



Mini-Howto für Fli4L und BattleCom hinter Router (für v. ab 1.5.x und v. 2.0.x) - update

Problem

Portforwarding (für beide Möglichkeiten)

Möglichkeit 1: BC-Server und BC-Client auf EINEM Rechner

Möglichkeit 2: BC-Server auf einem eigenen Rechner, BC-Client auf dem Spielrechner

Problem:

Viele Online - Spieler nutzen das Freeware-Programm BattleCom ("BC"), um sich im Spiel über Kopfhörer und Mikrophon miteinander zu unterhalten. Dazu hostet einer eine Session und alle anderen können unter seiner (internetgültigen) IP dazu stossen. Wer bislang einen Router besass, konnte mit dem im Beispielskript angegebenen Portforwarding ebenfalls joinen - nur, was tun, wenn alle einen Router besitzen? Vor diesem Problem standen wir nun...

Der erste Punkt ist, dass man mit dem BattleCom-Client keine Session hinter einem Router starten kann. Nur die Serverversion kann dies. Und - entgegen der Aussage des alten Howtos - kann man auch auf dem gleichen Rechner per BC-Client connecten (thnx für den Hinweis eines anderen Fli4l-Users!!). Somit ergeben sich zwei Möglichkeiten, eine BC-Session hinter einem Router zu hosten, eine einfache und eine stabilere (wer kennt nicht die unvermittelten Abstürze eines Win*-Rechners, wäre doch schade, wenn dann alle disconnected werden....). Getestet wurde dies auf Rechnern unter Win2k. Ich gehe davon aus, dass es aber mit allen directX-lastigen Win*-Versionen funktioniert.

#####

Für beide Möglichkeiten gelten folgende Routereinstellungen:

Portforwarding:

v. 2.0.x (base.txt)

```
#battlecom
PORTFW_1='2300-2400 192.168.xxx.xxx tcp' # auf die IP des BC-Servers!
PORTFW_2='2300-2400 192.168.xxx.xxx udp' #
PORTFW_3='47624-47625 192.168.xxx.xxx tcp' #
PORTFW_4='47624-47625 192.168.xxx.xxx udp' #
PORTFW_5='28800-28900 192.168.xxx.xxx udp' #
```

oder

v. 2.0.x und v. ab 1.5.x (portfw.sh)

```
#-----
# example for Battlecom (using ipautofw):
#-----
/usr/sbin/ipmasqadm autofw -A -v -r udp 2300 2400 -h $hiddenhost1 # dann hiddenhost1 auf die IP des BC-Servers setzen
/usr/sbin/ipmasqadm autofw -A -v -r tcp 2300 2400 -h $hiddenhost1
/usr/sbin/ipmasqadm autofw -A -v -r tcp 47624 47625 -h $hiddenhost1
/usr/sbin/ipmasqadm autofw -A -v -r udp 47624 47625 -h $hiddenhost1
/usr/sbin/ipmasqadm autofw -A -v -r udp 28800 28900 -h $hiddenhost1
```

ACHTUNG: Durch einen Fehler im Beispielscript kommt keine Verbindung zustande, da dort ein Port vergessen wurde! Statt "47624 47624" muss immer "47624 47625" bzw. "47624-47625" eingegeben werden - die oben angegebenen Ports sind richtig und funktionieren - dies gilt auch für Clients, die hinter (FLI4L) Routern sitzen! *schwirr*

```
#####
```

Möglichkeit 1: BC-Server und BC-Client auf EINEM Rechner

- Starte den BC-Server auf Deinem Win*-Rechner
- Starte danach den BC-Client und gib unter "Join Session" unter "Server Adress" "localhost" an - damit verbindest Du Dich mit dem Server auf Deinem eigenen Rechner - klar.
- Teile Deinen Freunden die IP-Adresse (voilà, Imonc....) Deines Routers mit
- diese können nun problemlos joinen

Pro/Contra:

- gut ist, man benötigt nur einen Rechner, um hinter einem Router BC zu hosten
- schlecht ist, bei einem reboot werden alle disconnected.
- gut allerdings, dass nach dem reboot alle wieder unter der gleichen IP joinen können (sofern der Router nicht neu ausgewählt hat...), ein Vorteil gegenüber den reinen Windowshostern, die ja nach dem reboot mit Neueinwahl eine neue, dynamische IP-Adresse zugewiesen bekommen.

```
#####
```

Möglichkeit 2: BC-Server auf einem eigenen Rechner, BC-Client auf dem Spielrechner

- Dazu benötigen wir nun einen zweiten Rechner (*huch, da steht ja schon einer*), auf dem wir den BC-Server starten.
- Es wird ein Portforwarding (**ACHTUNG**, s.o., unbedingt **LESEN!**) vom Router auf die lokale Rechner-IP des BC-Servers eingerichtet.
- Danach startet man - wie gewohnt - den BC-Client auf dem Spielrechner und verbindet sich mit der lokalen IP des BC-Servers
- Per IMONC schnappt man sich die aktuelle, ausgehende IP des Routers und teilt sie den anderen mit (wie bei

Möglichkeit 1 auch)

- Diese verbinden sich dann - wie gewohnt - mit ihrem BC-Client durch den Router hindurch auf den BC-Server.

wichtig:

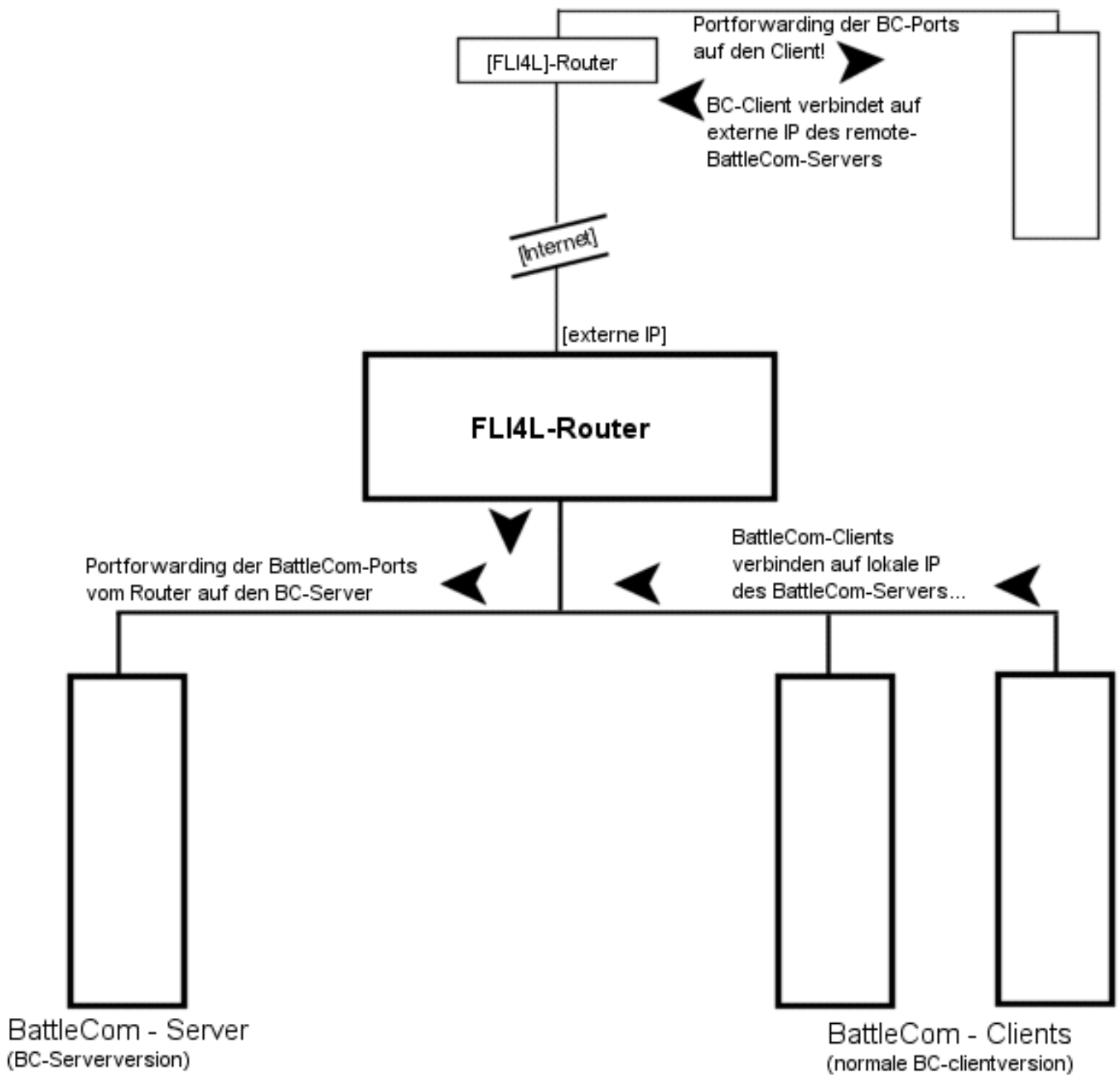
- Sitzen die anderen auch hinter einem Router, so muss ebenfalls ein Portforwarding von ihrem(!) Router auf ihren(!) Spielrechner, der den BC-Client betreibt eingerichtet sein! - logo.
- Am Besten wählt man einen aus, der bei sich den BC-Server startet, sonst muss man immer wieder das Portforwarding ändern - in unserem Fall betreibe ich den Server standardmässig, alle anderen benutzen ihren Client. Nachteil ist, man ist nicht mehr so flexibel - wenn ich schlafen will, muss der Server aus...

Pro/Contra:

- Vorteil ist, dass die BattleCom-Verbindung aufrecht erhalten bleibt, auch wenn ich meinen Spielrechner neu starten muss, niemand wird disconnected...
- Nachteil ist, dass man natürlich einen zweiten Rechner dafür abstellen muss:
- Weiterhin muss man, wenn man wieder selbst nur als Client eine andere Session im Internet joinen will, die portfw auf den Spielrechner zurückstellen muss.
- Vorteil ist allerdings bei wichtigen "Ereignissen" (Clanwar), dass niemand der anderen sein Spiel neu starten muss, weil durch den Absturz des BC sich auch z.B. Counterstrike direkt mit verabschiedet. Es fehlt dann max. 1 Spieler.

Schema

Anbei noch ein Schema für den Aufbau des Netzes - ich hoffe, es ist verständlich ;)



Schema eines Netzes mit BC-Server (und einem externen Client) hinter Router

So denn, nice fraggin'

06.03.02, 17.06.02, [Lars-Hendrik Schneider](#)

[Klicken Sie hier, um die Seite auszudrucken.](#)