

35-Mise en réseau UDP

introduction

Ce didacticiel fournit des informations sur l'utilisation d'un réseau pour le transfert de messages Max. Nous examinons les objets *udpsend* et *udpreceive*, qui permettent l'envoi et la réception de messages via le protocole de réseau UDP.

Il existe de nombreuses raisons pour implémenter la mise en réseau d'ordinateurs dans les patcheurs Max. Souvent, dans les grands projets, le travail à faire dépasse ce qu'une seule machine peut gérer. Dans ces cas, il peut être utile d'utiliser un réseau existant pour collecter les cycles de processeur des machines inutilisées ou pour envoyer des messages de synchronisation entre les machines exécutant des sous-sections du patch de performances complet. En outre, les protocoles réseau peuvent être utiles pour partager des informations entre ordinateurs utilisés par plusieurs artistes, même dans des endroits différents. Les messages *udpsend* et *udpreceive* intègrent le protocole réseau UDP standard dans une paire d'objets Max, offrant ainsi un moyen simple de transmettre des messages entre deux ordinateurs d'un réseau.

Pour ouvrir le patch du didacticiel, cliquez sur le bouton vert **Ouvrir didacticiel** dans le coin supérieur droit de la fenêtre de documentation.

Généralités sur UDP

Le protocole UDP est un protocole de communication léger et flexible utilisé pour les échanges entre applications sur Internet. Il ne nécessite pas le maintien d'une connexion entre deux machines; l'application réceptrice surveille un port spécifique pour les paquets de données (appelés *datagrammes*) et les traite comme des messages entrants. Le programme émetteur doit spécifier *l'adresse* et le port de destination du récepteur.

Le protocole UDP n'est pas le protocole le plus robuste: il ne garantit pas que les paquets arriveront dans un ordre quelconque, et ne fournit aucun moyen de notification d'erreur ou de perte de paquets. Cependant, il s'agit d'un protocole à haut débit, souvent utilisé pour le contenu audio et vidéo en réseau, et il est facile de le contraindre à travailler avec des messages Max.

Envoi de messages de tous types

Jetez un coup d'œil à notre tutoriel. À gauche, un objet *udpsend* avec plusieurs types de messages Max différents connectés. C'est le côté «émission» de la connexion réseau: l'objet *udpsend* prend les messages Max, les transforme en datagrammes UDP et les envoie à la destination spécifiée par l'adresse TCP / IP et le port de la machine (les arguments de l'objet). Le patcheur nommé «network_receiver» est, comme son nom l'indique, le récepteur de ce patch. Si vous ouvrez le subpatcher, vous verrez un objet *udpreceive* qui surveille le port **7000** (il n'est pas nécessaire de spécifier une adresse réseau - l'objet suppose que vous surveillez le port sur «cet ordinateur»).

Lorsque vous cliquez sur ou modifiez l'un des messages connectés à l'objet *udpsend*, vous verrez qu'il apparaît dans le subpatcher. Les messages simples et complexes peuvent être envoyés par l'objet *udpsend*; le *multislider* envoie une liste de 12 entrées (précédé du mot **multislider** afin que nous puissions l'acheminer à l'extrémité réceptrice), mais même le message le plus simple peut être empaqueté et envoyé sur le réseau.

Nous devrions rapidement discuter de l'adresse réseau utilisée dans cet exemple. Etant donné que nous voulons tester ce système localement sur notre propre ordinateur, nous devons utiliser une

adresse réseau qui représente de manière cohérente l'ordinateur hôte. Cette adresse est appelée adresse de bouclage IP: **127.0.0.1**. Chaque fois que cette adresse réseau est utilisée, cela signifie «acheminer ceci en interne plutôt que via un réseau externe». Si vous souhaitez tester ce patch sur une paire de machines en réseau, l'objet *udpsend* devrait être modifié pour utiliser l'adresse TCP / IP de la machine destinataire.

Un programme de discussion de base

Le deuxième segment de notre patch de tutoriel est un exemple de programme de discussion simple. Là encore, un subpatcher est utilisé pour afficher les communications en action, bien que vous puissiez faire quelques changements simples pour que cela fonctionne entre deux ordinateurs - vous n'auriez qu'à changer l'adresse réseau de «localhost» (le nom symbolique de **127.0.0.1**) aux adresses appropriées des ordinateurs connectés. Cette partie du patch envoie du texte (entré dans un objet *textedit*) sur le port **7003** et reçoit des messages sur le port **7002**.

Contrairement à notre premier exemple, cela montre la communication bidirectionnelle en action. Lors de l'envoi d'informations entre deux ordinateurs, chacun a besoin d'un objet *udpsend* et d'un objet *udpreceive*, *udpsend* pointant vers l'adresse réseau de l'autre système. De plus, par convention, les communications bidirectionnelles se font par deux ports adjacents: **7002** et **7003**, dans ce cas. Si vous ouvrez le patcheur **chat_buddy**, vous constaterez qu'il communique également à l'aide de *localhost*, mais l'ordre des ports est inversé. Le texte est envoyé sur le port **7002** et reçu sur le port **7003**. Vous pouvez maintenant taper du texte dans l'un des patchs de discussion (dans l'objet blanc *textedit*). Lorsque vous appuyez sur la touche retour, il apparaît dans la zone grise de l'autre patch.

Résumé

La manipulation Max des messages sur le réseau utilise les objets *udpsend* et *udpreceive*, tirant parti du protocole simple et léger UDP pour transmettre de manière flexible tout type de message ou de liste. Au sein du même système, l'adresse réseau localhost peut être utilisée, mais le même mécanisme vous permettra d'envoyer des messages à travers le réseau à toute autre machine joignable.