DiCAP - An Architecture for Distributed Packet Capturing

Cristian Morariu, Burkhard Stiller

Department of Informatics IFI, Communication Systems Group CSG, University of Zürich

Motivation
Design and Implementation
Evaluation
Concluding Remarks

© 2009 UZH, IFI
Overview

- Most network monitoring tasks require packet inspection

- Network monitoring
  - Live
    - packets are inspected in real-time
    - data is dropped after inspection
    - e.g. traffic accounting, IDS

- High packet rates reduce the time available to handle a single packet

- Solutions for high packet rates:
  - packet sampling
    - decreased measurement accuracy
  - dedicated hardware
    - Expensive
Traditional Architecture for Traffic Analysis

- Simple
- Easy to deploy
- Not scalable due to a single analysis box

- More complex
- More scalable with respect to performance
- The single point of failure still present
Motivation

- Build a distributed architecture for IP traffic capturing that
  - Avoids a single point of failure
  - Avoids dedicated hardware
  - Based on off-the-shelf, inexpensive PCs
  - Allows the increase of packets that can be captured
DiCAP in Distribution Mode

Mirroring Device

Node Coordinator

Capture Nodes

C_1  libpcap

C_2  libpcap

...  ...

C_n  libpcap
Selection Policies (1)

- Round robin selection
  - Node coordinator introduces control packets in the mirrored traffic
  - Capture nodes are logically organized in a chain
  - After each packet, capture responsibility shifts to the next node in the chain
  - Node coordinator configures the chain
Selection Policies (2)

- Hash-based selection
  - A hash function is applied on packet headers
    - Current implementation uses IP identification field
  - Each node is responsible with a particular range of hash values
  - Each packet is captured by the node responsible with the respective range of hash values

- Advantages:
  - Easier synchronization
  - No need of control packet injection

- Disadvantages:
  - More computation needed for hash calculation
DiCAP Implementation – Distribution Mode

- Linux kernel
- NIC Driver
- NIC

Management Unit

Packet processor

Packet data forwarder

DiCAP Module
Evaluation Testbed

- Two nodes used for traffic injection using Linux pktgen
- One libpcap node for testing
- Up to 4 DiCAP nodes in distribution mode
3 different tests were performed

1. traditional libpcap on a single PC

2. DiCAP in distribution mode on 2 PCs

3. DiCAP in distribution mode on 4 PCs

DiCAP can improve the capture performance up to 700% when 4 PCs are used in parallel.
Concluding Remarks

- DiCAP can be used to distributedly capture packets on a high-speed link.
- It may be used to allow a distributed deployment of libpcap-based applications running on common, inexpensive PCs.
- It may significantly improve libpcap performance.
- Very simple design, easy to implement in hardware.
- Further investigation on using other hash functions may lead to performance improvement and better load balance.