COVERT CHANNELS

Implement a one-way inter-VM communication scheme that makes use of covert channels.

Experimental setup

Two Xen virtual machines running Linux reside on computer A. Each VM has:

- one virtual CPU. The VMs share the same physical CPU
- a 4GB hard disk image
- 256 MB or RAM
- an ssh server

The VMs' clocks are synchronized.

An ncat server resides on computer B and listens for incoming TCP connections on port 1234. A and B are connected via an Ethernet link. B can be reached from the VMs. As such, the link between A and B is the only bottleneck, should the VMs communicate with the ncat server.

The VMs are not able to exchange packets with each other.

Your task

Your task is to implement a sender and a receiver that make use of a covert channel to communicate. Each of them runs on a separate VM. One of the following media could serve as a covert channel:

- CPU
- cache (it doesn't have to be as complicated as the solution proposed in "Hey, You, Get Off of My Cloud")
- network (use the ncat server)

We strongly recommend against using the HDD.

The two machines must not exchange any data via conventional means.

Your solution must be tolerant to noise. You are expected to implement something like frame control sequences [1]. FCS mismatches are reported by the receiver, prompting the sender to resend the frames. For the sake of simplicity you can use a simple start-stop protocol with fixed-length frames.

The sender should read the data that must be sent from a file; likewise, the receiver should write it to a file. The files' format should be either:

- a string of ASCII characters, each representing 8 bits or
- a string of '0' and '1' characters, each representing one bit.

It is ok if the received data is padded with zeros.

Reprouducibility

Because the hardware used to check your homework is very likely not to have the same specs as the one it was developed on, refrain from hardcoding anything related to performance (e.g. how many loops the CPU can execute per second).

Instead, write a program/script that measures these constants and writes them to a file. If the program/script needs to run in both VMs at the same time, write a script that runs in Dom0 and calls the former via SSH.

Don't make any assumptions regarding IPs; expect them as arguments to your programs/scripts (if applicable).

Submission

You should submit an archive containing the following:

- all programs/scripts
- a document explaining:
  - your chosen solution and the motivation behind it
  - how to build your program(s)
  - how to run it (in detail)

Grading

- 5p: simple implementation with no noise tolerance
- 5p: tolerance to noise
- you will get a bonus for going beyond the call of duty

[1] http://en.wikipedia.org/wiki/Frame_check_sequence