eng.bce_connected_servers

Alexandr Kirilov (https://github.com/alexandrkirilov)

Blockchain example. Connected servers, client-server implementation.

This example based on what has been done almost 10 year ago on the principle "just for fun". This IT-joke is most illustrious example for explaining the blockchain implementation into server-server or client-server interaction or data exchange.

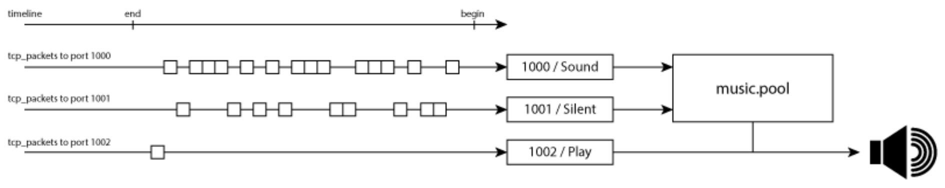
Technically in this example nothing complex or new:

- the server within installed FreeBSD at least 8.0 version (this version number is illustrating the fact that it's not new at all, this version been released 25 November 2009, you might to check it by your own in Wikipedia)
- installed package knockd
- installed package beep
- any client that is allowing you to send packets (UDP/TCP) via network, in this example used the client that is part of knockd

The little diversion into knockd and to what is it capable of - this is the daemon (the service) that is allowing to define any actions to TCP/UDP packets sequence that caught on network interface. At present time this daemon often used by System Administrators for improving security in case of servers maintenance.

The main reason of using beep - obvious, playing sound or making noice, whatever you like.

Now about what has been done "just for fun".



To be briefly - the sequence of TCP packet transforming to the file based on which compiling the parameter's string for the beep run command and playing music or making noise via system beeper. In our case it was music. With help of friendly musician, it was the "Imperial Marsh" from the "Star War" movie.

How it works?

For the begin been defined the time value (it was 0.1 second) for the active state of beeper. The was only two states: "Beep" - mean playing sound, "Silent" - nothing playing. After it assigned ports (look on image above):

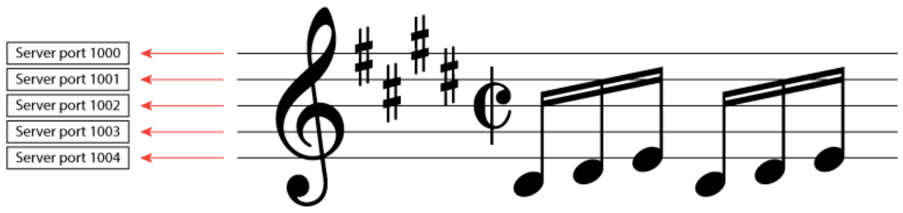
- 1000 - play sound 0.1 second, when TCP packed appeared on this port server adding line to the end of file music.pool that mean for beep package "play sound 0.1 sec"
- 1001 - be silent 0.1 second, when TCP packed appeared on this port server adding line to the end of file music.pool that mean "be silent 0.1 second"
- 1002 - when TCP packed appeared on this port that mean to start playing music via beep package with parameters generated from music.pool file

After installing all of configurations, client application been sending sequence of TCP packets to the ports 1000 and 1001 in special order and at the end of it sending one package to the 1002 port that been starting playing.

This solution might to play huge music compositions.

248
SONATE
Sonata quasi una Fantasia
Der Gräfin Julie Guicciardi gewidmet
L. van Beethoven, Op. 27 № 2
Adagio sostenuto
Si deve suonare tutto questo pezzo delicatissimamente e senza sordini
14
sempre pp e senza sordini

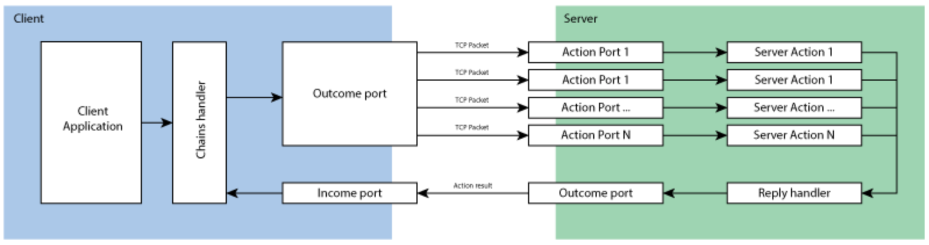
At first glance it might be appeared without any connection between servers interactions and music. But the thing that is making it similar - the language. Not the language like especially defined language (English, Chinese, Arabic, Russian, etc). The language like phenomena that is appeared like instrument that is used for interaction inside of group of objects.



Does it looks more understandable after image above?

In other words, the sequence of activation the sound source - is the "chain" of elements, where the elements is the states of sound source (on/off), where the "block" is the musical composition (sequence of states) that fixed by composer in following of his own idea and described by musical notations (protocol) for further reproducing it by other sources of any sounds by musicians (the objects that have known protocol and equipped by sound source) in precise that composer decided. If you change this sequence, for Beethoven's "Moonlight Sonata" became not the "Moonlight Sonata", it might whatever else.

How the all of it connected to server-to-server connection or server-client connection? By making chain of the packets blocked by idea to be fitted to the action of server application. You might to create own protocol that is based on blocked sequence of packets assigned to defined server application actions.



Most habitual approach one application - one port, and we are sending sequence of electric signals attached to this particular port. But! No one is stoping to use range of ports for one application.

The image above is illustrating logical schema of interaction between servers or clients and servers. When you are blocking chain of packets (where the elements of chain - TCP/UDP packets and the block - is the sequence) on a first level and after it on the second level you are blocking whole sequence by associating it to the action on the server. Looks like, in modern terms, blocked chains of blocked chains.

How it works: the client is sending the zero-length packet to the defined port, server is performing action and returning to the client result of action. If you need complex sequence of packets you might to define schema of the ports in conjunction to UTF-8 and transmit the string without sending string in general at all. The schema of mapping ports for improving security might be changed periodically and it will looks like dynamically generating the language for interaction.

In case of poor connection and high-load, especially for mobile developing this approach might be well enough solution for reducing traffic and improving security. At time of testing in some cases the traffic regression got 10 time less.

Follow author updates on **Linkedin**

Follow AR|BO|RE|US updates on **Twitter** and **Linkedin**