



Troubleshooting and Debugging Network Applications

Sasha Nedvedicky

solaris-rpe Prague



What's network application all about?

Network App. = Client + Net + Server

- we do not have all those parts under our control
- everything happens in time
- network has memory
- detailed logging is a MUST
- network is good servant, but evil master

Hey, your \$!#?* crap does not work!

- users always blame your application
- don't panic, ask questions
 - > what, does not work?
 - > when it's happened for the first time?
 - > how often it happens?
 - > what do they expect?
- do not expect precise and correct answers

Are you talking in English?

- mutual misunderstanding between developer and user
- non native English speakers in large companies
- always try to draw a picture
- think twice, ask once
- use the same language level as the weakest element in com. chain

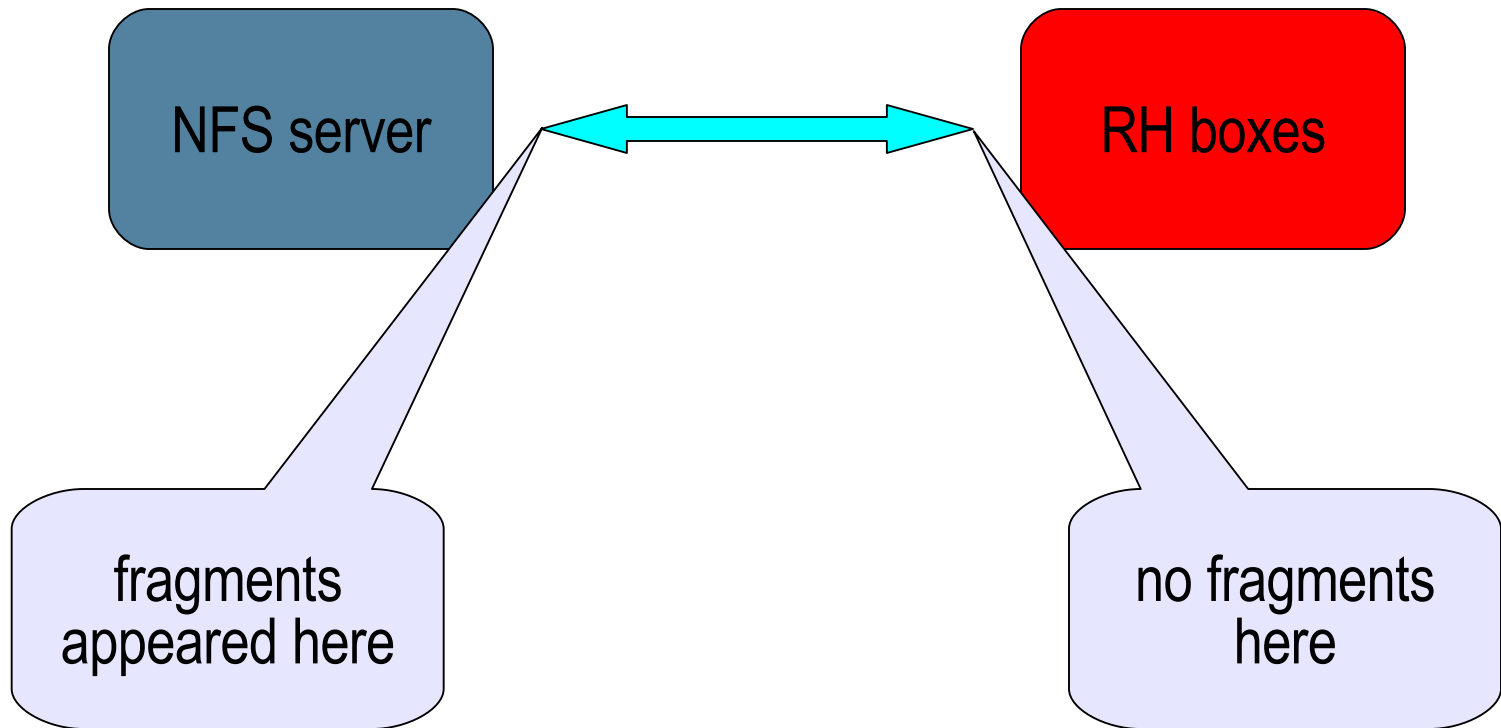
Fatal misunderstanding



Someone else's problem

- it's not a fault of your product (sometimes)
- real life example – best effort is highway to hell
 - > customer replaced some old RH servers with fedora
 - > suddenly Solaris 8 NFS server suffered by high CPU load

...continuation



- -> there must be some crappy router between NFS server and RH boxes

Lessons learned

- nothing was wrong in each node, it's rather misconfiguration exploited by mutual interaction of
 - > RH linux box
 - > router
 - > inefficient fragment handling on Solaris 8
- special/best effort is potentially harmful
- Jan Postel – be radical in what you send, be liberal in what you receive -> be radical always

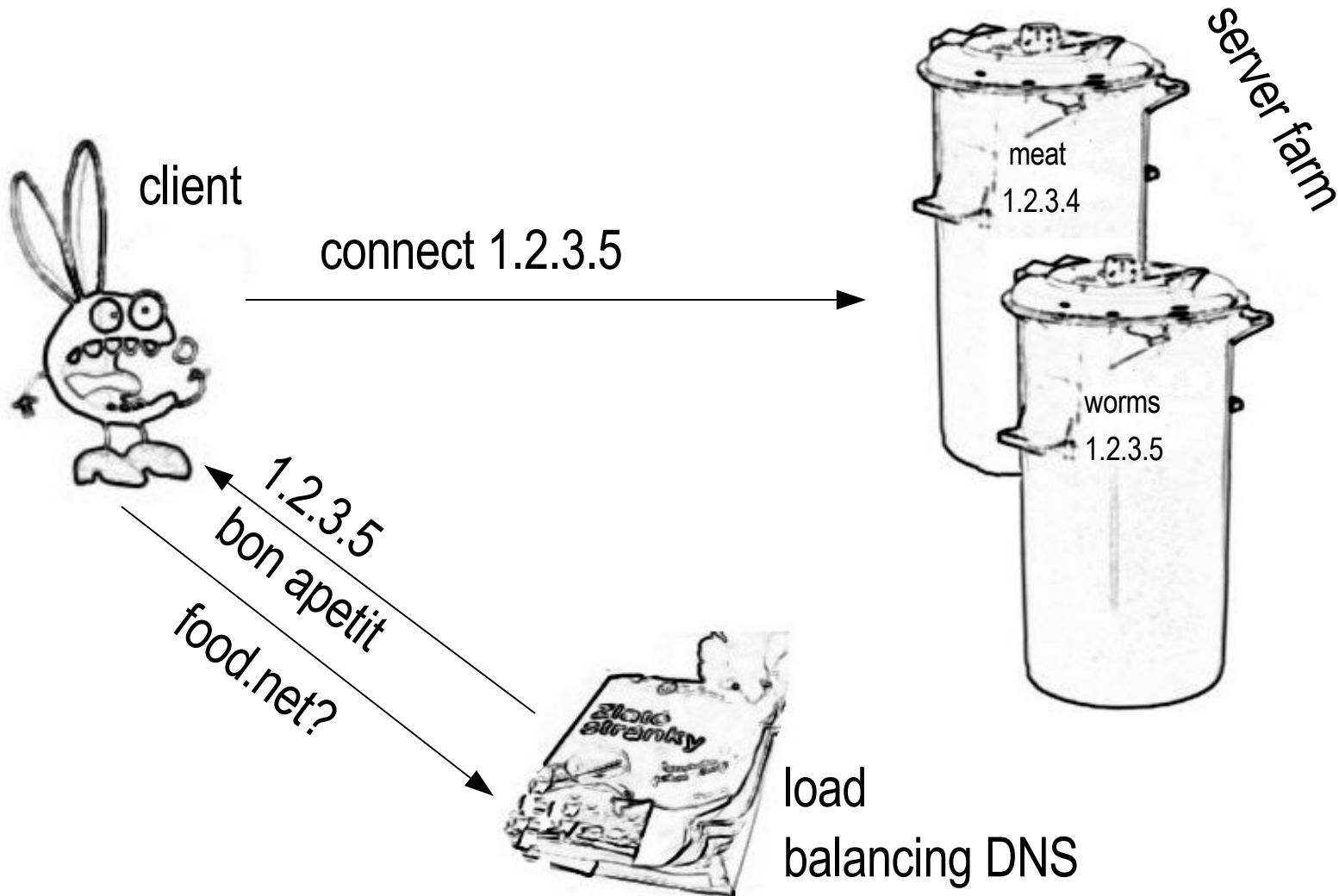
hmm... so there is a bug

- at this point we have to collect as much information as possible to reproduce bug in-house. we want to get:
 - > all possible logs (debug logs) we think might be useful
 - > packet dump
 - > in case server/client crashes/deadlocks, we also should be able to get coredtrace
 - > configuration
- it's all enough to try out to reproduce problem

Mind the Gap



Load balancing DNS



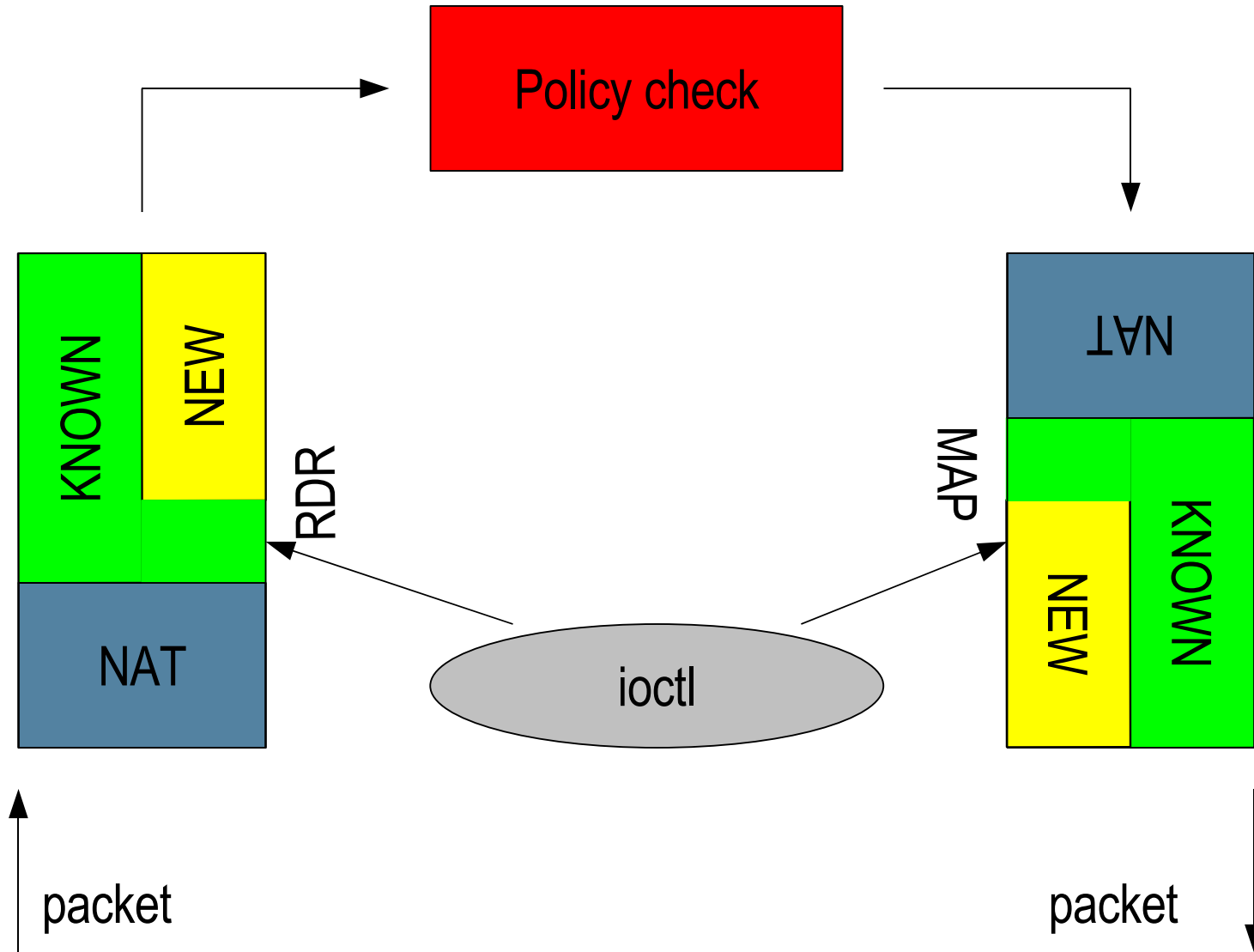
still bug free...

- it's obvious that DNS example is misconfiguration at server side
- chances we have bug free software are still high even if we are trying to reproduce problem
- do not think you know everything, there is always some surprise waiting for you

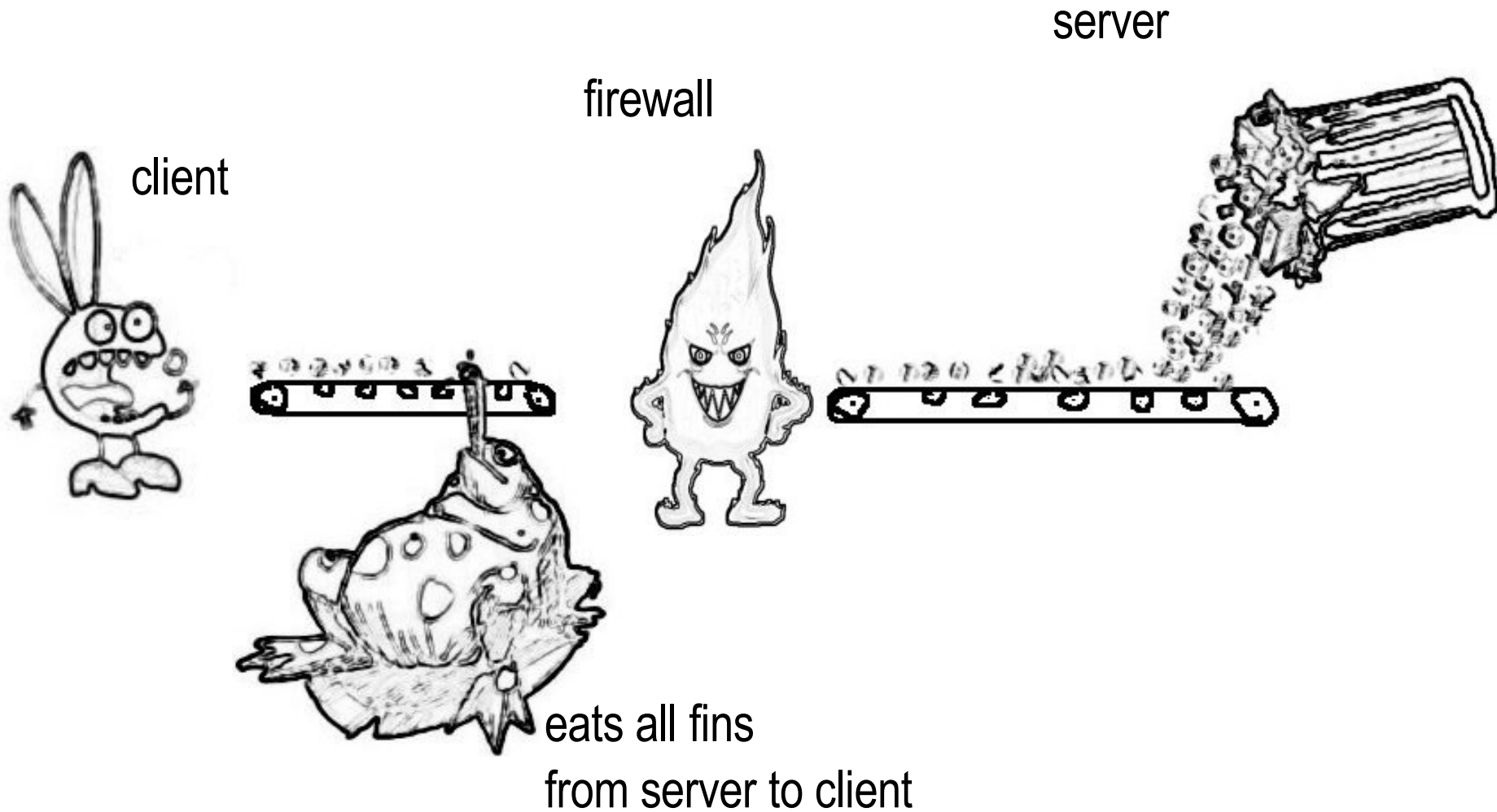
Be evil as much as user is

- Users usually push your application to its limits
- Network itself is very hostile environment
- Avoid using ideal conditions when testing, try to simulate the worst conditions you can imagine

Some packets are passed unNATed



Reproducing the problem



Tools

- sniffers/packet analyzers
 - > snoop, wireshark (ethereal), ...
- packet generators
 - > libnet, sendip, isic, fragrouter, tcp replay, ...
- swiss army knife
 - > netcat (nc), netsed, netgrep
- ssl tools
 - > stunnel, openssl
- use anything what can help you

NetCat as SMTP server and client

- in this example we are going to use two instances of netcat
- SMTP server listening at port 2525:
 - > nc -l -p 2525 < server.smtp
- SMTP client sending email to server
 - > nc localhost 2525 < client.smtp
- let's sent email to real daemon using netcat:
 - > nc localhost 25 < client.smtp

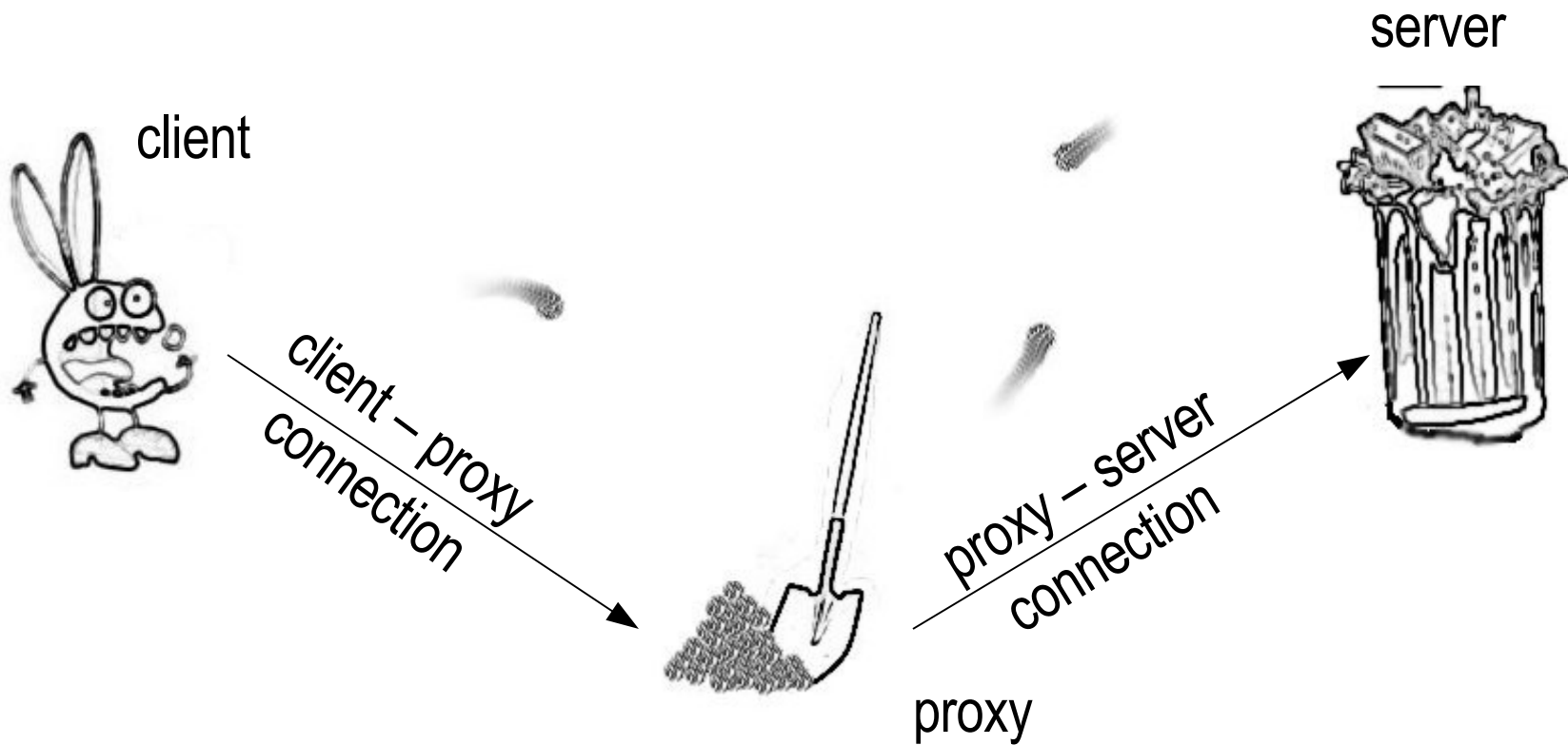
netgrep

- `ngrep -d eth1 -p '.*\www\.zcu\.cz.*' tcp port 80`

netsed

- `netsed tcp 2525 127.0.0.1 5050 s/nazdar/ahoj`

TCP Proxy



Using stunnel

- it is proxy providing SSL encryption to arbitrary TCP application
- there are two modes
 - > server mode (non SSL server)
 - > client mode (non SSL client)

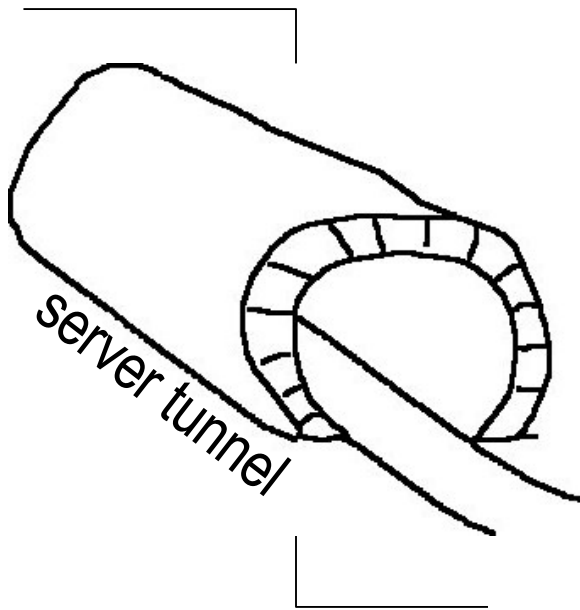
Building SSL app. proxy

- it is useful when dealing with interoperability issues
- customer asks: why your application is so slow when talking to server extra-fast-server? the application from extra-cost-vendor is much faster

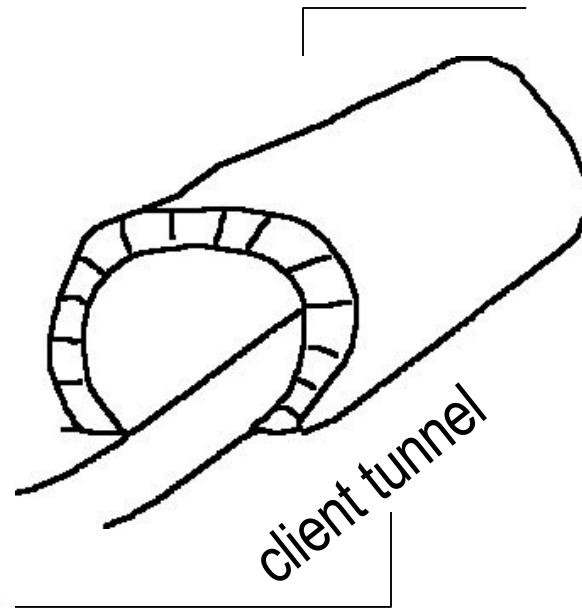
SSL proxy



ssl only
client



ssl only
server



proxy (plain only)

Mistakes

- misconfiguration
- false assumptions – every assumption must be verified
- you are seeing different logfiles and different packet dumps
- lack of luck



thank you for your time

Sasha Nedvedicky

