Detection of IP conflict in Linux

White paper by Jaroslav Imrich

Published: 01.12.2006

IP conflict occurs when two different devices in local network are using the same IP address. MS Windows informs about this event by showing the warning message and recording MAC address of conflicting system in system logs just like Solaris or OpenBSD. Situation in operating systems based on Linux kernel is completely different. If you use GNU/Linux and you have ever taken the part in IP conflict, you probably noticed only repeating losses of connectivity. You would really waste your time searching for the cause of these problems in system logs because when IP conflict occurs Linux kernel does absolutely nothing.

Detection of IP conflict

IP conflict can be detected only when your system receives packet that meets two requirements. Source IP address must be the same as yours and received packet must originate from different device. The origin of packet is determined from the source MAC address. MAC address is unique for every device in the network, so when your computer receives packet with the same source IP address as yours but with different source MAC address than yours, it is obvious that some other device on the local network uses your IP.

When your local network is based on switches (90% of networks do) your computer receives only packets that are directly addressed for it. Therefore it is much better to detect IP conflict from broadcast packets that are delivered to every device on local network. Most operating systems detect IP conflict from ARP broadcasts. Broadcasts are sent also by samba and other network services but these do not run on every computer in network, so it is always best to stick to ARP.

Losses of connectivity

ARP protocol is used to resolve MAC addresses of devices connected to local network and the losses of connectivity suffered during IP conflict are directly related to it.

Suppose following situation: Computer A with IP address 10.1.1.2 wants to communicate with computer B with IP address 10.1.1.3. Computer A sends ARP broadcast with question: "*Who has IP address 10.1.1.3? Answer to computer with IP address 10.1.1.2 and MAC address AA:AA:BB:BB:CC:CC*". This broadcast is received by every device on local network, but only computer B which is the

owner of corresponding IP address sends answer: "*10.1.1.3 belongs to me and my MAC address is 11:11:22:22:33:33*". Computer A temporarily stores the answer into its local ARP cache and is able to begin communication with the computer B. Persistence of normal entry in ARP cache is usually approximately one minute. It is very important to say that if your computer has an entry for any IP address in its ARP cache, it will not send ARP broadcast. It will use available data.

Limitations of this communication concept are clearly visible when there is a system with duplicate IP address on local network. It generates replies for ARP broadcasts sent by neighboring systems and it propagates wrong data into their ARP cache. They have wrong MAC address associated with your IP address and so they are unable to communicate with your computer. Use of static ARP entries can be sufficient prevention, but it is difficult to maintain static ARP entries in larger networks. In majority operating systems contents of the ARP cache can be shown and managed by the "arp" utility.

Preventing IP conflict

Naturally most modern operating systems try to avoid IP conflicts. During start-up or during the change of the network interface parameters they broadcast a special type of ARP request called "Gratuitous ARP". This way they try to find out if the IP address they are going to use is not already used by any other device on the local network.

Gratuitous ARP request generated by computer A from previous example looks like: "*Does anyone have IP address 10.1.1.2? Answer to computer with IP address 10.1.1.2 and MAC address AA:AA:BB:BB:CC:CC*". If computer A receives any response it will know that IP address 10.1.1.2 is already used by another device. In this situation most operating systems draw user attention by showing the warning message and they do not assign the IP address to the network interface.

Linux kernel **does not react** to Gratuitous ARP request. This can be considered a serious problem. Suppose following situation: Computer A running Linux uses IP address 10.1.1.2. Computer B is starting Windows XP configured to use the same IP address. Computer B broadcasts Gratuitous ARP request for this IP address and waits for reply. Linux running on computer A does not answer. Computer B starts to use this IP and causes the IP conflict. Computer A is unable to communicate with other network devices because every device on the local network updated corresponding entry in its ARP cache when it received Gratuitous ARP request from computer B. If computer A would respond to Gratuitous ARP request generated during startup of computer B, there would be no IP conflict.

Actually it would be possible to prevent 99% of IP conflicts just by sending reply to Gratuitous ARP immediately. Broadcasting of next Gratuitous ARP request would ensure that all neighboring devices update entries in their ARP caches with the correct values.

IP conflicts and Linux

It was already mentioned that Linux kernel currently does not contain any code

that would deal with detection of IP conflict and generation of replies for Gratuitous ARP requests. You can think of this as of the bug but at the other hand this "functionality" is used for example by High-Availability Linux Project. Absence of IP conflict detection was already solved before by unofficial kernel "ARP patch" created by Marc Merlin. It implements logging of this event through syslog interface, but it was rejected by the developers of Linux kernel with an argument that if this can be done in userspace it should not be in the kernel. I agree with Mr. Merlin and I also think logging of this event (and also the reaction to the Gratuitous ARP) should be done by the kernel. This functionality could be disabled in cases when needed for example by "/proc interface" just like IP forwarding.

I started project IPwatchD – http://ipwatchd.sourceforge.net/ – with target to create tool that can detect IP conflicts and generate replies to Gratuitous ARP requests. Basically IPwatchD is quite simple daemon that uses pcap library to capture all incoming ARP packets. It then compares IP and MAC addresses from packets with addresses of local interfaces trying to detect IP conflict. IPwatchD can operate on each network interface in two modes – passive and active. In passive mode it just generates syslog events. In active mode it also answers Gratuitous ARP request and sends following Gratuitous ARP requests to update cache of neighboring hosts with correct data.