Flush function

Contents

1 Introduction **1**

2 API **1**

3 Exemple **1**

4 Test **2**

1 Introduction

What: Emplementing the "flush" function that provide a simple but complete example explaining how we capture packets.

Object located at:

- *libLDA/run/flush.o*
- *libLDA/run/dump*

2 API

In order to minimize the number of packets copies, we provide a callback function to the libpcap.

```
void Flush::callback(u_char *user, const struct pcap_pkthdr *h, const u_char *bytes)
{
  int pktLen = 0;
  static int npkts = 0;
  static CALDIF::PktAccessor_LDA_pkt ldaPkt;

  /* full access to API with run object */
  Flush *run = (Flush*)user;
  run->nbFlush++;
}

void Flush::flush(int msTimeOut)
{
  while (lda->dispatch(msTimeOut, Flush::callback, (u_char*)this) > 0);
}
```

3 Exemple

```
Flush::Flush(int argc, char * argv[])
  : Run(argc, argv) {}

void Flush::acquire()
{
  this->flush(1000);
  ::usleep(100000);
}
```

```
int main (int argc, char * argv[])
{
  Flush run(argc, argv);
  run.configure();

  /* run acquire as thread */
  run.nbFlush = 0;
  run.start();

  /* slow control */
  while(true) {
    printf("... %i packets flushed.\n", run.nbFlush);
    run.nbFlush = 0;
    sleep(1);
  }

  run.stop();
  return 0;
}
```

4 Test

```
# run/dump poldhcp54.conf -f
...
Flush(    5):      0%      1      1      2      1
Flush: [buff] [ 1,5] [ 1,8] [ lda] [ dcc]
Flush(    0):      0%      0      0      0      0

*** Total : miss 0 / 5 Pkts = 0%
 DIF [ 1,5] : 1 received + 0 missing 1 pkts and 0 errors.
 DIF [ 1,8] : 1 received + 0 missing 1 pkts and 0 errors.
 DIF [ lda] : 2 received + 0 missing 2 pkts and 0 errors.
 DIF [ dcc] : 1 received + 0 missing 1 pkts and 0 errors.
 Dropped pkts:    0
 Unexpected pkts: 0
```