3
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110 00000010 01110000 01110111 ... 01110011 01111110<
v(s) = 0, v(r) = 0, Window: lhs=4 rhs=0, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 01110000 01110111 ... 01110011 01111110<
Physical layer: received frame >01111110 00000010 10000101 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000101 01111110<
v(s) = 0, v(r) = 0, Window: lhs=5 rhs=1, Number frames buffered = 3

```

Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110
00000010 00000000 01110011 ... 01100100 01111110<
    v(s) = 1, v(r) = 0, Window: lhs=5 rhs=1, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 00000000 01110011 ... 01100100 01111110<
Physical layer: received frame >01111110 00000010 10000110 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000110 01111110<
    v(s) = 1, v(r) = 0, Window: lhs=6 rhs=2, Number frames buffered = 3
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110
00000010 00010000 01100001 ... 01101110 01111110<
    v(s) = 2, v(r) = 0, Window: lhs=6 rhs=2, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 00010000 01100001 ... 01101110 01111110<
Physical layer: received frame >01111110 00000010 10000111 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000111 01111110<
    v(s) = 2, v(r) = 0, Window: lhs=7 rhs=3, Number frames buffered = 3
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110
00000010 00100000 01111001 ... 01101111 01111110<
    v(s) = 3, v(r) = 0, Window: lhs=7 rhs=3, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 00100000 01111001 ... 01101111 01111110<
Physical layer: received frame >01111110 00000010 10000000 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000000 01111110<
    v(s) = 3, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 3
Data Link Layer: Station 2: Data Link Layer: prepared and buffered I frame >01111110
00000010 00111000 00100000 ... 01100101 01111110<
    v(s) = 4, v(r) = 0, Window: lhs=0 rhs=4, Number frames buffered = 4
Physical layer: transmitted frame >01111110 00000010 00111000 00100000 ... 01100101 01111110<
Physical layer: received frame >01111110 00000010 10000001 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000001 01111110<
    v(s) = 4, v(r) = 0, Window: lhs=1 rhs=5, Number frames buffered = 3
Physical layer: received frame >01111110 00000010 10000010 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000010 01111110<
    v(s) = 4, v(r) = 0, Window: lhs=2 rhs=6, Number frames buffered = 2
Physical layer: received frame >01111110 00000010 10000011 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000011 01111110<
    v(s) = 4, v(r) = 0, Window: lhs=3 rhs=7, Number frames buffered = 1
Physical layer: received frame >01111110 00000010 10000100 01111110<
Data Link Layer: Station 2: received an RR frame (ack) >01111110 00000010 10000100 01111110<
    v(s) = 4, v(r) = 0, Window: lhs=4 rhs=0, Number frames buffered = 0
-----
-----Disconnection-----
Physical layer: received frame >01111110 00000001 10001100 01111110<
Physical layer: received frame >01111110 00000001 00000000 01010011 ... 01101111 01111110<
Physical layer: received frame >01111110 00000001 00010000 01110010 ... 01001000 01111110<
Physical layer: received frame >01111110 00000001 00100000 01000100 ... 01110000 01111110<
Physical layer: received frame >01111110 00000001 00110000 01101111 ... 01101001 01111110<
Physical layer: received frame >01111110 00000001 10000001 01111110<
Physical layer: received frame >01111110 00000001 10000010 01111110<
Physical layer: received frame >01111110 00000001 10000011 01111110<
Physical layer: received frame >01111110 00000001 10000100 01111110<
Physical layer: received frame >01111110 00000001 01000000 01100011 ... 01101110 01111110<
Physical layer: received frame >01111110 00000001 01010000 01100101 ... 01110000 01111110<
Physical layer: received frame >01111110 00000001 01100000 01110010 ... 01110100 01111110<
Physical layer: received frame >01111110 00000001 01110000 01110111 ... 01110011 01111110<
Physical layer: received frame >01111110 00000001 10000101 01111110<
Physical layer: received frame >01111110 00000001 10000110 01111110<
Physical layer: received frame >01111110 00000001 10000111 01111110<
Physical layer: received frame >01111110 00000001 10000000 01111110<
Physical layer: received frame >01111110 00000001 00000000 01110011 ... 01100100 01111110<
Physical layer: received frame >01111110 00000001 00010000 01100001 ... 01101110 01111110<
Physical layer: received frame >01111110 00000001 00100000 01111001 ... 01101111 01111110<
Physical layer: received frame >01111110 00000001 00111000 00100000 ... 01100101 01111110<
Physical layer: received frame >01111110 00000001 10000001 01111110<
Physical layer: received frame >01111110 00000001 10000010 01111110<
Physical layer: received frame >01111110 00000001 10000011 01111110<
Physical layer: received frame >01111110 00000001 10000100 01111110<
Physical layer: received frame >01111110 00000001 11000010 01111110<
Physical layer: received frame >01111110 00000010 11000010 01111110<
Data Link Layer: received DISC frame >01111110 00000010 11000010 01111110<
Secondary Station (2): Received disconnect indication -----
-----

```


Sortie de la couche physique (PhysicalLayerServer)

Physical Layer Server starting on port 4444

Connection from /169.254.151.92 accepted.

Accepted client

Physical Layer Server: connection from Physical Layer Client 0

Connection from /169.254.151.92 accepted.

Accepted client

Physical Layer Server: connection from Physical Layer Client 1

Connection from /169.254.151.92 accepted.

Accepted client

Physical Layer Server: connection from Physical Layer Client 2

```
Physical Layer Server: received frame from client 2: >011111110000000011100100101111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 0: >0111111100000000111100111001111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 2: >011111110000000101100100101111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 1: >0111111100000001011001111001111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 2: >011111110000000101000100001111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 1:
```

```

>01111111000000010000000000101001101111010001100001011101000110100101101111011011100010000001100100010000001110100011011110010000001010
000011100100110100101101101100001011100100111100100111010001000000100110101100101011100110111001101100001011001110110010100100000011
0011001101111011111110<, sending to other clients.

```

```
Physical Layer Server: received frame from client 1:
```

```

>01111111000000010000100000111001000100000011101000110010101110011011101000110100101101110011001110010000001100100011000010111010001100
00100100000011101000111001001100001011011100111001101100110011001010111001000101110001000000000110100001010010101000110100001100101001
0000001001000011111110<, sending to other clients.

```

```
Physical Layer Server: received frame from client 1:
```

```
>01111111000000010001000000100010001001100010000110010000001110000011100100110111101110100011011110110001101101111011011000010000001101
00101110011001000000110010001100101011100110110100101100111011011100110010101100100001000000111010001101111001000000111001101110101011
1000001110000011111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 1:
```

```

>0111111000000010001100000110111101110010011101000010000000110001101101111011011011011011011011011011100110100010110000101110
10001101001011011110110111000100000011011110111011001100101011100100010000000001101000010100110000100100000011100000110100001111001011
100110110100101111110<, sending to other clients.

```

```
Physical Layer Server: received frame from client 2: >011111110000000101000000101111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 2: >011111110000000101000001001111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 1:
```

```

>0111111100000001001000000011000110111000010110110000100000001101100011010010110111001101011001000000110001001100101011101000111011101100
10101100101011011100010000001110000011010000111100101110011011010010110001101100001011011000110110001111001001000000110001101101111011
0111001101111001111110<, sending to other clients.

```

```
Physical Layer Server: received frame from client 1:
```

```
01111111000000010010100000110010101100011011101000110010101100100001000000111001101110100011000010111010001101001011011110110111001110
01100101110000011010000101001001001011011100010000001110100011010000110100101110011001000000110110001100001011000100010000001100001001
0000001110000011111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 2: >011111110000000101000001101111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 2: >011111110000000101000010001111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 2: >011111110000000101000010101111110<, sending to other clients.
```

```
Physical Layer Server: received frame from client 1:
```

```

>01111111000000010011000000111001001101001011011010110000101110010011110010010000001110011011101000110000101110100011010010110111101101
11000100000011010010111001100100000011000110110111101101110011011100110010101100011011101000110010101100100001000000111010001101111001
0000001110100011111110<, sending to other clients.

```

```
Physical Layer Server: received frame from client 1:
```

```
>011111110000000010011100000011101110111011110010000001110011011100101011100011011011111011011100110010001100001011110010011110010010000001110
```

01101110100011000010111010001101001011011110110111001110011001011100000110100001010010101000110100001101001011100110010000001101101011
001010111001101111110<, sending to other clients.
Physical Layer Server: received frame from client 1:
>011111100000001000000000111001101100001011001110110010100100000011101110110100101101100011011000010000001100010011001010010000001110
0110110010101101110011101000010000001100010011110010010000001100101011000010110001101101000001000000101001101100101011000110110111011
011100110010001111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000101000011001111110<, sending to other clients.
Physical Layer Server: received frame from client 1:
>01111110000000100001000001100001011100100111100100100000011100110111010001100001011101000110100101101110110001000000111010001101
11100100000011101000110100001100101001000000101000011100100110100101101101011000011010000101001001101011
000010110111001111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000101000011101111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >0111111000000010100000001111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000101000000101111110<, sending to other clients.
Physical Layer Server: received frame from client 1:
>0111111000000010001000000111100100100000010010000100010001001100010000110010000001001001001000000110011001110010011000010110110101100
10101110011001000000111001101101000011000010110110001101100001000000110001001100101001000000111010101110011011001010110010000100000011
10100011011101111110<, sending to other clients.
Physical Layer Server: received frame from client 1:
>0111111000000010001110000010000001110011011001010110111001100100001000000111010001101000011010010111001100100000011011010110010101110
0110111001101100001011001110110010101111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000101000001001111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000101000001101111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000101000010001111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000011000110001111110<, sending to other clients.
Physical Layer Server: received frame from client 0:
>0111111000000001000000000101001101110100011000010111010001101001011011110110111000100000001100010010000001110100011011110010000001010
000011100100110100101101101100001011100100111100100111010001000000100110101100101011100110111001101100010110010100100000011
00110011011110111110<, sending to other clients.
Physical Layer Server: received frame from client 0:
>0111111000000001000100000111001000100000011101000110010101110011011101000110100101101110011001110010000001100100011000010111010001100
0010010000001110100011100100110000101101110011100110110011001100101011100100010111000100000000110100001010010101000110100001100101001
000000100100001111110<, sending to other clients.
Physical Layer Server: received frame from client 0:
>011111100000000100100000010001000100110001000011001000000111000001110010011011110111011010001101111011011000010000001101
00101110011001000000110010001100101011100110110100101100111011011100110010101100100001000000111010001101111001000000111001101110101011
100000111000001111110<, sending to other clients.
Physical Layer Server: received frame from client 0:
>011111100000000100110000011011110111001001110100001000000110001101101111011011010101101101011101011001101000110110000101110
100011010010110111101101110001000000110111011011001100101011100100010000000001101000010100110000100100000011100000110100001111001011
100110110100101111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000011000000101111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000011000001001111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >0111111000000001100000000110111110<, sending to other clients.
Physical Layer Server: received frame from client 2: >01111110000000011000010001111110<, sending to other clients.
Physical Layer Server: received frame from client 0:
>0111111000000001010000000110001101100001011011000010000001101100011010010110111001101001000000110001001100101011101000111011101100
101011001010110111000100000011100000110100001111001011100110110100101100011011000110110001111001001000000110001101101111011
011100110111001111110<, sending to other clients.
Physical Layer Server: received frame from client 0:
>01111110000000010101000001100101011000110110000100000011011000110100101101110011010010000001100010011001010111010001110111011001110
011001100101011101110001000000111000001101000011110010111001101101001011000110110001111001001000000110001101101111011
011100110111001111110<, sending to other clients.

