HELSINKI UNIVERSITY OF TECHNOLOGY

Policies – what they are and how they might be created?

Some general notes and a case study

Lecture for S-38.180 QoS in the Internet

10.10.2002 Mika Ilvesmäki

Networking laboratory

Contents

- Policy system framework and terminology
- Users or network
  - who decides about policy
- Classification
  - what info binds the packet to the policy?
- What to measure in a network to characterize applications?
- Flow analysis
- Case: Measurement based policy creation

Traffic management

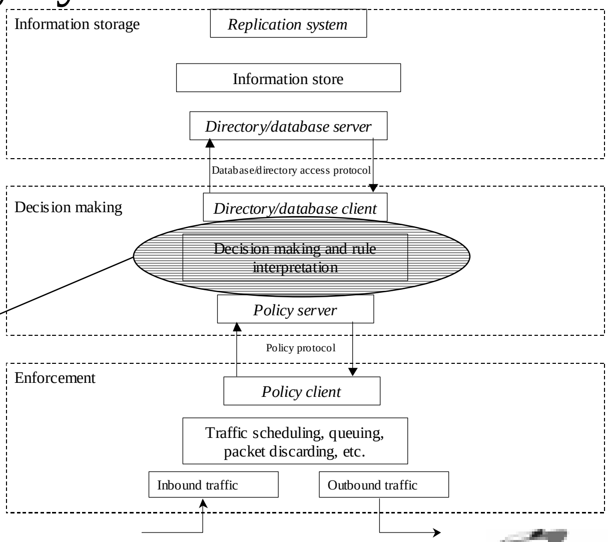
- TM systems consist of a set of high-level rules that are propagated out to enforcement points using a policy system
  - Policy must be enforced to ensure that the users are behaving properly
- Network should classify, handle, police and monitor the traffic

Terminology (RFC 3198)

- Policy is either:
  - A definite goal, course or method of action to guide and determine present and future decisions. "Policies" are implemented or executed within a particular context (such as policies defined within a business unit).
  - a set of rules to administer, manage, and control access to network resources [RFC3060].
- Policies are built with policy rules
  - Policy rule is a basic building block of a policy-based system. It is the binding of a set of actions to a set of conditions - where the conditions are evaluated to determine whether the actions are performed [RFC3060].
- Policy condition is usually a filter
  - A set of terms and/or criteria used for the purpose of separating or categorizing. This is accomplished via single- or multi-field matching of traffic header and/or payload data. "Filters" are often manipulated and used in network operation and policy. For example, packet filters specify the criteria for matching a pattern (for example, IP or 802 criteria) to distinguish separable classes of traffic.

Policy system structure

- Policy systems as such are pretty straightforward
  - Policy clients at routers ask the policy parameters from the policy server
  - Policy servers get the policy data from the information store
- Key question rarely given thought: How do you *create* the policy rules and the corresponding actions?
  - Static choices
  - Guesses
  - Dynamically
    - based on what?



HELSINKI UNIVERSITY OF TECHNOLOGY
Mika Ilvesmäki, Lic.Sc. (Tech.)

Traffic classes

- Based on experience and scalability studies the easiest way to bring service differentiation into the Internet is to use a limited amount of traffic classes (DiffServ).
  - But how many? 2, 3, 8 or more?
- Different traffic classes represent different priority levels
  - The problem is still: How do you know what packets go to which classes?

User decisions

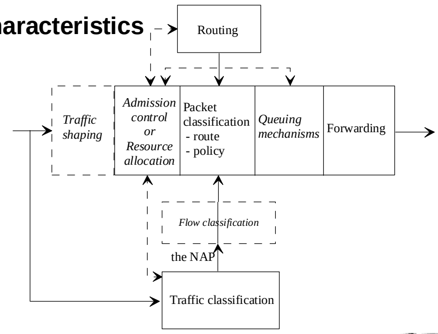
- Users may inform the network on the service level (class) of the packet.
  - resource restrictions -> admission control
  - malicious users may want to misuse the network capacity
  - users want to measure the service level they get -> added complexity/software/traffic
  - and... do all the users _really_ have the expertise to make the decisions?!
- Users should be required to provide only minimum of information on the traffic characteristics!

Network decisions

- Network determines the service level (class) of the packet
  - feedback from the use of resources
    - SLAs do not promise anything absolute in terms of network service
  - AAA (Authentication, Accounting and Administration) guarantees the service levels to appropriate users
- If network decides individual packet treatment it should know what kind of packet it is classifying
  - This requires knowing the application characteristics
    - by examining the packet headers and/or content
    - by information obtained from other network devices that know the packet's type

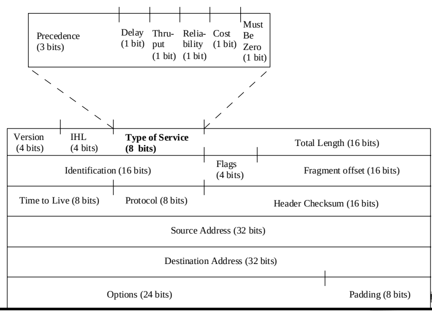
Measurement based policy creation

- Policy creation supports a QoS capable network
  - It co-exists with other functional blocks in the packet path and its basic task is to:
    provide info on application characteristics



What to measure?

- The basic data block in the Internet is the IP packet
  - packets are made of bytes that are made of bits.
    - no info on the overlaying application
- IP packet identifies the overlaying protocol
  - TCP or UDP as far as user apps are concerned
    - TCP/UDP port numbers identify, to some degree, the application used



Where's the info on the packet contents?

- Packet header information
  - layers 1 and 2 do not contain any information on packet content
  - layer 3 (IP) identifies the sending source and receiving destination the upper layer 4 protocol (TCP/UDP)
    - oversimplification: who sends packets where
  - layer 4 (UDP/TCP) identifies the port numbers used at source and destination
    - oversimplification: what application is used
    - source identifies the application that originates the packet and the destination tells us where the packets are headed
- Layers 3 and 4 are the first ones that contain any information on the application that the user is using to create packets in the network.
  - Aim is to limit the processing on packet so let's settle for using the layer 4 info at the highest.

Design guideline #1

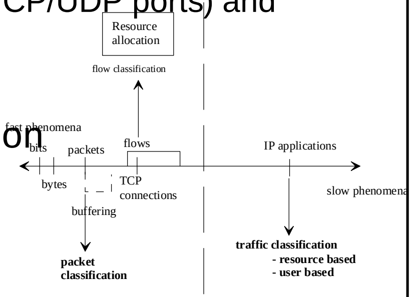
- Do not associate port numbers to QoS classes (-> potentially 65535 classes)
- Analyze traffic, get port number lists and bind the contents of the list to DiffServ Codepoints (DSCP), for instance.
  - Port number have nothing to do with QoS identification whereas DSCP is designed just for that

The story so far...

- Network decides the packet class
- Classification is done (with filters) based on information on the IP and TCP/UDP –level
  - What port numbers do we assign to what traffic classes?
    - Manual choices and configuration
      - educated guesses on where flows of different nature are located in the port-space
    - ... or maybe you could measure the network and decide the port numbers based on an analysis of the measurements
    - measurements should be done on packet level (and concentrate on packet header information)

Packet aggregation

- Packets aggregate into
  - TCP connections or
  - flows, governed by the fivetuple (proto, source and destination IP addresses and TCP/UDP ports) and the timeout
- Using the concept of flow we tend to get more information on the use of applications



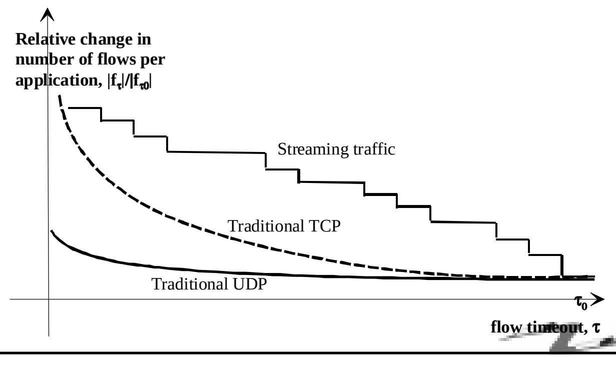
What is flow analysis - I

- Flow is defined by the aggregation level (5-tuple) and flow timeout
- Simplified: Divide packets into flows, observe the number of packets within a flow and the flow count
  - Possibly aggregate according to TCP/UDP Sport/Dport
- Assumption: Different types of applications may have different behavior in the packet/flow –space

HELSINKI UNIVERSITY OF TECHNOLOGY
Mika Ilvesmäki, Lic.Sc. (Tech.)

What is flow analysis - II

- If we vary the flow timeout value, we may get different flow counts for different types of applications

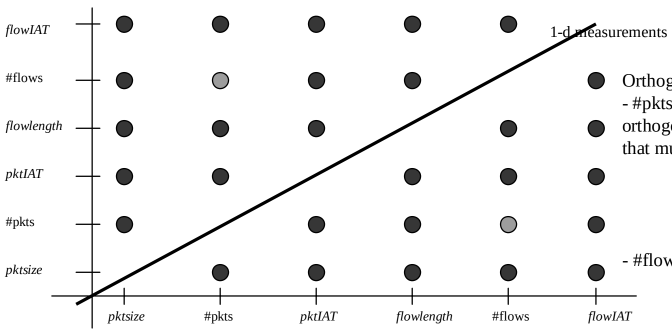


Possibilities of measured properties

- Against a port number (application) we can measure
  - #pkts, pktIAT, pktLength (per port number)
  - #flows, flowIAT, flowLength (per port number)
- Measurements are aggregated by the port numbers
  - loss of individual flow information
    - use of averages and variances where needed (IATs, Lengths)
  - flow/packet anomalies are lost in the ”noise”
  - #pkts and #flows per port number are exact (but aggregated) figures

Increasing the dimensionality of the measurements

- Packet phenomena may be better described if new, preferably orthogonal, measured properties are added
  - However, the curse of dimensionality may follow. Be careful!



-> #apps, appIAT etc.

Orthogonality (per application data):
- #pkts and #flows are relatively orthogonal (not depending on each other that much)
  - if we added a dimension like pkts/flows we would not get an orthogonal property
- #flows and flowIAT are less orthogonal

Design guidelines #2 and #3

- Do not imply policy within design
  - Use as value-neutral design as possible and leave room for freedom of choice
- Preserve end to end principle: ”If possible do everything at the edges.”
  - Profiling and marking should be done and used at the edges of the network
    - although measurements may, of course, be done anywhere in the network

Measurement analysis methods

- The measured properties may be sorted, or otherwise analyzed against
  - absolute boundaries (particular packet sizes, certain variance limits)
  - each other (all packets smaller/larger than the average packet size are classified/not classified)
- Multidimensional data may be clustered and classified
  - SOM, LVQ (if pre-classified samples are available) and other classification/cluster identification mechanisms
    - Remember Design Guideline #2

Automated measurement based policy creation (and a case) in one slide

- Decisions to be made
  - What header fields indicate the application?
    - TCP/UDP Sport
  - What is measured?
    - #pkts and # flows per TCP or UDP sport
  - How the measurements are analyzed?
    - Clustered, pre-classified samples, LVQ classifiers
      - Remember Design Guideline #2
  - How the results are interpeted and used?
    - Classification results provide lists of TCP/UDP Sports that indicate the applications to be classified to appropriate classes (remember Design Guideline #1 and #3)

Evaluation of the policy creation system

- Evaluate the network (element)
  - Use of transmission capacity, architecture dependent router resources (connection setup / class, packet forwarding / class etc.)
- Evaluate the effect on user
  - What applications are classified to priority
    - Relevance, application type, application count

Summary

- Policy is a definite goal, course or method of action to guide and determine present and future decisions in the network.
- As far as packet handling is concerned it might be smart to create policies (semi-) automatically, based on measurements.
- Measurements should be done on the packet level concentrating on the packet header infromation (and arrival information of the packet)
- Analysis of measurements is an upcoming field of research.