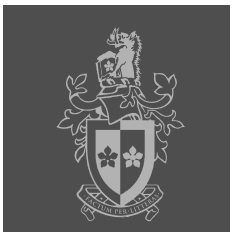
SWINBURNE

CENTRE FOR ADVANCED INTERNET ARCHITECTURES

Network Traffic based Application Identification Demonstrator (Net – AI) GUI

Kenny Nguyen
4150864@swin.edu.au

Supervisor: Sebastian Zander
CAIA INTERNSHIP



Outline

- Motivation
- Design
- Implementation
- Demonstration
- Conclusion and Future work

Motivation

- Involve with CAIA (research community, great place to learn)
- Passion for programming
- Algorithm
- GUI application

Why Packet Classification

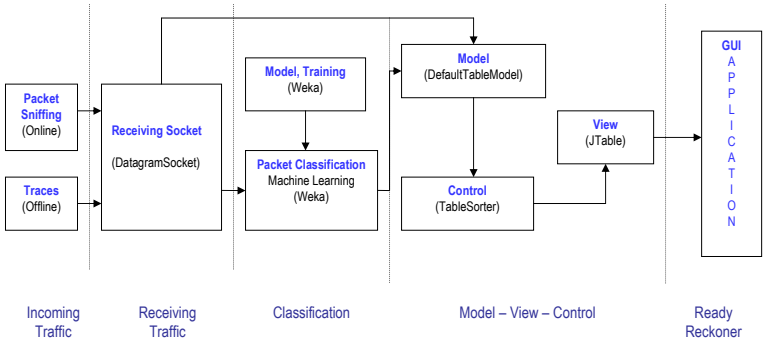
- Simple: Fair Dinkum Business
- No longer depends on:
  - Port Identification
  - Stateful Reconstruction
  - Signature-based method
- Alternative (Proposal): Machine Learning Algorithm
  - Identify end-host application
  - Priority Traffic Queuing
  - Meet the demand of QOS requirement
  - Detection, Security, Surveillance
  - Traverse efficiently, problem-free, with high performance

Design

- Use Model – View – Controller.
- Object - Oriented Design (Modularity)
- Heuristics “Rules of thumb”:
  - Structure of individual classes and objects
  - Collaboration, containment, association
  - Inheritance, including multi inheritance

→Then easy to accommodate new requirements.

Design



Implementation (aka Coding)

- Java (Platform Independent, wide spectrum)
- Java Version 1.5
- Java AWT and Swing packages
- Thread, HashMap, StringTokenizer.

Implementation

```
try
{
        BufferedReader reader = new BufferedReader(new FileReader("license.txt"));
        JTextArea text = new JTextArea(8, 50);
        String line;
        while ((line = reader.readLine()) != null)
        {
                text.append(line);
                text.append("\n");
        }
        text.setCaretPosition(0);
        text.setFont(new Font("Dialog", Font.PLAIN, 12));
        text.setEditable(false);
        text.setForeground(new Color(0,0,225));
        JOptionPane.showMessageDialog(
                pane,
                new JScrollPane(text),
                "Licensing: Please email Grenville Armitage <garmitage@swin.edu.au> ",
                JOptionPane.INFORMATION_MESSAGE,
                 new ImageIcon("swin.jpg"));
}
```

Implementation

```
try
{
  File f = new File("port.txt");
  FileInputStream fis = new FileInputStream(f);
  BufferedInputStream bis = new BufferedInputStream(fis);
  dis = new BufferedReader(new InputStreamReader(bis));

  // read the first record of the database
  while ( (dbRecord = dis.readLine()) != null)
  {
     StringTokenizer st = new StringTokenizer(dbRecord);
     key = st.nextToken();
     value = "";
     while (st.hasMoreTokens())
     {
       value = value + " " + st.nextToken();
     }
     portNumber.put(key,value);
  }
}
```


Implementation

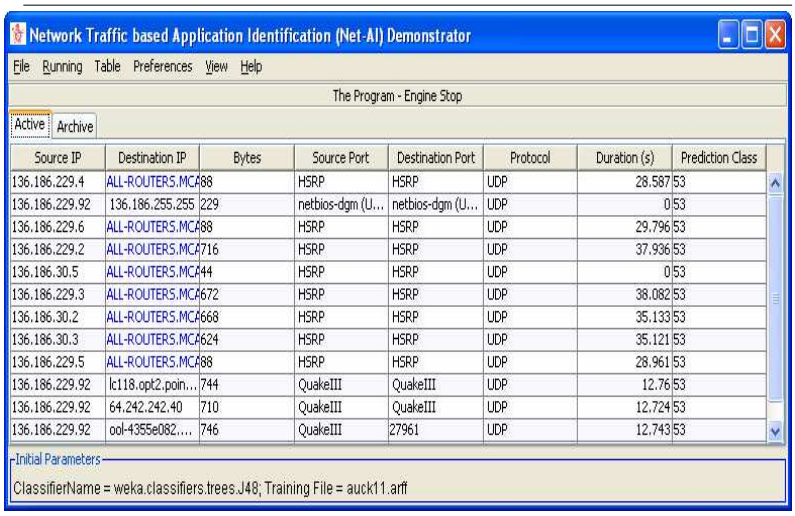
```
public int compare(Object o1, Object o2)
{
        String s1 = (String) o1;
        String s2 = (String) o2;

        int index = s1.indexOf(" ");
        String s1Prefix = s1.substring((index + 1));
        double s1Number = Double.parseDouble(s1.substring(0, index));

        index = s2.indexOf(" ");
        String s2Prefix = s2.substring((index + 1));
        double s2Number = Double.parseDouble(s2.substring(0, index));

        if (!s1Prefix.equals(s2Prefix))
        {
                return s1Prefix.compareTo(s2Prefix);
        }
        else
        {
                 return s1Number > s2Number ? 1 : s1Number < s2Number ? -1 : 0;
        }
}
```

Demonstration



Conclusion

- Running and testing under java 1.5 successfully.
- Easy to use GUI application
- Monitor & Classify network traffic
- Separate active and completed traffic
- Learning from Project
  - ☐ Strengthen programming, research skills
  - ☐ Day to day professionalism

Future work

- Save file in different format (CSV, XML, Excel Style Sheet, etc).
- Open traffic capture (flow statistics) file into table to analyse
- Create new Panel to analyse network traffic and provide statistic.
- Implement configuration dialog
- Implement filter function

We're there

Thank you for listening

Questions