Admission Control for Packet Networks

Thomas Michael Bohnert
University of Coimbra, Portugal
E-mail: tbohnert@dei.uc.pt
currently visiting
Tampere University of Technology, Finland
E-mail: thomas.bohnert@tut.fi

OUTLINE:

- Admission Control (AC)

  Definition, Features, Purpose, Objectives, Placement

- Admission Control Functions

  General Structure, Design Requirements

  Parameter Based Admission Control

  Probe Based Admission Control

  Measurement Based Admission Control

1. Admission Control (AC)

1. Definition

- As a set of actions, consectively executed, in order to decide whether a customer can be admitted onto a network segment or not

2. Admission Control - Identity

- Admission criteria are Quality of Service (QoS) demands, contracted QoS commitments and resource availability
- Admit a customer if its resource demand superimposed with those of admitted customers can be accommodated without compromising QoS commitments
- → AC is a preventive congestion avoidance mechanism

  Not to be confused with TCP congestion control or traffic policing

  TCP congestion control reacts on network congestion. Reactive congestion control

  Policer shape traffic according to traffic profiles in order to prevent it from jamming the network

  AC regulates network load in advance. Does not touch admitted traffic. That's policer's job

3. Why Admission Control?

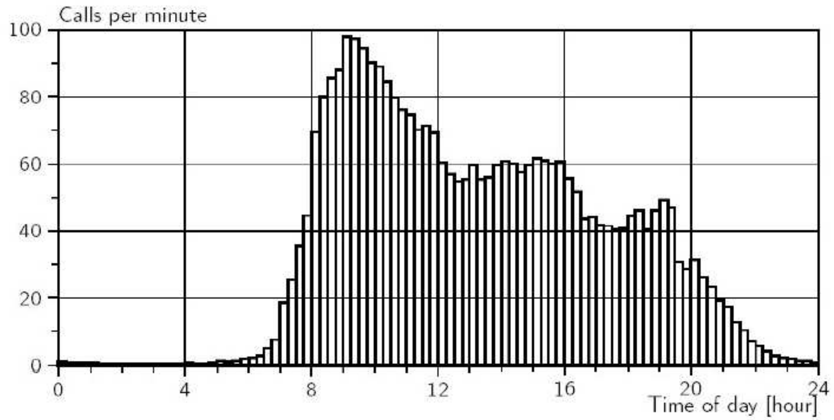


Figure 1: Concept of Busy Hour

- Traffic load and resource demand is time depended and absolutely random
- Networks are dimensioning is a trade-off between:

  Accommodation of worst-case traffic volume

  Captial Expenses (CAPEX) and Operational Expenses (OPEX)
- Network resources are commonly over booked, up to 1:50 but commonly 1:5 - 1:20
- Common practices: Maximal 3 percent average user blocking rate in busy hour

4. AC Objectives: Provider versus Client

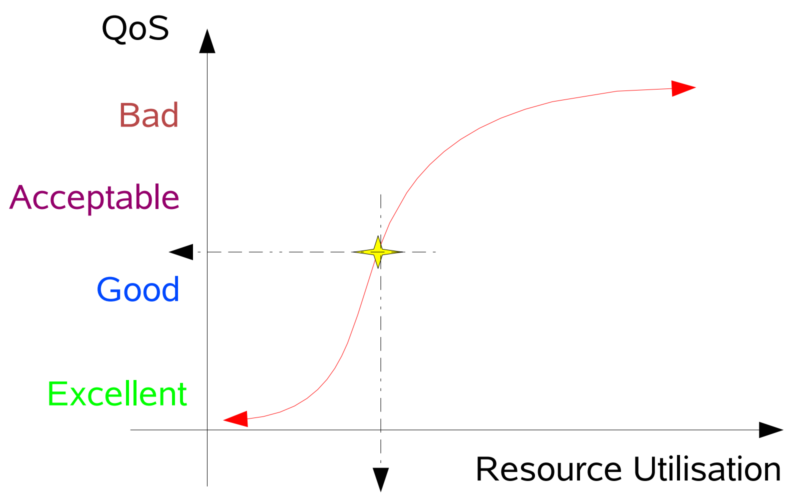


Figure 2: Client versus Provider: Conflicting Objectives

- The AC dilemma

  In IP networks, resource usage is typically maximised by leveraging Statistical Multiplexing

  But users expect and pay (!) for QoS

- ⇒ Optimal AC admits a maximum on flows onto a network, closely approaching the minimum QoS

5. Placement of Admission Control - Integrated AC

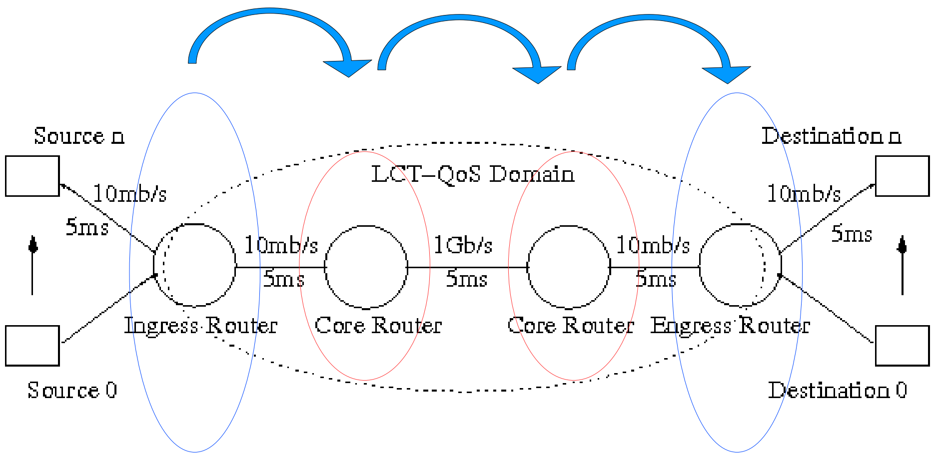


Figure 3: Local or Integrated Admission Control

- AC is implemented in each router. A new customer request admission hop-by-hop from its origin to its destination
- Hop-by-hop AC does assure End-to-End QoS and has been standardised by the IETF Integrated Service QoS Architecture [1].

6. Placement of Admission Control - Edge Based AC

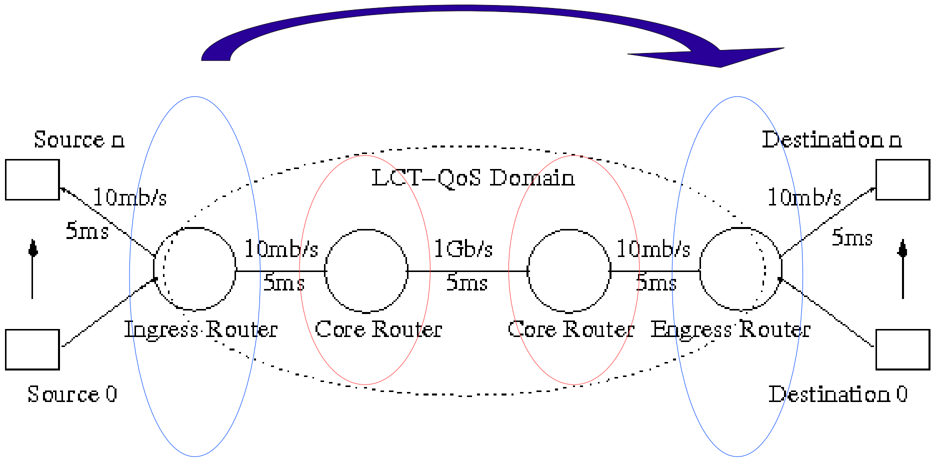


Figure 4: Admission Control at Network Edges

- AC is implemented in edge (ingress/egress) routers to regulate traffic load on access links
- From an End-to-End QoS perspective, this is justified by the current Internet configuration where congestion virtually exclusively occurs in access links
- Suits well in the IETF DiffServ architecture [2], but has not been standardised for

7. Placement of Admission Control - Endpoint AC

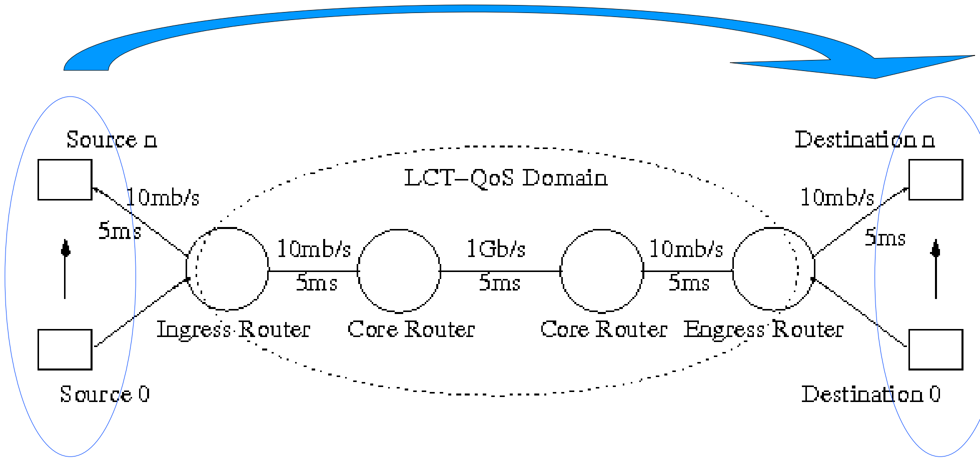


Figure 5: Endpoint Admission Control

- AC is implemented in customer equipment
- Application software, access routers, etc

8. Placement of Admission Control - Comparison

- Integrated AC

  + Assures true End-to-End QoS

  + Resource owner controls own resources

  - Lead time depends on number of hosts, i.e. can be extremely long

  - Considerable computation and state management burden for routers, especially core routers

- Edge Based AC

  + Complexity pushed to network edges. Core routes can concentrate on fast packet forwarding

  + Fits well in the incumbent Internet QoS architecture, DiffServ

  - No perfect QoS control as network core is considered as black box, free from QoS impairment

- Endpoint AC

  + Clients don't depend on network operators, advantageous for nomadic, roaming users

  - Resource owner, i.e. ISPs, do depend on client cooperation

9. ETSI Resource and Admission Control Subsystem - RACS [3]

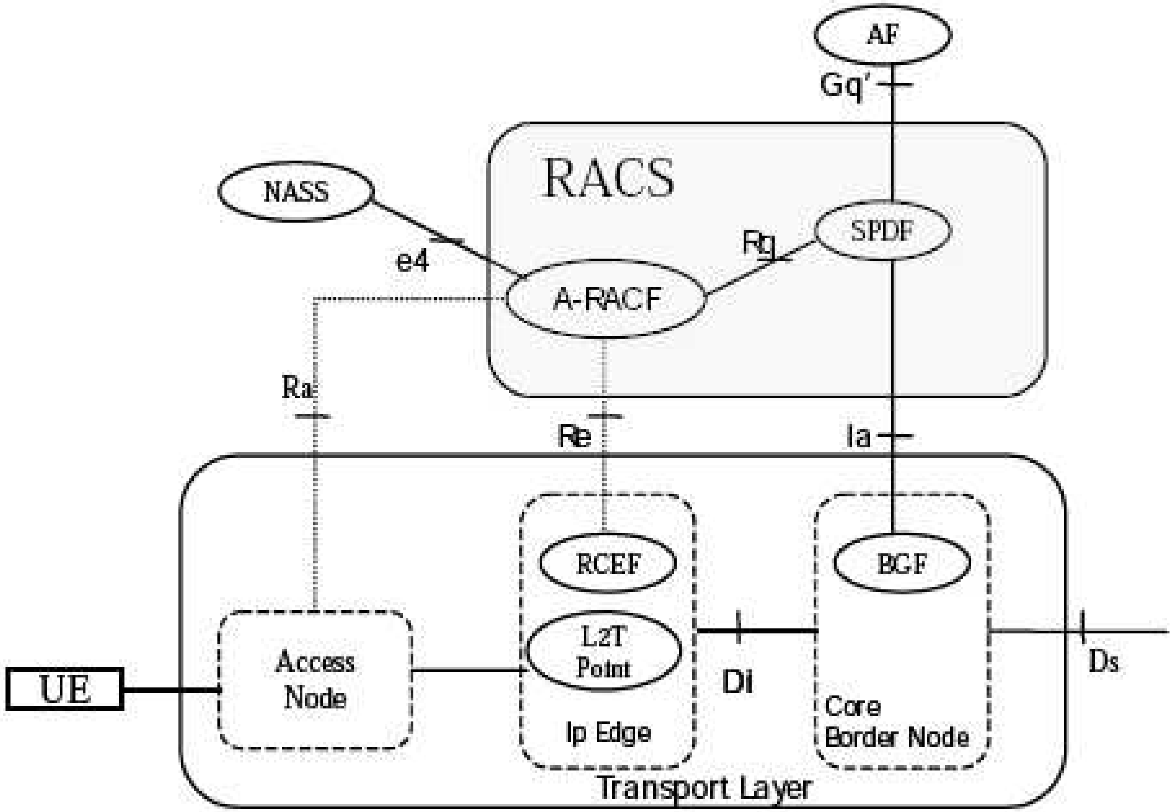


Figure 6: ETSI RACS - AC Integration

ETSI Resource and Admission Control Subsystem - RACS (cont'd)

- The Application Function (AF)

  Maps application layer QoS information into QoS request information to be sent the SPDF

- The Service-based Policy Decision Function (SPDF)

  Decides, based on policy rules defined by the network operator, if requests may be sent to an A-RACF

  Requests resources state information from the A-RACF

  Decides and communicates the decision back to the AF

- The Access-Resource and Admission Control Function (A-RACF)

  Checks whether the requested QoS resources can be made available for the involved access

- The Resource Control Enforcement Function (RCEF)

  Upstream and downstream traffic policing to ensure that traffic remains within the authorised limits

2. Admission Control Algorithms

1. AC Taxonomy

- Parameter-based Admission Control (PAC)

  Based on a priori knowledge and network traffic models, e.g. M/M/k

- Measurement-based Admission Control (MBAC)

  Probe-based Admission Control (PBAC)

  Generalisation based on actual measurements and estimation of QoS parameters

2. AC Deployment and Design Criteria

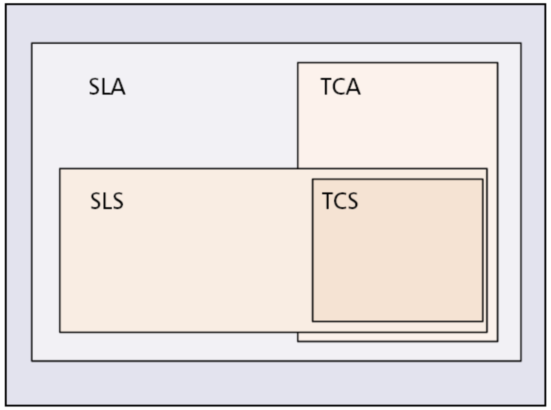


Figure 7: Service Level Specification (SLA) and Agreement (SLS) [4]

- Service Level Specification

  Negotiated between a customer and service provider on levels of service characteristics

  SLA is a broad term encompassing technical features as well as legal and charging aspects

- Service Level Specification

  Introduced to separate the technical part of the contract from the SLA

  Defined as set of parameters and values which define the service offered to traffic

AC Deployment and Design Criteria (cont'd)

- Configuration parameters must be Loss, Delay or Jitter, i.e in a QoS/SLA/SLS context
- QoS commitments must be guaranteed in any case
- Maximum utilisation of available resources
- Algorithm shall perform reliable and consistent under different conditions
- Performance fine-tuning should be autonomous
- A measurement-based algorithm should be maximally independent from traffic characteristics

3. Structure of a general Admission Control Function

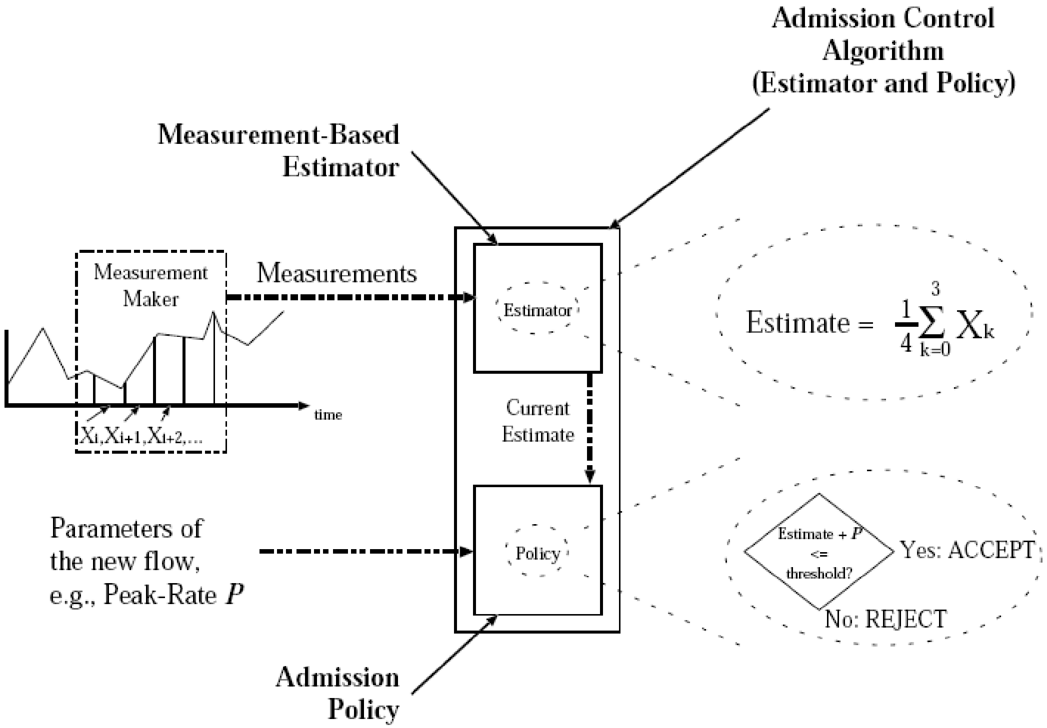


Figure 8: Logical Admission Control Modules [5]

Structure of a general Admission Control Function (cont'd)

- Meter Module

  Takes measurements, i.e. samples related random variables (RV)

  Buffer length at time t, server load at time t, quantities of arrival process, etc

  Units are number of packets, number of packets per time, packet interarrival time, number of bytes per time, etc

- Estimator Module

  Requests a sample of RVs from the Meter module

  Estimates quantities from the sampled time series

  Evaluates the estimates in purpose-built queueing models to quantify resources status, e.g. available bandwidth

- Policy Module

  Requests resources status from the Estimator module

  Decides based on a set of admission policy rules, resource demand and resource status about admission

4. Parameter-based Admission Control (PAC)

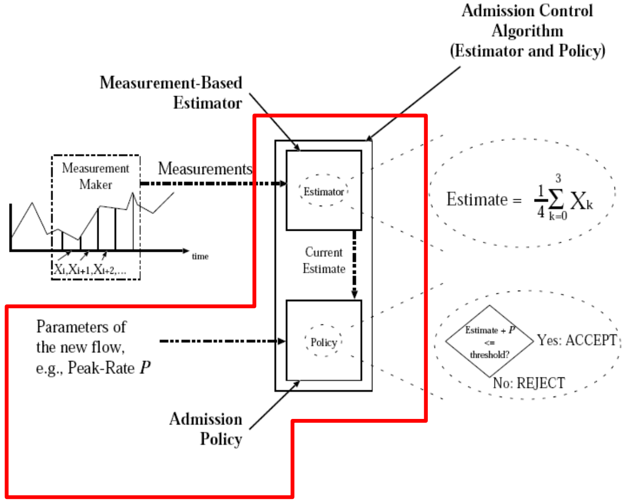


Figure 9: Components of Parameter-based Admission Control

- Concept

    Determine resource demand by evaluating source models

    Source models are based on a priori known characteristics

    No measurements, no estimation

    Simplest and in practice most frequent deployed algorithm

5. PAC - Rate or Simple Sum

- Most popular as very simple algorithm: Rate- or Simple Sum
-

$$boolAdmit = \begin{cases} \text{TRUE} & if \quad C_{aggr} \leq C_{link} \\ \text{FALSE} & else \end{cases} \tag{1}$$

$$C_{aggr} = \sum_{i=1}^{n} \rho_i + \rho_{new} \tag{2}$$

with C denoting capacity (bit/s) and ρ source peak rate (bit/s).

- Approximates the traditional teletraffic approach, M/M/m, Erlang-C
- Assumes continuous traffic emission, so-called Constant Bit Rate (CBR) sources
- Provides deterministic QoS if sources are appropriately policed

PAC - True nature of Traffic Sources

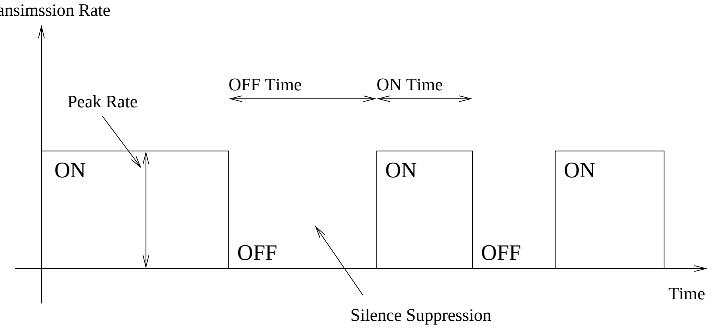


Figure 10: ON/OFF Sources Modelling

- Traffic sources practically never emit packets on a constant bit rate over long terms
- Example WEB browsing:

  Download web object - ON

  Digest information - OFF
- Voice Over IP with Voice Activity Detection (VAD)

  On and Off times are Exponentially distributed with mean 300ms and 600ms, respectively [6]
- Cascaded queues on a multi-hop path introduce jitter, i.e. ON and OFF times

PAC - Rate or Simple Sum (cont'd)

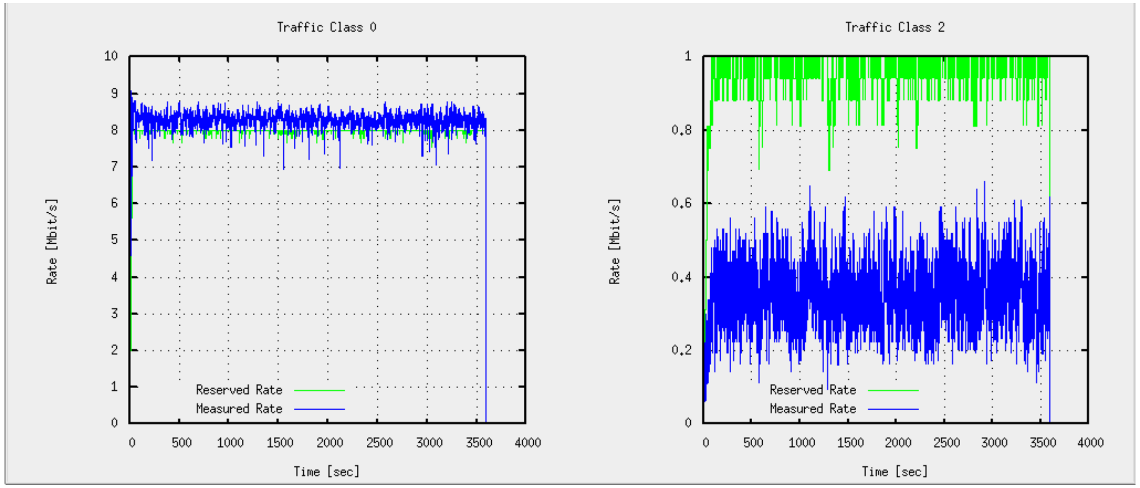


Figure 11: Simple Sum Algorithm: CBR source vs. ON/OFF Resource Utilisation

- The Simple Sum algorithm does not account for the true nature of the traffic
- Provides excellent QoS but leaves many resources unused
- Client's happiness is providers worry

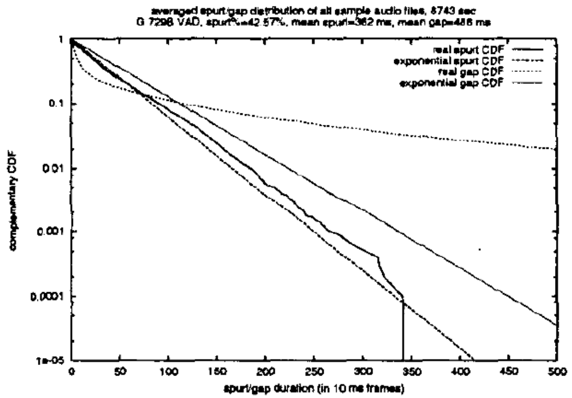
PAC - Are there no exact ON/OFF Models?

- Theoretically, exact modelling is possible, e.g. see [7]
- $$boolAdmit = \begin{cases} \text{TRUE} & if \quad P_{Loss} \leq P_{Loss_{SLS}} \\ \text{FALSE} & else \end{cases} \tag{3}$$

  with

  $$P_L = \frac{\sum_{S:\gamma_S \geq C} (\pi_S - F_S(x)))(\gamma_S - C)}{\sum_i^N \rho_i R_i} \tag{4}$$
- Source i 3-tuple $(R_i; b_i; \rho_i)$, with R_i denoting peak rate, b_i average ON period and ρ_i the fraction of time the source is active
- The buffer input process is described by a state vector $S = (s_1; s_2; \ldots; s_N)$, where s_i is 0 when source i is OFF and 1 else
- For any state the system input rate $\gamma_S = SR^T$ where $R = (R_; R_2; \ldots; R_N)$ is the peak rate vector
- The stationary probability that the input is in state S is denoted by π_S (system must be ergodic)
- $F_S(x)$ is the stationary probability that the queue length is smaller than x in state

PAC -Are there no exact ON/OFF Models? (cont'd)



(a) Averaged CDF by G.729B VAD

Figure 12: Exponential versus true ON/OFF Distribution [8]

- We saw, theoretically, exact modelling is possible
- However, ON and OFF time distributions are frequently much more complex than simple Exponential
- In reality, ON and OFF times often follow so-called heavy-tailed distributions
- The most popular one, the Pareto distribution which can have infinite variance!

PAC - Deployment Guidelines

- If PAC shall be deployed certain features are to be taken into account:
- Successful deployment depends on the availability of source models and their accuracy
- Can we be sure that assumptions match with reality?
- Factors are application characteristic, protocols, lower layer characteristics, etc
- Cascade of queues (networks) distort the source model
- Frequently depends on impractical parameters like Sustained Rate, which is impossible to know for interactive services
- Does typically not account for multiplexing gain, hence pose a pure worst-case model
- Only applicable for local integrated admission control in one hop-distance from the source

6. Probe-based Admission Control (PBAC)

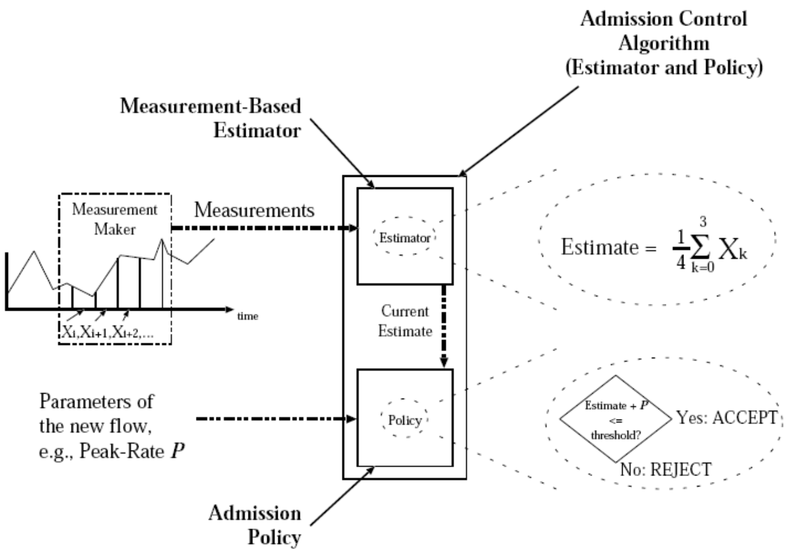


Figure 13: Components of Probe-based Admission Control

- Concept

  Probe the network for resources by sending packets from source to destination

  Measure QoS related parameters directly, mainly packet loss but sometimes also delay or jitter

  Simple to implement, adopted by manufacturers, e.g. CISCO VoIP Call Manager

PBAC - Formally Expressed

- Common algorithm structure
- $$boolAdmit = \begin{cases} \text{TRUE} & if \quad P_{Loss} \leq P_{Loss_{SLS}} \\ \text{FALSE} & else \end{cases} \tag{5}$$

  with

  $$P_{Loss} = 1/n \sum_{i=1}^{n} I_{\{Pkt_i == lost\}} \tag{6}$$

- Seems to be a simple, intuitive approach
- Does naturally fit well in the End-to-End principle of the Internet
- But are there and if, where are the weaknesses of this approach?

PBAC Architectural Issues - Distributed Policy

- Distributed policy

  Individual hosts decide about admission or rejection

- Example: A provider decides to map all VoIP traffic into one DiffServ class
- The admin knows: Different voice codecs have different sensitivities to loss, G.711 and G.724
- Consequently, he sets different admission thresholds, e.g. $P_l^{G.724} < P_l^{G.711}$?
- Correct?

- No. That's a common mistake. In this example, G.711 traffic will be admitted until $P_l^{G.711}$ is reached
- If enough G.711 traffic is offered, the VoIP class will operate on $P_l^{G.711}$, way too much for G.724 traffic
- Hence: "One class, on QoS objective"

PBAC Architectural Issues - Accuracy and Probing Delay

- Can we be sure that the loss fraction of a probe is below $P_{Loss_{SLS}}$?
- In order to be somewhat sure, the probe must last for many multiples of $1/P_{Loss_{SLS}}$ packet transmissions
- Clearly, this means that the setup delay depends on the probing transmission rate
- Example: $tx = 100 pkts/sec$, $P_{Loss_{SLS}} = 10^-3$
  $\Rightarrow$ Recommended probing time $\geq 30s$
- Is this acceptable for a Voice call?
- Try to guess you live in TOAS Lapinkaari and there is yet another false fire alarm. Until you can call the fire brigade coordination centre, to tell them about the false alarm, the fire fighters are already breaking into your kitchen ...

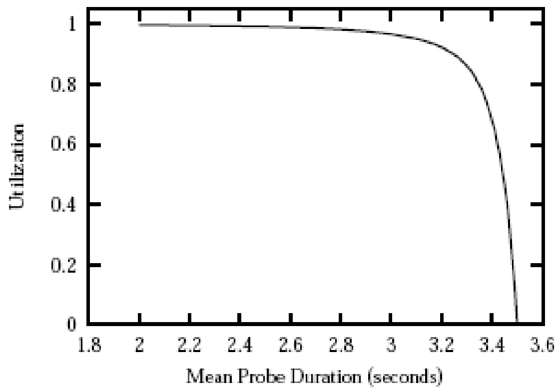
- What's the optimal probing rate?
- Are there ways to shorten setup delay?

PBAC Architectural Issues - In-band versus Out-band Probing

- In order to shorten the setup time, probing traffic could marked with a lower priority, e.g. a different DiffServ class
- This approach is called out-band probing, in contrast to in-band probing
- With out-of-band probing the data packet loss fraction is substantially lower than the probe packet loss fraction
- Thus the probing time needed to achieve a given level of loss is much less if one uses either out-of-band probing
- The problem, however, is that the difference is very difficult to quantify
- What if the lower priority class is totally overloaded while the higher priority class is almost unloaded?

- Hence, the question is: Is this really a propper setup delay shorten fix?

PBAC Architectural Issues - Trashing



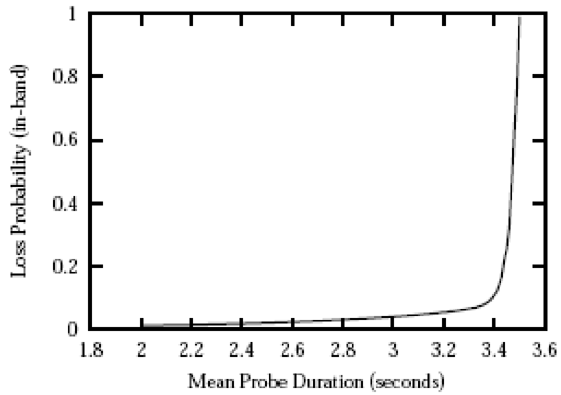


Figure 14: Effect on Network Performance by Probing traffic - Trashing [9]

- Example: Poisson arrivals (mean 3.5 sec), exponential flow times (mean 30 sec), link bandwidth is 10 Mbps, flow bandwidth 128 kbps
- When many flows are probing concurrently, they mutually prevent themselves from admission as the P_{Loss} rises by the probing traffic. This is called ”trashing”
- The origin of this problem is again the distributed nature of admission policy, i.e. admission requests are not serialised but happen concurrently

PBAC - Deployment Guidelines

- If implemented in applications, true End-to-End means
- Typically easy to implement
- Can be gradually deployed as no router, i.e. network architecture, modification is needed
- Requires applications to cooperate. A standard is, however, yet missing
- Difficult to implement in legacy applications
- Prone to poor accuracy and high setup delays
- Can cause or severely magnify congestion due to concurrent probing traffic
- Approach not favourable from providers point of view as resource control is left to clients. Contrary to the business relation

7. Measurement-based Admission Control (MBAC)

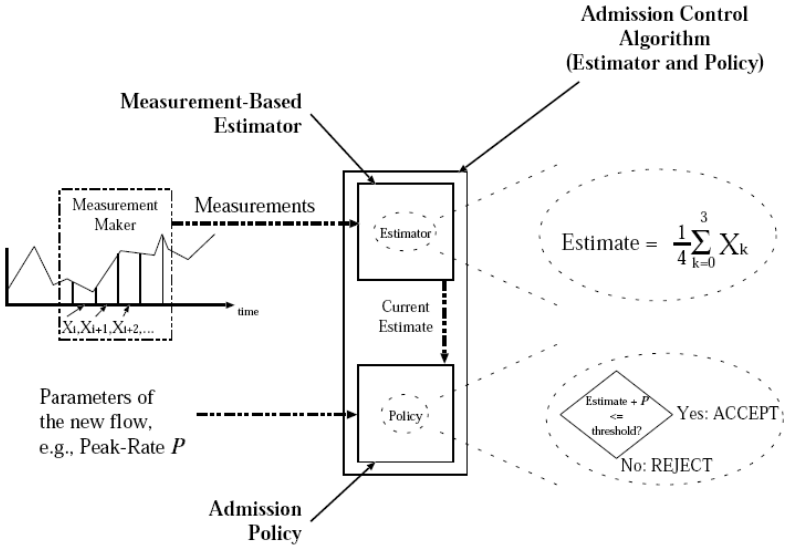


Figure 15: Components of Measurement-based Admission Control

- Concept

  Sample arrival/buffer occupancy/etc process locally

  Estimate (!) QoS parameters quantities by evaluating purpose built queueing models

  Local, integrated approach

8. MBAC - Equivalent Bandwidth based on Hoeffding Bounds

- Algorithm structure
-

$$boolAdmit = \begin{cases} \text{TRUE} & if \quad C(\epsilon) + p_f \leq C_{Link} \\ \text{FALSE} & else \end{cases} \tag{7}$$

  with

$$C(\epsilon) = \mu_S + \sqrt{\frac{ln(1/\epsilon)\sum_i^N (p_i)^2}{2}} \tag{8}$$

  where p_i denotes peak rate and N is the current number of admitted flows and S the aggregated arrival rate.

- $C(\epsilon)$ denotes the Equivalent Bandwidth
- It is defined as the bandwidth required to cater a traffic aggregate given a certain QoS objective
- In this particular case, the bandwidth needed such that $P\{S \geq \mu_S + n * t\} \leq \epsilon$ [10]

MBAC - Equivalent Bandwidth based on Hoeffding Bounds

- Derivation

$$P\{S \geq \mu_S + n * t\} \leq e^{\frac{-2n^2t^2}{\sum_i^n (p_i)^2}} \quad (9)$$

- This is the Hoeffding Bound. It bounds the sum S of a set of random variables. We require

$$P\{S \geq \mu_S + n * t\} \leq \epsilon \quad (10)$$

with μ_S is the mean of the sum. This yields

$$\epsilon = e^{\frac{-2n^2t^2}{\sum_i^n (p_i)^2}} \quad (11)$$

and therefore

$$t = \frac{1}{n}\sqrt{\frac{ln(1/\epsilon)\sum_i^N (p_i)^2}{2}} \quad (12)$$

- Hence, here the Hoeffding Bound is used to set a probabilistic threshold on the sum of the sources' transmission rates

MBAC - Equivalent Bandwidth based on Hoeffding Bounds

- How to estimate μ_S, the mean of the arrogate arrival rate?
- Originally, the Exponential Moving Average (EMA) has been proposed

$$\mu_S = (1 - w) * \mu_S + w * S_i \tag{13}$$

which has a time constant t as

$$t = \frac{-1}{ln(1 - w) * A} \tag{14}$$

- If we assume that the class arrival rate changes abruptly from 0 to 1, remains at the value 1. After t seconds, the estimated average arrival rate will have reached 63% of the new arrival rate value of 1. This yields

$$w \leq 1 - e^{-A/1} \tag{15}$$

- Hence, the definition of the time constant t controls the reactivity of the algorithm
- There is no guideline how to set t. Admins set this parameter based on experience as this parameter is highly traffic dependend

9. MBAC - Deployment Guidelines

- Resource management is under provider's control
- Adapts dynamically to changing conditions
- Estimates taken from traffic aggregates, does implicitely account for multiplexing gain
- Passive measurements, does not impair packet forwarding
- Application/Service independent
- Generally appears as the most flexible, perhaps universal method

but

- Performance varies considerably for conditions, i.e traffic characteristics
- Performance strongly depends on individual, model specific performance parameter fine-tuning
- In many cases, algorithms miss targeted QoS objectives due to ill configuration

3. Time for your questions

References

[1] R. Braden, D. Clark, and S. Shenker. *Integrated Services in the Internet Architecture: An Overview*. RFC 1633. IETF, jun 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An Architecture for Differentiated Services*. RFC 2475. IETF, dec 1998.

[3] *Resource and Admission Control Sub-system (RACS); Functional Architecture*. Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN). ETSI, Sophia Antipolis Cedex, France, jun 2006.

[4] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz. Quality of service terminology in IP networks. *IEEE Communications Magazine*, 41(3):153–159, mar 2003.

[5] A. W. Moore. *Measurement-based management of network resources*. PhD thesis, Universtiy of Cambridge, 2002.

[6] A. P. Markopoulou, F. A. Tobagi, and M. J. Karam. Assessing the quality of voice communications over internet backbones. *IEEE Transactions on Networking*, 11(5):747–760, oct 2003.

[7] Y. Xu and R. Guerin. Individual QoS versus Aggregate QoS: A Loss Performance Study. In *IEEE*

INFOCOM'02, volume 3, pages 1170–1179, New York, USA, jun 2002. IEEE.

[8] H. Schulzrinne and W. Jiang. Analysis of on-off patterns in VoIP and their effect on voicetraffic aggregation. In *Ninth International Conference on Computer Communications and Networks*, pages 82–87, mar 2000.

[9] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint admission control: Architectural issues and performance. In *Sigcomm*, pages 57 – 69, 2000.

[10] S. Floyd. Comments on measurement-based admissions control for controlled-load services. Technical report, August 1996.